



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

**“APLICACIONES ALTERNAS PARA EL INGENIERO
MECÁNICO ELECTRICISTA EN EL SISTEMA
OPERATIVO GNU/LINUX”.**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECÁNICO ELECTRICISTA

ÁREA: INGENIERÍA ELÉCTRICA ELECTRÓNICA

P R E S E N T A:

JESÚS FELIPE ELGUERA GARCÍA.

ASESOR:

ING. FRANCISCO RAÚL ORTÍZ GONZÁLEZ



MÉXICO, 2011.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



DEDICATORIA:

A mis padres,

José y Paula:

Sus esfuerzos no han sido en vano.

A Isela:

Por ser parte importante en mi vida.

A todos mis hermanos:

Quienes crecieron conmigo.

AGRADECIMIENTOS:

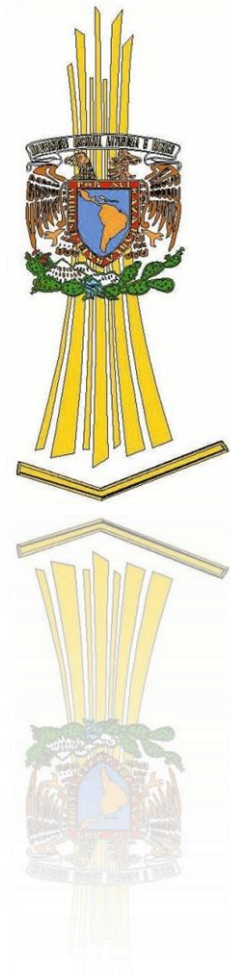
Quiero agradecer en primer lugar a Dios, por darme la capacidad, la inteligencia, la determinación, la tenacidad..., la vida, para poder terminar mi carrera.

Sin duda, mis agradecimientos para mis padres: José Elguera Mendoza y Paula García Cahue, quienes, con su cariño y apoyo, en todo momento de mi vida, me han sabido guiar y educar para ser un hombre de bien.

A mi novia Isela, quiero agradecerle todo su cariño, paciencia y apoyo.

Mi agradecimiento para mi asesor de tesis: ingeniero Francisco Raúl Ortíz González, quien confió en mí y me otorgó todo su apoyo.

Y, por supuesto, también agradezco a la UNAM, a la FES Aragón y a todos los profesores que tuve durante mi carrera, por darme las bases necesarias para desarrollarme profesionalmente. Así como, al ingeniero Lucio Domínguez y a la licenciada Carmen Herrera, por sus aportaciones y consejos.





Objetivos.	I
Justificación.	II
Metodología.	IV
Alcances y limitaciones.	V
Introducción.	VI
Capítulo I. Generalidades.	1
1.1. GNU.....	1
1.2. El Núcleo (Linux).....	4
1.3. Etimología.	8
1.4. Controversia por el nombre (Linux vs GNU/Linux).....	9
1.5. Free Software Foundation.	10
1.6. Filosofía del software libre.....	11
1.7. Distribuciones.....	17
Capítulo II. El sistema operativo GNU/Linux.	38
2.1. Introducción.....	38
2.2. ¿Cómo elegir una distribución?.....	40
2.3. Ubuntu.	41
2.3.1. Diferencia con otras distribuciones.....	42
2.3.2. Principios de Ubuntu.	43
2.3.3. Soporte de la distribución.....	44
2.3.4. El escritorio.	44
2.3.5. Numeración de versiones.....	45





2.4.	Hardware soportado.....	46
2.4.1.	Arquitecturas soportadas.....	47
2.4.2.	Procesadores múltiples.....	47
2.4.3.	Tarjetas gráficas.....	47
2.4.4.	Hardware de conectividad de red.....	48
2.4.5.	Tarjetas de red inalámbrica.....	48
2.4.6.	Periféricos y otro hardware.....	49
2.5.	Requisitos de memoria y espacio en disco.....	49
2.6.	Iniciando el sistema.....	50
2.6.1.	GRUB.....	51
2.6.2.	El arranque.....	51
2.7.	Entorno de escritorio.....	52
2.7.1.	GNOME.....	53
2.7.2.	KDE.....	57
2.8.	La terminal.....	59
2.9.	Usuarios y grupos.....	60
2.10.	Paquetes y administración de paquetes.....	62
2.10.1.	¿Qué es un paquete?.....	62
2.10.2.	¿Fuente o binario?.....	62
2.10.3.	Dependencias de paquetes.....	63
2.10.4.	Canales de software/repositorios.....	64
2.11.	Instalar aplicaciones.....	66
2.11.1.	Instalar paquetes de los repositorios.....	67
2.11.2.	Instalar paquetes de otras fuentes.....	69
2.11.3.	Compilar paquetes.....	69
2.12.	Actualizaciones automáticas.....	70
2.13.	Sistemas de archivos.....	70
2.13.1.	Archivos de configuración importantes.....	74





Capítulo III. Aplicaciones para ingeniería mecánica eléctrica en GNU/Linux. 76

3.1. Introducción.	76
3.2. Aplicaciones para CAD (Computer Aided Design – Diseño Asistido por Computadora).	77
3.2.1. QCAD (CAD 2D).	77
3.2.2. Blender (Modelado 3D).	82
3.2.3. Medusa4 (2D y 3D).	88
3.2.4. Otras herramientas CAD.	92
3.2.4.1. BricsCAD.	92
3.2.4.2. FreeCAD.	92
3.2.4.3. VariCAD.	93
3.3. Aplicaciones para EDA (Electronic Design Automation – Diseño Electrónico Automatizado).	93
3.3.1. gEDA.	93
3.3.2. Alliance.	98
3.3.3. KICAD.	101
3.3.4. Otras aplicaciones para electrónica.	108
3.3.4.1. GHDL.	108
3.3.4.2. Piklab.	109
3.3.4.3. VIPEC.	109
3.4. Aplicaciones para matemáticas.	110
3.4.1. GNU Octave.	110
3.4.2. Scilab.	113
3.4.3. Maxima.	116
3.4.4. Otras aplicaciones para matemáticas.	120
3.4.4.1. Euler.	120
3.4.4.2. Genius.	121
3.4.4.3. Termograf.	121





3.5. Utilidades para la producción diaria.....	121
3.5.1. OpenOffice.org.....	121
3.5.2. Firefox.....	125
Conclusiones.	126
Apéndice A. Instalación de Ubuntu.	128
Apéndice B. Comandos más utilizados en GNU/Linux.	133
Bibliografía.....	140
Mesografía.....	141





Objetivos.

Generales:

- Este trabajo de tesis pretende '*atacar*' la problemática (*que es un hecho real*) a la que se enfrentan los profesionistas de ingeniería mecánica eléctrica (*y muy en particular en el área de eléctrica-electrónica*): el conocimiento de otro sistema operativo.
- Dar otra perspectiva de trabajo, al estudiante o profesionista, que la puede canalizar como una herramienta más en su vida profesional.

Particulares:

- Tener la habilidad de operar el sistema operativo GNU/Linux, con propósitos y en tareas tan diversas, como pueden ser; en los propósitos profesionales o de la vida cotidiana; en tareas tan sencillas o tan complejas, dependiendo del perfil profesional que se tenga o que se pretenda tener.
- Dar a conocer aplicaciones, más probables, que un ingeniero mecánico electricista se puede encontrar en su campo laboral y como obtenerlas e instalarlas.





Justificación.

¿Para qué aprender a utilizar otro sistema operativo?

- A un ingeniero, el saber utilizar un segundo, o más sistemas operativos, le da un valor agregado a su perfil profesional; con esto se ve beneficiado en sus competencias laborales (*generalmente se prefiere a un profesionalista que conozca más de un sistema operativo, así como a uno que domine más de un idioma*).
- El tener la oportunidad de elegir (*en general cualquier cosa*) es un poder y una libertad, que garantiza la capacidad de análisis de un ingeniero, pues conoce los pros y los contras de cada sistema elegible, haciendo una evaluación y eligiendo el que mejor se adapte a determinada tarea.

¿Por qué GNU/Linux y no otro sistema operativo?

- Hay una gran variedad de sistemas operativos; desde los comerciales y por ello más conocidos, como son: Windows y Mac OS; hasta los no tan conocidos, pero no por ello menos importantes o potentes: GNU/Linux, Free BSD, Solaris.
- El sistema GNU/Linux, es muy flexible en cuanto a hardware se refiere, se puede tener un sistema muy potente con pocos recursos de hardware. Por ejemplo, en el caso de la distribución Debian GNU/Linux (*que es en la que está basada la distribución con que se ha trabajado en esta tesis*), ningún otro sistema operativo, comercial o no comercial, tiene soporte para tantas arquitecturas de procesador (*14 arquitecturas*), alpha, amd64, hppa, i386, powerpc, sparc, s390; etc.





- Un ingeniero tiene una gran variedad de aplicaciones a su alcance en GNU/Linux.
Si es, por ejemplo, un ingeniero con perfil electrónico puede desempeñar tareas de manera eficiente en el área de diseño eléctrico y electrónico, en la programación de microcontroladores o PLD's (*desde un simple gal, hasta un potente fpga*), creación de redes industriales (*con PLC's*); etc. Si su perfil es eléctrico, puede realizar planos de instalaciones eléctricas, puede simular redes eléctricas; etc. Si por otro lado, su perfil es mecánico, crea modelos mecánicos, simula mecanismos, realiza los dibujos técnicos; etc.
- Si un estudiante de ingeniería mecánica eléctrica tiene la oportunidad de elección; en este caso, de usar un sistema operativo diferente y, en consecuencia, aplicaciones basadas en ese sistema operativo; le dará una visión más amplia para resolver problemas en su área de conocimiento; lo cual le dará una ventaja competitiva, en el campo laboral, con respecto a un estudiante que sólo sabe usar lo que le enseñaron en su plan de estudios.





Metodología.

- La metodología empleada para el desarrollo de la tesis fue utilizar como estrategia la realización de una investigación de carácter cualitativa. Con ello se pretende dar solución a un problema que no puede ser explicado por números, como se puede hacer con una investigación cuantitativa. Se realizó un trabajo descriptivo y documental.
- En la primera fase se realizó una colección de material bibliográfico y mesográfico.
- Posteriormente, en una segunda fase, se hizo un extracto y análisis de la información recopilada, dándole una estructura y desechando lo que no era útil para alcanzar los objetivos.
- En una tercera fase de la investigación, se realizaron pruebas en la instalación del sistema operativo.
- En la cuarta fase, se comprobó y fiabilizó que el sistema operativo y todas las aplicaciones, presentadas en la tesis, son funcionales. Esto quiere decir que, con toda confianza, se tiene la veracidad de cómo obtener el sistema operativo y cada aplicación, de cómo instalar el sistema operativo y cada aplicación.
- Toda aplicación, para ser recomendada en este trabajo, fue probada y verificada para que cumpla con el propósito especificado por los creadores de las aplicaciones.
- Y, finalmente se llegó a conclusiones con las cuales se cumplieron los objetivos planteados.





Alcances y limitaciones.

Se establecen las bases generales de un sistema GNU/Linux, es lo que cualquier usuario debe saber del sistema. Aprenderá a administrar archivos, conocer la estructura del sistema de archivos; saber cómo obtener los paquetes, así como instalarlos y saber dónde se instalaron; desinstalar aplicaciones; conocer y utilizar los comandos más frecuentes; etc.

Si por alguna razón se requieren conocimientos más profundos del sistema, por ejemplo: utilizar emuladores o mejor aún una máquina virtual para acceder a una aplicación que sólo exista en Windows; crear redes, para compartir datos, entre sistemas GNU/Linux y otros sistemas operativos; o crear scripts para automatizar algún proceso; son conocimientos que no se contemplan en esta tesis y requerirá acceder a bibliografía más especializada.

Son los alcances y limitaciones para éste trabajo de tesis, no así para los usuarios, pues teniendo los conocimientos básicos, podrá aprender lo que necesite.





Introducción.

En la actualidad muchos usuarios, universidades, empresas y gobiernos están haciendo la migración a GNU/Linux y eso ha despertado grandes expectativas en todo el mundo. Entendiendo por migración a utilizar aplicaciones que les son necesarias en sus actividades diarias en un sistema operativo abierto y libre, totalmente lo contrario a los sistemas operativos comerciales, con los que actualmente trabajan.

Mucha gente no hace la migración por miedo a no tener aplicaciones que son utilizadas a diario en el campo profesional, especialmente para los ingenieros (*ya sea electricistas, mecánicos, civiles, químicos, etc.*) que necesitan aplicaciones avanzadas para la realización de sus actividades y la eficiencia y éxito en sus profesiones.

Este trabajo de tesis pretende darles a conocer, principalmente, a los estudiantes de ingeniería mecánica eléctrica, alternativas de trabajo a lo que están acostumbrados o a lo que conocen.

GNU/Linux es un sistema operativo, basado en Unix. Hay dos características muy peculiares que diferencian a GNU/Linux del resto de sistemas operativos que podemos encontrar en el mercado: la primera, es que es libre, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna empresa desarrolladora de software por el uso del mismo; la segunda, es que el sistema viene acompañado del código fuente.

GNU/Linux se distribuye bajo la Licencia Pública General de GNU, por lo tanto, el código fuente tiene que estar siempre accesible y cualquier modificación o trabajo derivado debe de tener esta licencia.





En el capítulo I se da un panorama de cómo se gestó el sistema operativo GNU/Linux, así como la filosofía en la que se basaron sus creadores; cómo inició el proyecto, quiénes lo crearon, de donde proviene el nombre, cómo ha evolucionado desde sus orígenes hasta llegar a las modernas distribuciones que existen en la actualidad.

De estas distribuciones se destacan las más importantes y sus características principales; así como, las diferencias que existen entre ellas. Esto nos ayuda a determinar la opción más idónea para iniciar en el sistema operativo GNU/Linux.

En el capítulo II se describen las características generales y las ventajas del sistema GNU/Linux (*particularmente nos enfocamos en la distribución más popular y fácil de trabajar, que es Ubuntu, en su versión 10.04*) como, por ejemplo: es menos probable que un sistema basado en Linux se colapse, además tiene mejor capacidad para ejecutar múltiples programas al mismo tiempo y es más seguro que muchos otros sistemas operativos.

Debido a estas ventajas, GNU/Linux es el sistema operativo que ha experimentado mayor crecimiento en el mercado de los servidores. Y, últimamente, GNU/Linux está empezando a ser popular entre los usuarios domésticos y en el sector empresarial.

En el capítulo III se describen algunas aplicaciones que pueden ser útiles a los estudiantes y profesionales de ingeniería. Enfatizando que GNU/Linux presenta muchas ventajas debido a que:

- ✓ Generalmente son mas robustas y rentables que sus alternativas.
- ✓ Proveen un flexible y conveniente entorno de desarrollo.
- ✓ Tienen un diseño libre, extendible y modular.
- ✓ Son orientadas a redes.





- ✓ Se pueden hacer máquinas potentes y seguras con muy pocos recursos.
- ✓ Día a día, más y más programas y aplicaciones están disponibles para este sistema operativo y la calidad de los mismos aumenta de versión a versión.
- ✓ Cada vez son más las empresas, desarrolladoras de software comercial, que distribuyen sus productos para Linux.





CAPÍTULO I. GENERALIDADES.

1.1. GNU.

La historia de GNU (ver **1.3 Etimología**) está ligada a la del sistema operativo Unix.



Fig. 1.1.1. Ken Thompson y Dennis Ritchie, Creadores de Unix

UNIX es un sistema operativo desarrollado, en principio, en el año de 1969 por un grupo de empleados de los laboratorios Bell de AT&T, entre los que figuran Ken Thompson, Dennis Ritchie (*Fig. 1.1.1*) y Douglas McIlroy. Es un sistema operativo no libre, portable, multitarea y multiusuario; esto

quiere decir que es un software para ejecutarse en diferentes plataformas (*portabilidad*), el código fuente del software es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra. A mayor portabilidad menor es la dependencia del software con respecto a la plataforma. Es un software que permite que varios procesos sean ejecutados al mismo tiempo compartiendo uno o más procesadores o núcleos (*multitarea*); además permite proveer servicio y procesamiento a múltiples usuarios simultáneamente (*multiusuario*).

Los recursos que se comparten son por lo regular una combinación de:

- ✓ Procesador.
- ✓ Memoria.
- ✓ Dispositivos de almacenamiento.
- ✓ Programas.
- ✓ Periféricos como impresoras, plotters, scanners, etc.





El proyecto GNU fue iniciado, por Richard Stallman (*Fig. 1.1.2*), con el objetivo de crear un sistema operativo clon de Unix compuesto sólo de software libre.



Fig. 1.1.2 Richard Stallman

El 27 de septiembre de 1983 se anunció públicamente el proyecto por primera vez en el grupo de noticias net.unix-wizards. Al anuncio original, siguieron otros ensayos escritos por Richard Stallman que establecieron sus motivaciones para realizar el proyecto GNU, entre las que destaca: *“volver al espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras”*. El desarrollo de software comenzó en enero de 1984.

El sistema GNU fue diseñado para ser totalmente compatible con UNIX. El hecho de ser compatible con la arquitectura de UNIX implica que GNU esté compuesto de pequeñas piezas individuales de software, muchas de las cuales ya estaban disponibles, que pudieron ser adaptados y reutilizados; otros en cambio tuvieron que ser codificados nuevamente.

Para asegurar que el software GNU permaneciera libre, para que todos los usuarios pudieran ejecutarlo, copiarlo, modificarlo y distribuirlo, el proyecto debía ser liberado bajo una licencia diseñada para garantizar esos derechos al tiempo que evite restricciones posteriores de los mismos. La idea se conoce en inglés como copyleft, 'copia permitida', (*en clara oposición a copyright, 'derecho de copia'*.) y está contenida en la Licencia General Pública de GNU (*GPL*). En 1985, Stallman creó la Free Software Foundation (*Fundación para el Software Libre*) para dar soportes logísticos, legales y financieros al proyecto GNU.

La Free Software Foundation contrató programadores para contribuir a GNU, aunque una porción sustancial del desarrollo fue, y continúa siendo, producida por voluntarios.





A medida que GNU ganaba renombre, negocios interesados comenzaron a contribuir al desarrollo o comercialización de sus productos y el correspondiente soporte técnico. El más prominente y exitoso de ellos fue Cygnus Solutions, ahora parte de Red Hat.

A continuación se enlistan algunos programas desarrollados y mantenidos por el proyecto GNU:

- ✓ Bison. Generador de analizadores sintácticos.
- ✓ Bash. Intérprete de comandos.
- ✓ BFD. Archivos de bibliotecas.
- ✓ Binutils. Ensamblador GNU, Enlazador GNU.
- ✓ DotGNU. Substituto de .NET.
- ✓ Emacs. Editor de texto extensible y autodocumentado.
- ✓ GCC. Compilador optimizado para varios lenguajes, en particular C.
- ✓ GNU Ghostscript. Aplicaciones para PostScript y PDF.
- ✓ GIMP. Programa de edición fotográfica.
- ✓ Glibc. Biblioteca para lenguaje C.
- ✓ GMP. Biblioteca para cálculos con precisión arbitraria.
- ✓ GNOME. Ambiente de escritorio gráfico.
- ✓ GSL. Biblioteca Científica para GNU.
- ✓ Gzip. Aplicaciones y bibliotecas para compresión de datos.
- ✓ Hurd. Un micronúcleo y un conjunto de servidores que funcionan del mismo modo que el núcleo UNIX.
- ✓ Maxima. Un sistema para cálculos algebraicos.
- ✓ Octave. Un programa para cómputo numérico similar a MATLAB.
- ✓ Texinfo. Sistema de documentación.





1.2. El Núcleo (*Linux*).

GNU pretendía la utilización de su propio proyecto de núcleo, llamado Hurd, sin embargo, su desarrollo no continuó como se esperaba. Así que al aparecer el núcleo Linux se completaron los requisitos mínimos y surgió el sistema operativo GNU que utilizaba el núcleo Linux.

El núcleo o kernel es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora; o en forma más básica, es el encargado de gestionar recursos. Como hay muchos programas y el acceso al hardware es limitado, también se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo. El núcleo tiene grandes poderes sobre la utilización de los recursos materiales (*hardware*), en particular, de la memoria. Administra el espacio en el sistema de archivos, que incluye: acceso, protección y administración de usuarios; comunicación entre usuarios y entre procesos.



Fig. 1.2.1. Linus Torvalds

En abril de 1991; Linus Torvalds (*Fig. 1.2.1*), de 21 años, cuando estudiaba en la Universidad de Helsinki, en Finlandia; comenzó un proyecto que más tarde llegó a ser el núcleo Linux. Torvalds solía tener acceso en los grandes servidores UNIX de la universidad. Empezó con un intento por obtener un núcleo de sistema operativo gratuito similar a Unix que funcionara con microprocesadores Intel 80386.

El sistema operativo que él usó durante el desarrollo fue Minix y el compilador inicial fue el GNU C, que aún es la opción principal para compilar Linux hoy en día. MINIX es un clon de Unix, distribuido junto con su código fuente (*de mucho menor tamaño*), desarrollado por el profesor Andrew S. Tanenbaum en 1987. La última versión oficial de Minix es la 3.1.5, publicada el 5 de Noviembre de 2009.





Minix fue creado para enseñar a los alumnos el diseño de sistemas operativos en la Vrije Universiteit de Ámsterdam. La razón de su desarrollo fue porque Unix estaba bajo restricciones de licencia de AT&T, era demasiado complicado y corría sobre máquinas complejas; algo completamente antipedagógico.

Gracias a su reducido tamaño y su amplia documentación, resulta bastante apropiado para personas que desean instalar un sistema operativo compatible con Unix en su computadora personal, así como aprender sobre su funcionamiento interno.

Linus Torvalds escribió el programa expresamente para el hardware que usaba, e independiente de un sistema operativo, porque quiso usar las funciones de su nueva computadora personal con un procesador 80386.

El 25 de agosto de 1991, anunció este sistema en un envío a la red Usenet, en el newsgroup (*grupo de noticias*): comp.os.minix. Esto es parte del texto:

“Hola a todos aquellos que usan Minix. Estoy haciendo un sistema operativo (gratis) (solamente una afición, no será grande ni profesional como el GNU) para clones 386(486)... Me gustaría recibir cualquier comentario sobre las cosas que les gustan/disgustan en Minix, ya que mi SO se le parece un poco...”

... Conseguiré algo práctico dentro de unos meses, me gustaría saber qué características quiere la mayoría de la gente. Cualquier sugerencia es bienvenida, pero no prometeré que las pondré en práctica :-)

Linus Benedict Torvalds (torvalds@kruuna.helsinki.fi)”





En las Fig. 1.2.2 y 1.2.3 se visualiza el desarrollo cronológico del núcleo:

Versión	Fecha	Archivos	Líneas de código	Tamaño (MB)	Tamaño comprimido (MB) (bz2)
0.01	17 de septiembre de 1991	88	8,413	0.267	0.06
0.11	8 de diciembre de 1991	100	11,907	0.363	0.076
0.95	7 de marzo de 1992	122	19,200	0.533	0.111
1.0.0	13 de marzo de 1994	561	170,581	4.633	0.969
1.1.0	6 de abril de 1994	561	170,320		
1.2.0	6 de marzo de 1995	909	294,623		
1.3.0	12 de junio de 1995	992	323,581		
2.0.0	9 de junio de 1996	2,015	716,119	21.7	4.499
2.1.0	30 de septiembre de 1996	1,727	735,736		
2.2.0	26 de enero de 1999	4,599	1,676,182		
2.3.0	11 de mayo de 1999	4,721	1,763,358		
2.4.0	4 de enero de 2001	8,187	3,158,560	96.8	18.79
2.5.0	23 de noviembre de 2001	9,893	3,833,603		
2.6.0	18 de diciembre de 2003	15,007	5,475,685	170.7	31.7
2.6.25	16 de abril de 2008	23,810	8,396,250	258.8	46.4
2.6.30	10 de junio de 2009	27,878	10,419,567	322.3	56.7

Fig. 1.2.2. Desarrollo del Núcleo



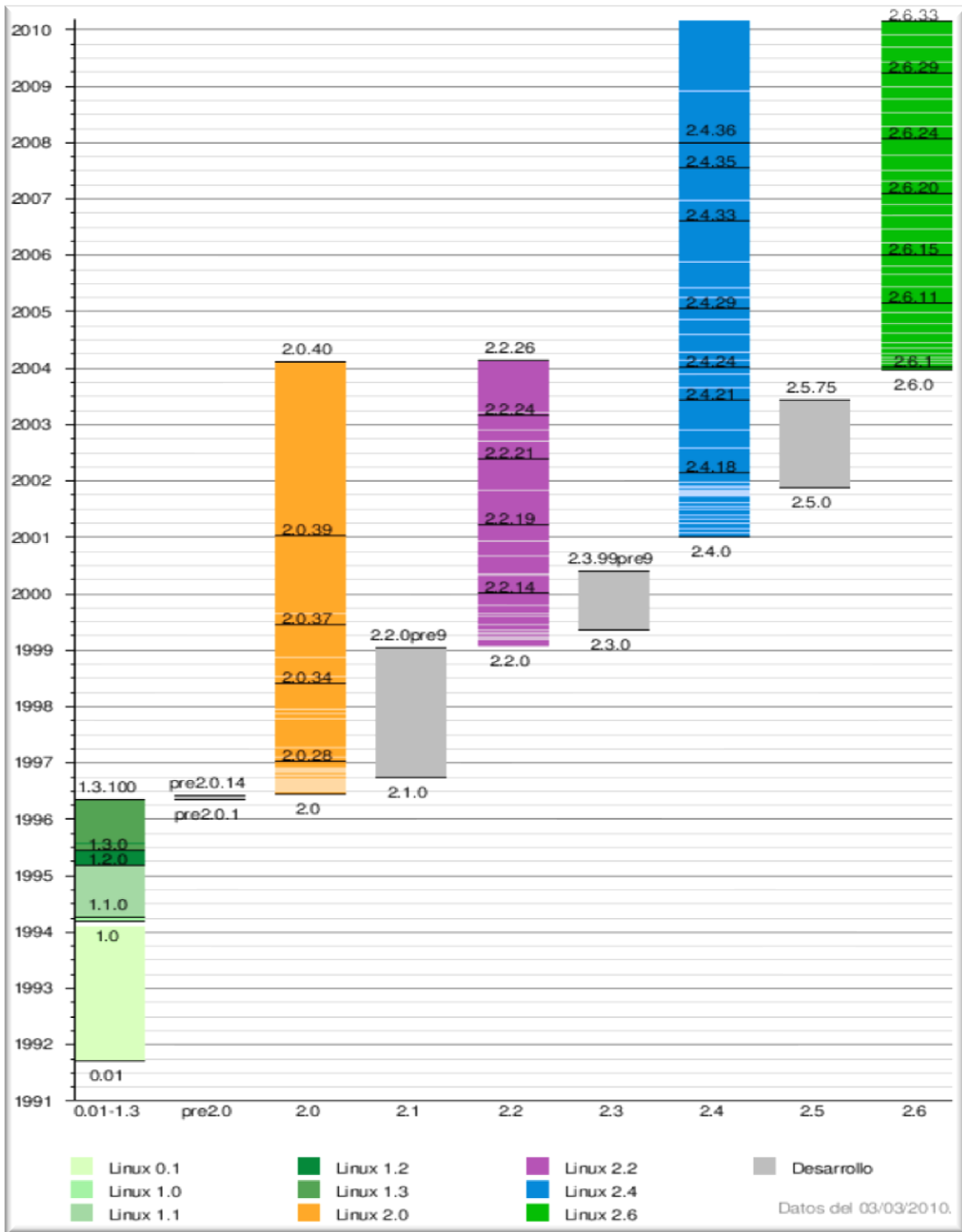


Fig. 1.2.3. Versiones del Núcleo.





1.3. Etimología.

GNU es un acrónimo recursivo para "**G**nu **N**o es **U**nix", viene de las herramientas básicas del sistema operativo, creadas por el proyecto GNU, iniciado por Richard Stallman en 1983 y mantenido por la Free Software Foundation.

En español, se recomienda pronunciarlo 'ñu' como el antílope africano, o fonéticamente; por ello, el término mayoritariamente se deletrea (*G-N-U*) para su mejor comprensión. En sus escritos y entrevistas Richard Stallman finalmente siempre dice: "*Se puede pronunciar de cualquier forma, la única pronunciación errónea es decirle 'linux'.*".

El nombre Linux viene del núcleo Linux, inicialmente escrito por Linus Torvalds en 1991. Linus Torvalds había querido llamar a su invención 'Freax', una palabra compuesta por freak (*anormal o raro*), free (*libre*), y X (*haciendo alusión a Unix*). Durante el inicio de su trabajo sobre el sistema, él almacenó los archivos bajo el nombre 'Freax' por aproximadamente medio año. Torvalds ya había considerado el nombre 'Linux', pero al principio lo había descartado por ser demasiado egocéntrico o egoísta.

Para darle a otra gente la oportunidad de cooperar en el sistema o también para sugerir mejoras, los archivos fueron colocados en el servidor ftp (*ftp.funet.fi*) de la Universidad de Tecnología de Helsinki (*Helsinki University of Technology*) (*HUT*), en septiembre de 1991.

Ari Lemmke, colega de Linus Torvalds en la HUT, que en ese entonces era responsable de los servidores, no estuvo de acuerdo con el nombre 'Freax', prefiriendo el nombre 'Linux'. Él simplemente renombró los archivos colocados sobre el servidor con el nombre 'Linux', sin consultar a Linus Torvalds. Más tarde, sin embargo, Linus Torvalds accedió a usar el nombre 'Linux'.





1.4. Controversia por el nombre (*LINUX* vs *GNU/Linux*).

La contribución de GNU es la razón por la que existe controversia a la hora de utilizar Linux o GNU/Linux para referirse al sistema operativo formado por las herramientas de GNU y el núcleo Linux en su conjunto.

A pesar de que Linux es, en sentido estricto, el sistema operativo, parte fundamental de la interacción entre el núcleo y el usuario (*o los programas de aplicación*) se maneja usualmente con las herramientas GNU. Sin embargo, una parte significativa de la comunidad, así como muchos medios generales y especializados, prefieren utilizar el término Linux para referirse a la unión de ambos proyectos.

GNU/Linux es el término promovido por la Free Software Foundation (*FSF*), y su fundador Richard Stallman, para el sistema operativo que incluye las utilidades de la FSF y el núcleo Linux. Entre otros, notablemente Debian GNU/Linux usa este término en su nombre, pero existe una tendencia a usar el término '*Linux*' para referirse a todo el sistema operativo.

La FSF argumenta el uso del término GNU/Linux porque GNU fue un proyecto de larga trayectoria para desarrollar un sistema operativo libre, del cual el núcleo sólo fue una pieza. Los defensores del término '*Linux*' argumentan que los usuarios y los desarrolladores que han escogido de forma notable mantener este nombre se debe a que es más corto, aunque reconocen que GNU es el más grande contribuyente.

Un sistema operativo basado en GNU/Linux tiene muchos componentes, incluyendo el núcleo Linux y los programas desarrollados, en su mayoría por el proyecto GNU.

Algunas personas objetan que el nombre '*Linux*' debería usarse para referirse solamente al núcleo, no a todo el sistema operativo.





La gente que insiste en el término GNU/Linux quisiera que la FSF consiguiera la mayor parte del crédito porque Richard Stallman y sus amigos escribieron muchas de sus herramientas. Ni esta teoría ni el término GNU/Linux ha ganado más que la aceptación de la minoría. Tal vez Stallman está frustrado porque Torvalds obtuvo la gloria que él quería tener.

Lo cierto es que, no siempre es correcto usar el nombre GNU/Linux, esto debido a que, es posible instalarlo en una amplia variedad de hardware; tanto servidores como en computadoras de escritorio y portátiles, en computadoras de bolsillo, teléfonos móviles, dispositivos empotrados, videojuegos, etc.; para todas éstas opciones puede darse el caso en que las partes de GNU se reemplacen por alternativas más adecuadas.

1.5. Free Software Foundation.

La Free Software Foundation (*FSF*), creada por Richard Stallman y otros entusiastas del software libre en 1985, es una organización sin ánimo de lucro cuyo objetivo es promocionar la libertad y defender los derechos de los usuarios de software libre.

A medida que nuestra sociedad se hace más dependiente de las computadoras, el software que se ejecuta es de importancia crítica. El software libre trata de tener presencia en la tecnología que utilizamos en nuestros hogares, escuelas y negocios, donde se busca tanto el beneficio individual como el comunitario.

La Free Software Foundation está trabajando para asegurar la libertad de los usuarios de computadoras, promoviendo el desarrollo y uso de software libre (*en particular el sistema operativo GNU*), haciendo campaña contra las amenazas a la libertad informática del usuario, como la Gestión Digital de Restricciones (*DRM*).





1.6. Filosofía del software libre.

El '*software libre*' es, ante todo, una forma ética de entender el software o los programas de computadora. Eso incluye tanto la fabricación de los programas como su distribución y su utilización. Por lo tanto, el software libre no está orientado únicamente a (*ni es patrimonio exclusivo de*) la gente capaz de escribir un programa o a la gente que posee empresas y por tanto está en condiciones de hacer negocio con los programas.

Entonces, el software libre tiene que ver con el usuario tanto como con los desarrolladores o los comerciantes.

Así que el software libre como concepto y el movimiento que lo hace crecer y evolucionar es un concepto y un movimiento ético.

Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Se da ésta definición de software libre para mostrar claramente qué debe cumplir un programa de software en concreto para que se le considere software libre. De modo más preciso, se refiere a cuatro derechos o libertades de los usuarios del software:

- ✓ La libertad de usar el programa, con cualquier propósito.
- ✓ La libertad de estudiar el funcionamiento del programa y adaptarlo a nuestras necesidades.
- ✓ La libertad de distribuir copias, con lo que puede ayudar a otros usuarios.
- ✓ La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie.





El derecho a utilizar. El primer derecho o libertad, es el que trata sobre el derecho a utilizar software. Por ejemplo, ¿acaso no tiene la gente derecho a utilizar un programa que ha comprado en una tienda? La respuesta es que cuando una persona *'compra'* un programa de computadora que no es software libre (*programas privados*) por lo general no dispone del derecho de utilización ilimitada (*que debería ser esencial*): el usuario está limitado a utilizar el programa para determinados objetivos (*prohibido usar este programa de forma comercial*) o en determinados sitios (*prohibido usar este programa en el país X y el país Y*) o en un número determinado de máquinas (*prohibido usar este programa en más de una máquina al mismo tiempo*). Estas restricciones son muy habituales cuando hablamos de software privado.

El derecho a entender. Por otra parte tenemos la segunda libertad para el usuario: el derecho a entender cómo funcionan los programas que nos distribuyen y adaptarlo a nuestras necesidades. Nuevamente, este derecho fundamental se ve vulnerado cuando de software privado hablamos: por lo general, el software privado se distribuye en forma de ejecutables (*equivalentes a los archivos '.exe' en entornos Windows*) sin que le acompañe el código fuente correspondiente.

El código fuente de un programa es su forma entendible y modificable por un programador. En cierto modo, es como el manual de instrucciones de algún aparato.

Si compramos un control remoto, ajustado de fábrica a nuestro televisor, y no viene con la opción de modificarlo, no seremos capaces de utilizarlo en otro de nuestros televisores, el de la cocina, por ejemplo. En eso consiste adaptar los artefactos que utilizamos a nuestras necesidades de la vida diaria. En el caso que nos ocupa, el de los programas de computadora, para ello necesitamos forzosamente el código fuente.





El derecho a compartir. Vamos ahora al tercer derecho o libertad: el derecho a distribuir programas de computadora de forma gratuita o, alternativamente, cobrando algo a cambio de hacerlo. Este es tal vez el punto que mejor comprenden los usuarios de programas privativos cuando quieren aprender qué es el software libre. Es natural, ya que la industria del software privativo hace continuos esfuerzos para intentar convencer a la sociedad de que copiar programas de cómputo es algo que no debe hacerse.

El poder ser copiado sin necesidad de grandes recursos (*con una unidad de grabación basta*) y la característica peculiar de que la copia no pierde calidad respecto al original no es algo malo: por el contrario, es casi lo mejor que tiene el software libre. Realizar copias de programas privativos es algo ilegal en la mayoría de los países. Por eso existe el software libre: es perfectamente legal copiarlo y distribuirlo. De esta forma tanto el usuario como la sociedad se benefician y nadie sale perdiendo (*la copia original no funciona peor por haber hecho una o millones de copias*).

Es importante un detalle: **el software libre no tiene por qué ser gratis.** Es perfectamente posible distribuir software libre a cambio de dinero. Así es como pueden ganarse la vida los programadores y distribuidores. Ahora bien, eso no justifica el hecho de vulnerar los derechos de la gente que paga por obtener una copia del programa: **el usuario puede distribuir sus propias copias, cobrando por ello si lo desea.**

El derecho a mejorar. El usuario de software privativo no puede mejorar los programas que utiliza: aunque quiera y sepa hacerlo, por lo general no tiene acceso al código fuente. Y aunque lo tuviera (*puede distribuirse el código fuente y no obstante no ser software libre*) sería ilegal modificar ese código fuente.

El software libre siempre se distribuye con su código fuente y además es totalmente legal modificarlo.





La objeción típica a este derecho suele ser: *"bien, tengo el código fuente. Pero YO no sé modificarlo para mejorarlo, de qué me sirve entonces tenerlo"*. La respuesta a esa pregunta (*muy habitual y muy comprensible*) es que para mejorar o modificar un programa no es necesario hacerlo uno mismo. De la misma forma que una persona puede llevar su coche a un taller a que se lo reparen (*o se le introduzcan algunas mejoras*) y además puede hacerlo al taller que más le guste (*el fabricante del coche no puede poner restricciones*) el usuario de software debe poder contratar a un profesional (*o a una empresa o pedir a su primo el informático que lo haga gratis*) que mejore el software libre que ha comprado o que ha descargado de internet.

Si el mismo usuario compra o descarga software privativo y quiere mejorarlo, entonces depende totalmente de la compañía que le vendió el programa. Si la compañía decide que las mejoras deseadas por el usuario no son rentables o no son convenientes por una u otra razón, no las incorporará al programa. El resultado es que con software privativo el usuario siempre pierde.

Con software libre, el usuario es libre y no está atado a ninguna compañía. El usuario puede distribuir el programa que incorpora sus mejoras gratuitamente o por un pago.

Y otra cosa importante: el usuario también tiene derecho a no distribuir sus mejoras si no quiere. Puede descargar o comprar software libre, introducirle mejoras y no redistribuir ni hacer públicas dichas mejoras.

Cualquier persona u organización tiene la libertad de ejecutar libremente el programa, de usarlo en cualquier tipo de sistema de computación, para cualquier tipo de trabajo y propósito, sin estar obligado a comunicarlo a su programador o alguna otra entidad específica.





El propósito de los usuarios es el que importa, no el propósito de los programadores. Como usuario es libre de ejecutar un programa para sus propósitos, modificarlo como le convenga; y si lo distribuye a otra persona, también esa persona es libre para ejecutarlo para sus propósitos, pero usted no tiene derecho a imponerle sus propios propósitos.

Para que estas libertades puedan ser reales, deben ser irrevocables; si el programador del software tiene el poder de revocar la licencia, o de cambiar retroactivamente sus términos, el software no es libre.

Ciertos tipos de reglas sobre la manera de distribuir software libre son aceptables, cuando no entran en conflicto con las libertades principales. Por ejemplo, el '*copyleft*' (*definido muy resumidamente*), es la regla en base a la cual, cuando redistribuye el programa, no puede agregar restricciones para denegar a las demás personas las libertades principales.

Esta regla no entra en conflicto con las libertades principales; más bien las protege. En el proyecto GNU se usa el '*copyleft*' para proteger legalmente estas libertades para todos.

Se debe tener en cuenta que los criterios, como los establecidos en esta definición de software libre, requieren pensar con cuidado su interpretación. Para decidir si una licencia de software específica es una licencia de software libre, se juzga en base a estos criterios para determinar si concuerda su espíritu, conjuntamente con la terminología precisa. Si una licencia incluye restricciones demasiado grandes, es rechazada. Algunas veces, los requisitos de una licencia muestran una cuestión que hace necesaria una reflexión más profunda, incluyendo la discusión con un abogado, antes que se pueda decidir si el requisito es aceptable. Cuando llegan a una conclusión sobre una nueva cuestión, suelen actualizar estos criterios para que resulte más fácil ver por qué ciertas licencias se califican o no.





Por otro lado, mucha gente cree que el espíritu del software libre es que no se debería cobrar dinero por distribuir copias de software o que se debería cobrar lo menos posible, o sólo lo suficiente para cubrir el costo. Esto es un malentendido. En realidad, se recomienda a la gente que distribuye software libre que cobre tanto como desee.

En inglés, la palabra '*free*' tiene dos significados generales válidos. Puede significar tanto '*libre*' como '*gratis*'. Cuando hablamos de software libre (*o free software en inglés*), estamos hablando de libertad, no de precio.

En algunos casos los programas libres son distribuidos gratuitamente y en otras ocasiones por un precio muy alto. A veces, el mismo programa se puede conseguir de ambos modos de fuentes distintas. El programa es libre a pesar del precio, porque los usuarios tienen libertades ya mencionadas.

Los programas que no son libres habitualmente se venden a un precio alto, pero en ocasiones alguna tienda le dará una copia sin costo. Sin embargo, eso no lo hace software libre. Con precio o sin él, el programa no es libre si los usuarios no tienen libertad.

Como el software libre no tiene nada que ver con el precio, un precio bajo no lo hace más libre o que esté más cerca de serlo. Entonces, si está redistribuyendo copias de software libre, podría poner un precio y ganar algo de dinero. Redistribuir software libre es una actividad buena y legítima. Si lo hace, es correcto que obtenga un beneficio de ella.

El software libre es un proyecto comunitario y todo aquel que dependa del proyecto debería buscar formas de contribuir a construir la comunidad.

Para un distribuidor, la mejor forma de contribuir al desarrollo de software es donar parte del beneficio a la Fundación para el Software Libre (*FSF*), u otro proyecto de desarrollo de software libre.





Distribuir software libre es una oportunidad de obtener fondos para el desarrollo. Financiando el desarrollo, puede mejorar el mundo del software libre.

Para poder contribuir con fondos, se necesita tener un beneficio. Si se pone un precio demasiado bajo, no sobrará nada para contribuir al desarrollo.

Salvo por una situación especial, la Licencia Pública General de GNU (*GPL de GNU*) no establece restricciones acerca de cuánto puede cobrar uno por distribuir una copia de software libre. Podemos no cobrar, cobrar un centavo, un peso, o mil millones de dólares. Eso depende de uno y del mercado, por ende no hay que quejarnos si nadie quiere pagar mil millones de dólares por una copia.

1.7. Distribuciones.

Una distribución Linux, o distribución GNU/Linux (*coloquialmente llamadas distros*), es una distribución de software basada en el núcleo Linux, que incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores. Por lo general están compuestas, total o mayoritariamente, de software libre, aunque a menudo incorporan aplicaciones o controladores propietarios.

Además del núcleo Linux, las distribuciones incluyen habitualmente las bibliotecas y herramientas del proyecto GNU. Dependiendo del tipo de usuarios a los que la distribución esté dirigida se incluye también otro tipo de software como:

- ✓ Herramientas de oficina (procesador de textos, hoja de cálculo, etc).
- ✓ Reproductores de audio y video.
- ✓ Herramientas administrativas, etc.





Existen distribuciones que están soportadas por empresas grandes como: Fedora (*Red Hat*), openSUSE (*Novell*), Ubuntu (*Canonical Ltd.*), Mandriva (*Mandriva Soft*); y distribuciones mantenidas por la comunidad como Debian y Gentoo.

Antes de que surgieran las primeras distribuciones, un usuario de Linux debía ser algo experto en Unix; no sólo debía conocer qué bibliotecas y ejecutables necesitar para iniciar el sistema y que funcionara, sino también los detalles importantes que se requieren en la instalación y configuración de los archivos en el sistema.

Las distribuciones GNU/Linux comenzaron a surgir poco después de que el núcleo Linux fuera utilizado por otros programadores además de los creadores originales.

Entre las distribuciones más antiguas se incluían:

- ✓ Dos discos denominados H J Lu's 'Boot-root', con el núcleo y un mínimo de herramientas para utilizar.
- ✓ MCC Interim Linux, primera distribución Linux formal, que se podía descargar desde un servidor público FTP de la Universidad de Mánchester, en febrero de 1992.
- ✓ TAMU, creado por entusiastas de la Universidad de Texas A&M al mismo tiempo que SLS.
- ✓ SLS (Softlanding Linux System).
- ✓ Yggdrasil Linux creó el primer CD-ROM de una distribución GNU/Linux.

La distribución SLS (Softlanding Linux System) no estuvo bien mantenida; así pues el informático estadounidense, Patrick Volkerding lanzó una distribución basada en SLS a la que llamó '*Slackware*'; lanzada el 16 de julio de 1993. Ésta es la distribución más antigua que está actualmente en desarrollo activo.





Los usuarios vieron en Linux una alternativa a los sistemas operativos DOS y Microsoft Windows en la plataforma PC; Mac OS en Apple Macintosh y las versiones de uso bajo licencia (*de pago*) de UNIX. La mayoría de estos primeros usuarios se habían familiarizado con el entorno UNIX en sus trabajos o centros de estudios. Estos adoptaron GNU/Linux por su estabilidad, reducido (*o nulo*) coste y por la disponibilidad del código fuente del software incluido.

Las distribuciones eran originalmente una cuestión de comodidad para el usuario medio, evitándole la instalación (*y en muchos casos compilación*) por separado de paquetes de uso común, pero hoy se han popularizado incluso entre los expertos en éste tipo de sistemas operativos (*UNIX - Linux*).

Si bien, históricamente, Linux estuvo mejor posicionado en el mercado de los servidores, aparecieron distribuciones que se centraron en hacer un sistema de fácil instalación y uso en computadoras personales, tales como: Mandriva, OpenSuse y Ubuntu, entre otras.

Gran parte del software incluido es software libre y distribuido tanto en binario compilado como en forma de código fuente. Algunas distribuciones incorporan software privativo, no disponible en forma de código fuente.

Muchas distribuciones proveen un sistema de instalación gráfica como lo hacen otros sistemas modernos. Distribuciones independientes como Gentoo Linux y Linux From Scratch proveen el código fuente de todo el software y solo incluyen los binarios del núcleo, herramientas de compilación y de un instalador; el instalador compila todo el software para el CPU específico de la PC del usuario.

Las distribuciones están divididas en '*paquetes*'. Cada paquete contiene una aplicación específica o un servicio. Ejemplos de paquetes son una librería para manejar el formato de imagen con extensión PNG, una colección de tipografías o un navegador web.





El paquete es generalmente distribuido en su versión compilada y la instalación y desinstalación de los paquetes es controlada por un sistema de gestión de paquetes. Cada paquete elaborado para ese sistema de paquetes contiene meta-información tal como fecha de creación, descripción del paquete y sus dependencias. El sistema de paquetes analiza esta información para permitir la búsqueda de paquetes, actualizar las librerías y aplicaciones instaladas, revisar que todas las dependencias se cumplan y obtenerlas de manera automática, si no se cuenta con ellas.

Aunque las distribuciones casi siempre vienen con una gran cantidad de software, los usuarios pueden instalar software que no fue incluido en la distribución. Un ejemplo podría ser el instalar una versión experimental, o de prueba, de alguna de las aplicaciones de la distribución o alguna alternativa a esa aplicación que viene por default.

La mayor parte de las distribuciones instalan los paquetes, incluyendo el núcleo Linux y otras piezas fundamentales del sistema operativo con una configuración preestablecida. Esto hace la instalación más sencilla, especialmente para los usuarios nuevos, pero no es siempre aceptable, pues hay programas que deben de ser cuidadosamente configurados para que sean funcionales, para que operen correctamente con otra aplicación o para que su seguridad sea robusta. En estos casos, los administradores se ven obligados a invertir tiempo reconfigurando y revisando software soportado por la distribución.

Hay distribuciones en que la instalación puede llegar a ser muy lenta, pues es posible ajustar y configurar la mayor parte o la totalidad del software incluido en la distribución. No en todas es posible. Algunas ofrecen herramientas de configuración para ayudar en el proceso.

Es también posible armar un sistema a la medida en su totalidad, descartando incluso el uso de una distribución.





Lo primero que hay que hacer es generar un sistema base que permita conseguir, compilar, configurar e instalar el código fuente.

En general, las distribuciones Linux se pueden clasificar en:

- ✓ Comerciales o no comerciales.
- ✓ Ser completamente libres o incluir software privativo.
- ✓ Diseñadas para uso en el hogar o en las empresas.
- ✓ Diseñadas para servidores, escritorios o dispositivos empujados.
- ✓ Orientadas a usuarios regulares o usuarios avanzados.
- ✓ De uso general o para dispositivos altamente especializados, como un cortafuegos, un enrutador o un cluster computacional.
- ✓ Diseñadas e incluso certificadas para un hardware o arquitectura específicos.
- ✓ Orientadas hacia grupos en específico, por ejemplo a través de la internacionalización y localización del lenguaje, o por la inclusión de varios paquetes para la producción musical o para computación científica.
- ✓ Soportadas bajo distintos tipos de hardware.

La diversidad de las distribuciones Linux es debido a cuestiones técnicas, de organización y puntos de vista diferentes entre usuarios y proveedores. El modo de licenciamiento del software libre permite a cualquier usuario con conocimientos e interés suficiente pueda adaptar o diseñar una distribución de acuerdo a sus necesidades. Hay distribuciones que no requieren instalación (*Live CD*).

Una distribución live o Live CD o Live DVD, es una distribución almacenada en un medio extraíble, tradicionalmente un CD o un DVD (*de ahí sus nombres*), que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro de una computadora, para lo cual usa la memoria RAM como disco duro virtual y el propio medio como sistema de archivos.





Este tipo de distribuciones son ideales para ser utilizadas en demostraciones, operaciones de recuperación, cuando se utiliza una máquina ajena o como medio de instalación para una distribución estándar. Actualmente, casi todas las distribuciones tienen una versión CD/DVD autónomo o 'vivo'.

La mayoría de las distribuciones están, en mayor o menor medida, desarrolladas y dirigidas por sus comunidades de desarrolladores y usuarios.

En muchas ciudades y regiones, asociaciones locales conocidas como grupos de usuarios promueven este sistema operativo y el software libre. Suelen ofrecer conferencias, talleres o soporte técnico de forma gratuita o introducción a la instalación de GNU/Linux para nuevos usuarios.

En las distribuciones son muy comunes las salas de chat IRC, grupos de noticias y los foros; para el soporte a usuarios; las listas de correo suelen ser el medio principal para discutir sobre el desarrollo.

Distribuciones que se especializan en grupos específicos:

- ✓ OpenWrt, diseñada para ser empotrada en dispositivos ruteadores.
- ✓ Edubuntu, un sistema del proyecto Ubuntu, diseñado para entornos educativos.
- ✓ MythTV, orientada para equipos multimedia.
- ✓ Musix, es una distribución destinada a los músicos.
- ✓ mkLinux, Yellow Dog Linux o Black Lab Linux, orientadas a usuarios de Macintosh y de la plataforma PowerPC.
- ✓ 64 Studio, una distribución basada en Debian, diseñada para la edición multimedia.
- ✓ ABC GNU/Linux, distribución para la construcción de clusters Beowulf desarrollado por Iker Castaños Chavarri, Universidad del País Vasco (*España*).





- ✓ BackTrack, es una distribución basada en Ubuntu, con una colección de herramientas de software forenses y de seguridad.
- ✓ Parted Magic, es un live cd/usb cuyo propósito fundamental es la partición de Discos Duros; también ofrece herramientas para la creación de imágenes de CD, pruebas a Discos Duros, rescate de datos; etc.
- ✓ Clonezilla Live, es una distribución live CD basada en Debian, está orientada a la clonación de Discos Duros para respaldo, así como la restauración de éstos.
- ✓ LinuX-gamers Live DVD, es un DVD que puede arrancar al encender la computadora. Esta basada en Arch Linux, con una gran colección de juegos 3D, contiene drivers propietarios de las tarjetas NVIDIA y ATI.

Distribuciones más importantes:



Fig. 1.7.1. Logo

Debian GNU / Linux (Fig. 1.7.1. Logo). Fue anunciada por primera vez en 1993. Su fundador, Ian Murdock, preveía la creación de un proyecto totalmente no comercial desarrollado por cientos de desarrolladores voluntarios en su tiempo libre. Superando en número a los escépticos y optimistas en ese momento. Mucha gente pensaba que estaba destinada a desintegrarse, pero la realidad era muy diferente. El proyecto

Debian (Fig. 1.7.2. Debian GNU/Linux 5.0, con escritorio GNOME) no sólo sobrevivió, sino que prosperó y, en menos de una década, se convirtió en la mayor distribución de Linux.

El éxito de Debian puede ilustrarse por las siguientes cifras:





Es desarrollado por más de 1000 voluntarios, sus repositorios de software contienen más de 20000 paquetes (*compilado para 11 arquitecturas de procesador*), y es responsable de inspirar a más de 120 distribuciones basadas en Debian.



Fig. 1.7.2. Debian GNU/Linux 5.0

Éstas cifras son incomparables por cualquier otro sistema operativo basado en Linux. El desarrollo actual de Debian se realiza en tres ramas principales: *'inestable'*, *'pruebas'* y *'estable'*.

Esta integración progresiva y la estabilización de paquetes y características, junto con mecanismos bien establecidos del proyecto de control de calidad, han hecho que se hayan ganado su reputación.

Sin embargo, este estilo de desarrollo largo y complejo también tiene algunos inconvenientes: las versiones estables de Debian no son particularmente puestas al día y envejecen rápidamente, las nuevas versiones estables se publican cada 1 a 3 años. Aquellos usuarios que prefieren los últimos paquetes y tecnologías se ven obligados a utilizar la rama de *'prueba'* o *'inestable'* de Debian.

Pros. Muy estable, estricto control de calidad, incluye alrededor de 20000 paquetes de software, soporta más arquitecturas de procesador que alguna otra distribución Linux.





Contras. Conservadora, debido a su soporte para varias arquitecturas de procesador, las tecnologías más actuales no son siempre incluidas en la versión estable, largo ciclo de liberación de versiones estables.

Manejo de paquetes de software. Advanced Package Tool (*APT*) usando paquetes DEB.

Alternativas sugeridas, basadas en Debian. MEPIS Linux, Ubuntu, Sidux, KNOPPIX (*live DVD*), Dreamlinux, Elive (*escritorio con Enlightenment*), Xandros (*commercial*), 64 Studio (*multimedia*).



Red Hat (*Fig. 1.7.3. Logo.*). Red Hat es el líder en el desarrollo, implementación y administración de soluciones Linux y de código abierto para la infraestructura de Internet, que van desde dispositivos embebidos a servidores Web seguros.

Fig. 1.7.3. Logo.

Sus orígenes se remontan a 1994 cuando fue lanzado por dos visionarios Linux, Bob Young y Marc Ewing, bajo el nombre de Red Hat Linux (*Fig.1.7.4. Red Hat Enterprise Linux 5.5, con escritorio GNOME*). El código abierto es la base de su modelo de negocio.

El primer producto de la compañía, Red Hat Linux 1.0, fue lanzado en el mismo año y fue seguida rápidamente por varias actualizaciones de corrección de errores de programación. En 1997, Red Hat introdujo su revolucionario sistema de gestión de paquetes RPM con resolución de dependencias y otras características avanzadas lo que contribuyó al rápido crecimiento de la popularidad de la distribución.





En 2003, justo después del lanzamiento de Red Hat Linux 9, la compañía hizo algunos cambios radicales a su línea de productos. Conservó la marca Red Hat para sus productos comerciales, especialmente el de Red Hat Enterprise Linux, e introdujo una nueva distribución, Fedora Core, patrocinada por Red Hat, pero



orientada a la comunidad, diseñada para el aficionado Linux.

Después de las críticas iniciales de los cambios, la comunidad Linux aceptó la 'nueva'

Fig.1.7.4. Red Hat Enterprise Linux 5.5

distribución como una continuación lógica de Red Hat Linux. Al mismo tiempo, Red Hat se convirtió rápidamente en la mayor y más rentable empresa de Linux en el mundo, con un innovador producto de iniciativas interesantes, tales como el programa de certificación Red Hat Certified Engineer (RHCE).

Es una de las distribuciones más innovadores de la actualidad. Sus contribuciones al kernel de Linux, glibc (*la biblioteca estándar de lenguaje C de GNU*) y gcc (*GNU Compiler Collection, colección de compiladores GNU*) son bien conocidas y su integración a otras características a nivel de empresa son muy apreciados entre los clientes de la compañía.

Las soluciones de Red Hat combinan capacitación, servicios de gestión, apoyo técnico; etc. Por un lado negativo, carece todavía de una estrategia clara orientada al escritorio, para que el producto sea más fácil de usar por los usuarios.

Pros. Altamente innovadora, características excepcionales de seguridad, gran número de paquetes soportados; estricta adherencia a la filosofía del software libre.





Contras. Las prioridades de Red Hat tienden a inclinarse hacia las funciones de empresa, en lugar de facilidad en el uso de escritorio.

Manejo de paquetes de software. YUM gráfico y utilidad de línea de comandos utilizando paquetes RPM.

Ediciones disponibles. Red Hat Enterprise Linux para i386, IA64, PowerPC, y arquitecturas x86_64.

Alternativas sugeridas, basadas en Red Hat. CentOS, Scientific Linux.



Fig. 1.7.5. Logo

Ubuntu (Fig. 1.7.5. Logo). Es la distribución más popular, orientada al escritorio. Está dirigida a los usuarios noveles en Linux (Fig. 1.7.6. Ubuntu 10.04, con escritorio GNOME). El lanzamiento de Ubuntu fue anunciado por primera vez en septiembre de 2004. A pesar de ser un relativo recién llegado al

escenario de las distribuciones de Linux, el proyecto despegó como ninguna otra antes; sus listas de correo pronto se llenaron con discusiones de usuarios y desarrolladores. Ha contribuido notablemente al desarrollo de un

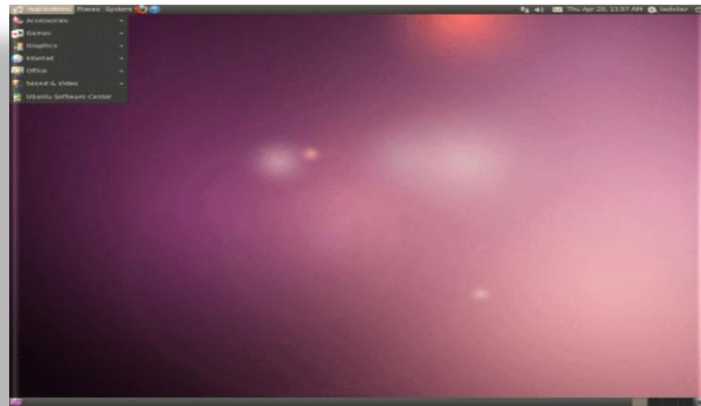


Fig. 1.7.6. Ubuntu 10.04

sistema, basado en Linux, fácil de utilizar, que puede competir muy bien con cualquier sistema operativo disponible en el mercado.





Algunas de las razones del éxito de Ubuntu: En primer lugar, el proyecto fue creado por Mark Shuttleworth, millonario Sudáfricano, que fue el segundo turista espacial del mundo.

El proyecto es financiado por su compañía Canonical Ltd, establecida en la Isla de Man. En segundo lugar, Ubuntu ha aprendido de los errores de otros proyectos similares y se enfocó en los usuarios finales. En tercer lugar, gracias a su fundador, Ubuntu ha sido capaz de enviar CD's gratis para todos los usuarios interesados, contribuyendo así a la rápida propagación de la distribución.

En el aspecto técnico, está basado en Debian '*Sid*' (*rama inestable*), con algunos paquetes importantes, como GNOME, actualizados a sus últimas versiones.

Tiene un calendario de lanzamientos preestablecido (*cada seis meses*), con una versión ocasional de soporte a largo plazo (*LTS*), versión que es compatible con las actualizaciones de seguridad para 3 a 5 años, dependiendo de la edición. Otras características especiales de Ubuntu incluyen un Live CD instalable, apariencia muy original, asistente de migración para usuarios de Windows, soporte para las últimas tecnologías, tales como los efectos de escritorio 3D, fácil instalación de controladores para las tarjetas gráficas ATI y NVIDIA, así como para redes inalámbricas, y el apoyo para la instalación de códecs (*abreviatura de codificador - decodificador*) multimedia no libres.

Pros. Ciclo de liberación de versiones oficiales preestablecidas y periodo de soporte largo; '*amigable*' para el usuario final; documentación oficial y contribuida por los usuarios.

Contras. Carece de compatibilidad con Debian, en el sentido que es difícil instalar los paquetes de Ubuntu en la versión estable de Debian.

Manejo de paquetes de software. Advanced Package Tool (*APT*) usando paquetes DEB.





Ediciones disponibles. Ubuntu, Kubuntu, Xubuntu, Ubuntu Studio y Mythbuntu para 32-bit (i386) y 64-bit (x86_64); Ubuntu Server Edition, también para procesadores SPARC.

Alternativas sugeridas, basadas en Ubuntu. gOS (escritorio con aplicaciones de Google), OpenGEU (escritorio con Enlightenment), Ultimate Edition (escritorio), CrunchBang Linux (escritorio con Openbox), gNewSense (sólo software libre).



Fig. 1.7.7. Logo

nombre de SuSE.

Al comienzo, la naciente empresa solía vender discos informáticos los cuales contenían la versión alemana de Slackware Linux, pero no pasó mucho tiempo hasta que se transformara en una distribución independiente a través de su versión 4.2 lanzada en mayo de 1996.

OpenSuse (Fig. 1.7.7. Logo). El origen de openSUSE se remonta al año 1992, cuando cuatro entusiastas usuarios de Linux, los alemanes Roland Dyroff, Thomas Fehr, Hubert Mantel y Burchard Steinbild, lanzaron el proyecto bajo el



Fig. 1.7.8. openSUSE 11.3

En los siguientes años, los desarrolladores adoptaron el formato de paquetes RPM e introdujeron YaST, una amigable herramienta gráfica de administración.





Cuenta con lanzamientos frecuentes, excelentes publicaciones y documentación, y una amplia disponibilidad de SuSE Linux en tiendas a lo largo de Europa y norte América, esto le ha significado un aumento en la popularidad de la distribución.

SuSE fue comprado a finales del 2003 por Novell. Inc. Amplios cambios en el desarrollo, de licencia y disponibilidad de SUSE Linux fueron aplicados casi de inmediato. YaST fue lanzado bajo Licencia General Pública, la imágenes ISO fueron distribuidas libremente desde servidores públicos y lo más significativo fue que el desarrollo de la distribución fue por primera vez abierto para todo público (*Fig. 1.7.8. openSUSE 11.3, con escritorio KDE.*).

Desde el lanzamiento del proyecto openSUSE y la versión 10.0, en octubre del 2005, la distribución ha llegado a ser completamente libre en todo sentido.

El código de openSUSE ha llegado a ser el sistema base de los productos comerciales de Novell, primeramente llamado Novell Linux, pero renombrado más adelante como SUSE Linux Enterprise Desktop y SUSE Linux Enterprise Server.

Las principales razones por las cuales openSUSE obtiene una alta puntuación son sus entornos de escritorio agradables y refinados (*KDE y GNOME*), un excelente sistema de administración de aplicaciones (*YaST*) y, para aquellos que adquieren la edición pagada, se adjuntan las mejores publicaciones disponibles.

Desafortunadamente, el reciente acuerdo entre Novell y Microsoft, el cual aparentemente concede a Microsoft, derechos de propiedad intelectual sobre Linux, ha dado como resultado una serie de condenas por parte de muchas personalidades asociadas a Linux y ha hecho que muchos usuarios cambien de distribución.





Aunque Novell ha minimizado el tema del negocio y Microsoft no ha ejercido ningún derecho aún, el tema continua siendo una espina para la comunidad Linux de esta compañía.

Pros. Cuenta con herramientas de configuración intuitivas y seguras; numerosos paquetes de repositorios y un sitio web con excelente infraestructura.

Contras. El convenio establecido entre Novell y Microsoft en noviembre de 2006 el cual aparentemente legaliza derechos de propiedad intelectual de Microsoft sobre Linux; la instalación de su pesado escritorio y las aplicaciones gráficas es a veces percibido como '*saturado y lento*'.

Manejo de paquetes de software. Aplicación gráfica y línea de comando YaST para paquetes RPM.

Ediciones disponibles. openSUSE para 32 bit (i386), 64 bit (x86_64) y procesadores PowerPC (ppc); SUSE Linux Enterprise Desktop/Server for i586, IA64, PowerPC.



Fig. 1.7.9. Logo

Mandriva (Fig. 1.7.9. Logo). Mandriva Linux fue lanzada por Gaël Duval, en julio de 1998, bajo el nombre de Mandrake Linux. Al principio, era sólo una edición remasterizada de Red Hat Linux, con el escritorio KDE, más fácil de usar; pero las versiones posteriores también agregaron varias novedades propias, tales como un nuevo instalador, la detección de hardware mejorada y una utilidad intuitiva para crear particiones de disco. Como resultado de estas mejoras, Mandrake Linux floreció. Después la fortuna de la recientemente establecida MandrakeSoft fluctuó ampliamente al borde de la quiebra a principios de 2003 a una oleada de adquisiciones en 2005.





Finalmente, se fusionó con la empresa Conectiva de Brasil, lo cuál originó que la compañía cambiará su nombre a Mandriva (Fig. 1.7.10. Mandriva Linux 2010.1, con escritorio KDE.).

Mandriva Linux es fundamentalmente una distribución de escritorio. Totalmente orientada a los usuarios noveles que quieren hacer una transición desde otros sistemas operativos.



Fig. 1.7.10. Mandriva Linux 2010.1

Sus características más apreciadas son la facilidad de uso, la actualización constante de sus repositorios de software, un excelente administrador del sistema y un amplio soporte en la internacionalización.

Tiene un modelo de desarrollo abierto, mucho antes que otras distribuciones populares, con las pruebas beta intensas y frecuentes lanzamientos estables.

En los últimos años, también ha desarrollado una gama de 'live CDs' instalables y ha lanzado Mandriva Flash, un completo sistema Mandriva Linux en un dispositivo USB de arranque. Fue la primera distribución importante que ofrece ayuda para la instalación en las populares netbooks, como el ASUS Eee PC. A pesar de la excelencia técnica, Mandriva Linux ha pasado por muchas altas y bajas en los últimos años, esto tiene que ver en parte con la aparición de otras distribuciones fáciles de usar, que han alcanzado a Mandriva en ese aspecto; pero también con algunas decisiones polémicas que han ocasionado el distanciamiento de un sector de la base de usuarios de la distribución.





Su '*Mandriva Club*', originalmente diseñado para proporcionar un valor agregado a clientes de pago, ha estado recibiendo muchas críticas. La compañía ha abordado algunas de las críticas y sigue enfrentándose a una dura batalla para convencer a los nuevos usuarios de Linux, o usuarios de otras distribuciones, para probar (y *comprar*) sus productos.

Pros. Excelente distribución para usuarios sin experiencia en Linux, especialmente la edición comercial. Una utilidad de configuración central muy fácil de usar, soporte en línea, disponible en una amplia variedad de idiomas.

Contras. Carece de una estrategia de marketing global para competir con otras grandes distribuciones.

Manejo de paquetes de software. URPMI con Rpmrake (*una interfaz gráfica para URPMI*) usando paquetes RPM.

Ediciones disponibles. Mandriva '*Free*', edición con sólo software libre, se puede descargar para arquitecturas de 32 bits (i586) y 64 bits (x86_64); Mandriva '*One*', edición con controladores '*no libres*', instalable en arquitecturas de 32 bits (i586) y 64 bits (x86_64); la edición comercial Mandriva '*PowerPack*', con controladores y aplicaciones '*no libres*', para 32 bits (i586) y 64 bits (x86_64).

Alternativas sugeridas, basadas en Mandriva. PCLinuxOS.



Fig. 1.7.11. Logo

Slackware (Fig. 1.7.11. Logo). Slackware Linux, creado por Patrick Volkerding en 1992, es la más antigua distribución de Linux. Creada a partir del proyecto SLS (*ahora suspendido*), Slackware 1.0 apareció en 24 disquetes y fue construido sobre la versión del núcleo Linux 0.99pl11-alfa. Rápidamente se convirtió en la distribución Linux más popular, con algunas estimaciones poniendo su cuota de mercado de hasta el 80% de todas las instalaciones de Linux en 1995.





Su popularidad disminuyó drásticamente con la llegada de Red Hat Linux y otros, pero Slackware Linux todavía sigue siendo un sistema operativo muy apreciado entre los administradores de sistemas.

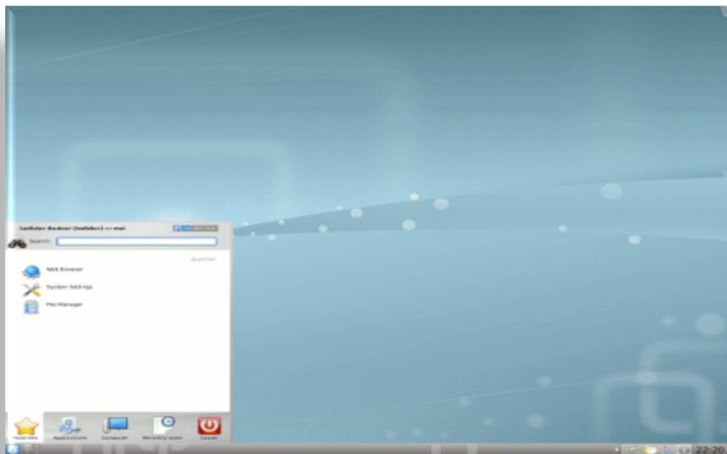


Fig. 1.7.12. Slackware Linux 13.1

Slackware es una distribución '*limpia*' y muy técnica, con un número limitado de utilidades personalizadas. Usa un instalador simple, basado en texto, y un, relativamente primitivo, sistema de gestión de paquetes, que no

resuelve dependencias de software. Como resultado, Slackware es considerado una de las distribuciones más limpias y con menos '*bugs*' (*fallos en el programa*) hoy en día. Toda la configuración se realiza mediante archivos de texto editables. Está orientada a usuarios con conocimientos avanzados en Linux (Fig. 1.7.12. Slackware Linux 13.1, con escritorio KDE.).

Si bien esta filosofía de la simplicidad tiene sus defensores, el hecho es que en el mundo actual, Slackware Linux se está convirtiendo en un sistema '*básico*', en lugar de una distribución completa con una amplia variedad de software compatible. La única excepción es el mercado de servidores, donde Slackware se mantiene popular, aunque incluso en este caso, el procedimiento de actualización de la distribución es compleja y la falta de apoyo oficial, con herramientas automatizadas para actualizaciones de seguridad, hace que sea cada vez menos competitiva. La actitud conservadora hace que se requiera mucho trabajo manual después de la instalación.





Pros. Muy estable, limpia y libre de ‘bugs’, fuerte adhesión a los principios de UNIX.

Contras. Número limitado de aplicaciones con apoyo oficial, conservadora en términos de selección de paquetes de base; procedimiento muy complejo de actualización del sistema.

Manejo de paquetes de software. ‘pkgtool’ por medio de paquetes TXZ.

Ediciones disponibles. CD’s y DVD de instalación de 32 bits (*i486*) y 64 bits (*x86_64*).

Alternativas sugeridas, basadas en Slackware. Zenwalk Linux (*escritorio*), VectorLinux (*escritorio*), SLAX (*live CD*), Slamd64 Linux (*64 bit*), Bluewhite64 Linux (*64 bit*), Wolvix (*escritorio, live CD*), GoblinX (*escritorio, live CD*).



Fig. 1.7.13. Logo

Gentoo (Fig. 1.7.13. Logo). Gentoo Linux es una versátil y totalmente libre distribución de Linux orientada a desarrolladores y profesionales de la red. A diferencia de otras distribuciones, Gentoo Linux tiene un avanzado sistema de gestión de paquetes llamado ‘Portage’. Portage tiene un número de características avanzadas incluyendo dependencias. El concepto de Gentoo Linux fue ideado en torno al año 2000 por Daniel Robbins, un ex desarrollador de Linux y FreeBSD. Fue la inmersión de su creador en FreeBSD y su función ‘autobuild’ llamados ‘ports’, que lo inspiró a incorporar algunos de los principios de gestión de software en FreeBSD dentro de Gentoo bajo el nombre de ‘portage’.





La idea era desarrollar una distribución que permitirá a los usuarios compilar el kernel y las aplicaciones directamente desde el código fuente en sus propios equipos, manteniendo así un sistema altamente optimizado y siempre al día.

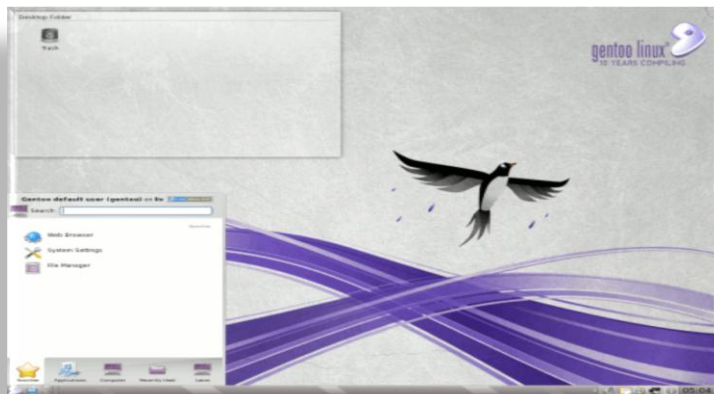


Fig. 1.7.14. Gentoo Linux 10.1

El proyecto lanzó su versión 1.0 en marzo de 2002. La gestión de paquetes de Gentoo se consideró una alternativa superior a algunos sistemas de gestión de paquetes binarios, especialmente las RPM.

Gentoo fue diseñado para usuarios avanzados (Fig. 1.7.14. *Gentoo Linux 10.1, con escritorio KDE.*). Por ejemplo, en principio, el proceso de instalación es complicado y tedioso, que requiere horas o incluso días para la compilación en línea de comandos, para construir una distribución completa de Linux; sin embargo, en 2006 el proyecto simplificó el procedimiento de instalación mediante el desarrollo de un CD instalable de Gentoo en vivo. Además de proporcionar amplias opciones de configuración, soporte para muchas arquitecturas y la capacidad de mantener el sistema actualizado sin volver a instalar. La documentación de Gentoo ha sido etiquetada como la mejor documentación en línea de cualquier distribución.

Gentoo Linux ha perdido mucha de su gloria original en los últimos años. Algunos usuarios de Gentoo han llegado a la conclusión de que el tiempo de compilación de paquetes de software sólo ofrece beneficios marginales de optimización.





Desde la dimisión del fundador de Gentoo y benevolente dictador del proyecto en 2004, la recién creada Fundación Gentoo ha estado luchando con la falta de instrucciones claras y frecuentes conflictos de desarrollo.

Queda por ver si Gentoo puede recuperar su carácter innovador del pasado o si lentamente se desintegrará.

Pros. Excelente infraestructura de gestión de software, personalización y opciones de ajuste sin precedentes, excelente documentación en línea.

Contras. Inestabilidad ocasional y riesgo de fallas, sino se tienen conocimientos avanzados; el proyecto adolece de falta de direcciones y luchas internas frecuentes entre sus desarrolladores.

Manejo de paquetes de software. 'Portage' usando paquetes fuente (SRC).

Ediciones disponibles. CD mínimo de instalación y CD en vivo (con GNOME) para las arquitecturas de procesador Alpha, AMD64, HPPA, IA64, MIPS, PPC, SPARC y x86.

Alternativas sugeridas basada en Gentoo. Sabayon Linux (escritorio, live CD / DVD), Ututo (sólo software libre).





Capítulo II. El sistema operativo GNU/Linux.

2.1. Introducción.

GNU/Linux es un sistema operativo, basado en Unix. En general, un sistema operativo consiste en varios programas fundamentales que necesita la computadora para poder comunicarse y recibir instrucciones de los usuarios.

Dos características muy peculiares diferencian a GNU/Linux del resto de sistemas operativos que podemos encontrar en el mercado: la primera, es que es libre, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna empresa desarrolladora de software por el uso del mismo; la segunda, es que el sistema viene acompañado del código fuente.

GNU/Linux se distribuye bajo la Licencia Pública General de GNU, por lo tanto, el código fuente tiene que estar siempre accesible y cualquier modificación o trabajo derivado tiene que tener esta licencia.

El sistema ha sido diseñado y programado por multitud de programadores alrededor del mundo. El núcleo del sistema sigue en continuo desarrollo bajo la coordinación de Linus Torvalds, la persona de la que partió la idea de este proyecto a principios de la década de los noventa; dirige el trabajo de varios cientos de desarrolladores, con la ayuda de algunas personas de confianza.

Hoy en día, grandes compañías, como IBM, SUN, HP, Novell, Canonical y RedHat; entre otras muchas, aportan a Linux grandes ayudas tanto económicas como de código.

Día a día, más y más programas y aplicaciones están disponibles para este sistema operativo, y la calidad de los mismos aumenta de versión a versión.

Cada vez son más las empresas, desarrolladoras de software comercial, que distribuyen sus productos para Linux.





Es menos probable que un sistema basado en Linux se colapse, además tiene mejor capacidad para ejecutar múltiples programas al mismo tiempo y es más seguro que muchos otros sistemas operativos.

Debido a estas ventajas, Linux es el sistema operativo que ha experimentado mayor crecimiento en el mercado de los servidores. Últimamente, Linux está empezando a ser popular entre los usuarios domésticos y en el sector empresarial.

Características generales de un sistema basado en Linux:

- ✓ **Multitarea:** La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo, 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 se puede utilizar GNU/Linux son: IBM-PC (*que incluye las arquitecturas I386 (x86), IA64 o AMD64 (x86_64)*), Macintosh (*que incluye la arquitectura Gecko y PowerPC*) o SPARC; también existen versiones para su utilización en otras plataformas, como Amiga y Atari.
- ✓ **Multiprocesador:** Soporte para sistemas con más de un procesador está disponible para Intel, AMD y SPARC.

El proyecto GNU ha desarrollado un conjunto de herramientas de software libre para ser utilizados por sistemas operativos tipo Unix, como Linux. Estas herramientas permiten a los usuarios desarrollar tareas que van desde las más básicas, como copiar o eliminar archivos del sistema; a las más avanzadas, como escribir y compilar programas o hacer edición sofisticada en una gran variedad de formatos de archivos.





Tantas opciones de distribuciones, y paquetes en general, confunden a veces a los usuarios de otros sistemas operativos que no están acostumbrados a trabajar con un sistema operativo abierto, que presenta opciones tan variadas.

En los temas posteriores se pretende despejar las dudas que puedan tener los usuarios noveles, para que sean capaces de utilizar un sistema basado en Linux sin mayores problemas.

2.2. ¿Cómo elegir una distribución?

Cómo ya se describió en el capítulo anterior, hay tantas distribuciones del sistema GNU/Linux como colores. Para decidirse entre una de las muchas distribuciones que hay actualmente en el mercado no hay ninguna fórmula exacta.

La decisión depende de varios factores, entre los cuales están el hardware del que disponemos, las necesidades concretas que tengamos para trabajar, nuestro nivel de conocimientos sobre la materia y la disposición que tengamos para *'tocar y jugar'* con el sistema, si es que este tipo de cosas nos llaman la atención.

Mucha gente prefiere que el sistema le ayude a hacer su trabajo de manera rápida y fácil, sin tener que preocuparse por configurar cosas *'extrañas'* como por ejemplo, montar sistemas de archivos; otros prefieren modificar a sus anchas la computadora y el software para aprender, sacar mayor rendimiento, etc. Es cuestión de gustos y objetivos.

Para hacer de su elección lo más acertada posible, se puede dar una serie de consejos a nivel de ayuda.

Lo primero que debemos hacer es comprobar que el hardware del que disponemos, o tenemos la intención de adquirir, esté soportado por la distribución.





Lo segundo que tendríamos que tomar en consideración, es el uso que le vamos a dar al sistema operativo, es decir, si lo queremos sólo para jugar, trabajar, hacer un servidor, etc. Y las necesidades que tendremos, ver discos duros de otros sistemas operativos, trabajar en red, etc.

Tendremos que valorar de forma objetiva el nivel de conocimientos que tenemos, y el tiempo que disponemos para aprender y '*jugar*' con el sistema. El aprendizaje de un nuevo sistema operativo como Linux, y más si provenimos de entorno Windows, requiere su tiempo.

Haciendo estas valoraciones, podemos escoger una u otra distribución. En lo particular y para el propósito de esta tesis, he escogido la última versión de la distribución Ubuntu.

2.3. Ubuntu.

Es importante puntualizar que de aquí en adelante todas las características presentadas están basadas, especificadas y orientadas a la distribución Ubuntu 10.04 LTS (*Fig. 2.3.1*).



Ubuntu ofrece un sistema predominantemente enfocado a computadoras de escritorio, aunque también proporciona soporte para servidores. Basado en Debian, Ubuntu concentra su objetivo en la facilidad y la libertad de uso, los lanzamientos regulares (*cada 6 meses*) y una fácil instalación.

Fig. 2.3.1. Ubuntu 10.04 LTS.





Ubuntu está patrocinado por Canonical Ltd., una empresa privada, fundada y financiada por el empresario sudafricano Mark Shuttleworth.

El nombre de la distribución proviene del concepto zulú y xhosa de ubuntu, que significa *'humanidad hacia otros'*. Ubuntu es un movimiento sudafricano encabezado por el obispo Desmond Tutu, quien ganó el Premio Nobel de la Paz en 1984 por sus luchas en contra del Apartheid en Sudáfrica.

Mark Shuttleworth, mecenas del proyecto, se encontraba muy familiarizado con la corriente, y tras ver similitudes entre los ideales de los proyectos GNU, Debian y en general con el movimiento del software libre; decidió aprovechar la ocasión para difundir los ideales de Ubuntu.

El eslogan de Ubuntu, *'Linux para seres humanos'* (en inglés, *'Linux for Human Beings'*), resume una de sus metas principales: hacer de Linux un sistema operativo más accesible y fácil de usar por cualquier persona, en cualquier parte del mundo.

La versión estable más reciente de Ubuntu es la 10.04, nombre clave *'Lucid Lynx'* (*Lince Lucido, en español*) liberada el 29 de abril de 2010. Es versión LTS (*Long-Term Support*) con soporte extendido de 3 años para escritorio y 5 para servidores.

2.3.1. Diferencia con otras distribuciones.

Existen muchas distribuciones diferentes de GNU/Linux: Debian, Gentoo, RedHat o Mandriva, son algunos ejemplos. Ubuntu es tan sólo otro competidor en lo que ya constituye un mundo altamente competitivo. Entonces, ¿qué hace a Ubuntu diferente?





- ✓ Basado en Debian (*una de las distribuciones más respetadas, tecnológicamente avanzadas y mejor soportadas*), Ubuntu pretende crear una distribución que proporcione un sistema GNU/Linux actualizado y coherente para la informática de escritorio y servidores. Ubuntu incluye una cuidadosa selección de los paquetes de Debian y mantiene su poderoso sistema de gestión de paquetes que nos permite instalar y desinstalar programas de una forma fácil y limpia.
- ✓ A diferencia de la mayoría de las distribuciones, que vienen con una enorme cantidad de software que pueden ser, o no, de utilidad; la lista de paquetes de Ubuntu se ha reducido para incluir sólo aplicaciones importantes y de alta calidad.
- ✓ Con la mirada puesta en la calidad, Ubuntu proporciona un entorno robusto y funcional, adecuado tanto para uso doméstico como profesional y se publica una nueva versión cada seis meses.
- ✓ Ubuntu se ha centrado en las necesidades de los usuarios, haciendo un sistema fácil de usar, pretendiendo terminar con el mito de que Linux es difícil y/u orientado a gente con avanzados conocimientos informáticos.

2.3.2. Principios de Ubuntu.

La filosofía de Ubuntu se basa en los siguientes principios:

- ✓ Ubuntu siempre será gratuito; la gente de Ubuntu quiere que lo mejor de su trabajo esté libremente disponible para todos.
 - ✓ Para hacer que Ubuntu pueda ser usado por el mayor número de personas posible, Ubuntu emplea las mejores herramientas de traducción y accesibilidad que la comunidad del Software Libre es capaz de ofrecer.
-





- ✓ Ubuntu publica de manera regular y predecible, una nueva versión cada seis meses. Puede usar la versión estable o probar y ayudar a mejorar la versión en desarrollo.
- ✓ Ubuntu está totalmente comprometido con los principios de desarrollo del software de código abierto, anima a la gente a utilizarlo, mejorarlo y compartirlo.

2.3.3. Soporte de la distribución.

Cada versión de Ubuntu está soportada por Canonical al menos durante 18 meses con actualizaciones genéricas y de seguridad. Ubuntu 10.04 LTS es una versión especial preparada para un uso empresarial y está soportada durante 3 años para el escritorio y durante 5 años para el servidor.

Como resultado, es posible recibir soporte para Ubuntu 10.04 LTS durante un periodo de tiempo mayor de lo normal. Por tanto, esta versión de Ubuntu se identifica como '*LTS*' o '*Long-Term Support*' (*Soporte a Largo Plazo*).

2.3.4. El escritorio.

El escritorio predeterminado en Ubuntu es GNOME (*ver 2.7.1*), GNOME es líder como escritorio y como plataforma de desarrollo para sistemas GNU/Linux.

El proyecto Kubuntu ofrece a los usuarios de Ubuntu una opción alternativa al entorno de escritorio GNOME. Gracias a ello, los usuarios de Ubuntu pueden instalar y usar el escritorio KDE (*ver 2.7.2*) fácilmente en su sistema. Para una completa utilización de KDE en Ubuntu, instale el paquete 'kubuntu-desktop'. Una vez instalado, podrá elegir si usa Gnome o KDE.





Otra opción es Xfce. Este escritorio ofrece a los usuarios una opción alternativa al entorno de escritorio GNOME mucho más ligera y perfecta para equipos con pocos recursos o antiguos. Se puede instalar y usar el escritorio Xfce instalando el paquete 'xubuntu-desktop'. Una vez instalado, podrás elegir el escritorio que quieras usar.

2.3.5. Numeración de versiones.

La numeración de las versiones de Ubuntu indica la fecha de lanzamiento de la distribución, más concretamente el año y el mes. El primer lanzamiento (*Warty Warthog*) fue en octubre de 2004, por lo tanto la versión fue la 4.10. La versión

actual (*Lucid Lynx*) fue lanzada en Abril de 2010, por lo que su número de versión es 10.04.

Versión	Nombre clave	Fecha de lanzamiento
4.10	Warty Warthog	20/10/04
5.04	Hoary Hedgehog	08/04/05
5.10	Breezy Badger	13/10/05
6.06 LTS	Dapper Drake	01/06/06
6.10	Edgy Eft	26/10/06
7.04	Feisty Fawn	19/04/07
7.10	Gutsy Gibbon	18/10/07
8.04 LTS	Hardy Heron	24/04/08
8.10	Intrepid Ibex	30/10/08
9.04	Jaunty Jackalope	23/04/09
9.10	Karmic Koala	29/10/09
10.04 LTS	Lucid Lynx	29/04/10

Hay dos versiones de Ubuntu al año. La primera versión de Ubuntu, por ejemplo, fue Ubuntu 4.10 y fue lanzado el 20 de octubre de 2004. Los números de versión para las versiones futuras son provisionales, y si la liberación se ha retrasado el número de versión cambia en consecuencia.

Las versiones de Ubuntu también se dan con nombres 'clave' (*Fig. 2.3.5.1*), utilizando un adjetivo y un animal (por ejemplo, '*Dapper Drake*' y el '*Intrepid Ibex*').

Fig. 2.3.5.1. Versiones publicadas.





Con excepción de los tres primeros lanzamientos, los nombres en clave han sido en orden alfabético, lo que permite identificar de qué versión es más reciente.

Las entregas están programadas para ser aproximadamente un mes después del lanzamiento de GNOME. En consecuencia, cada versión de Ubuntu viene con una versión actualizada de GNOME.

2.4. Hardware soportado.

Linux ha sido criticado con frecuencia por falta de controladores para cierto hardware de computadoras de escritorio. Sin embargo, el progresivo incremento en la adopción de Linux en el escritorio ha mejorado el soporte de hardware por parte de terceros o de los propios fabricantes, provocando que, en los últimos años, los problemas de compatibilidad se reduzcan.

Existen algunas compañías que hacen difícil el soporte al no publicar las especificaciones de su hardware. Esto quiere decir que a lo mejor no pueda usar su hardware con GNU/Linux.

Empresas como IBM, Intel Corporation, Hewlett-Packard, Dell o MIPS Technologies, tienen programadores en el equipo de desarrolladores del núcleo Linux que se encargan de mantener los controladores para el hardware que fabrican. Este grupo de programadores también se le suman los que provee grandes distribuidores de soluciones Linux como Novell o Red Hat.

Algunas empresas proporcionan controladores de hardware para Linux, pero sin liberar el código fuente. Esto implica que se deba depender de ellas para solucionar posibles errores o mejorar los controladores. Dos empresas que hacen esto son Nvidia y AMD, aunque esta última ayudó a la creación de controladores libres liberando especificaciones sobre su hardware.





Ubuntu cuenta con un potente sistema automático de detección de hardware. Si una vez terminada la instalación se detecta hardware, para el cual no se proporcionó un controlador en el disco de instalación (*por cuestiones legales*), se le dará la opción al usuario si quiere instalar controladores no libres en su sistema.

2.4.1. Arquitecturas soportadas.

Ubuntu soporta oficialmente dos arquitecturas de hardware: x86 y x86_64; sin embargo ha sido portada extraoficialmente a otras arquitecturas, por ejemplo: PowerPC, SPARC y Playstation 3. A partir de Ubuntu 9.04 (*abril de 2009*) se empezó a ofrecer soporte oficial para procesadores ARM. Prácticamente todos los procesadores basados en x86 y x86_64 utilizados en computadoras personales modernas están soportados, incluyendo todas las variedades de Intel y AMD. Esto también incluye los procesadores VIA (*anteriormente Cyrix*), así como a los procesadores con múltiples núcleos.

2.4.2. Procesadores múltiples.

Existe soporte para procesadores múltiples (*también llamado 'multi-procesamiento simétrico' o SMP*). La imagen del núcleo Linux para Ubuntu se compiló con el soporte SMP-alternatives. Esto significa que Ubuntu detectará el número de procesadores (*o núcleos de procesador*) y desactivará automáticamente SMP en los sistemas con un sólo procesador o núcleo.

2.4.3. Tarjetas gráficas.

Prácticamente cualquier tarjeta de vídeo moderna es compatible con Ubuntu.





El soporte de Ubuntu para interfaces gráficas está limitado por el soporte subyacente proporcionado por los fabricantes que proporcionan controladores no libres.

Los puertos de vídeo AGP, PCI y PCIe funcionan bajo Ubuntu. Puede encontrar más detalles sobre tarjetas, monitores, dispositivos de señalamiento y buses en '<http://xorg.freedesktop.org>'

2.4.4. Hardware de conectividad de red.

Todas las tarjetas de interfaz de red (*también llamadas 'network interface card' o NIC*) están también soportadas en Ubuntu en el sistema de instalación.

Los controladores modulares se cargarán automáticamente por regla general. Esto incluye la mayoría de las tarjetas PCI y PCMCIA.

2.4.5. Tarjetas de red inalámbrica.

En general también se soportan las tarjetas de red inalámbricas. El núcleo oficial de Linux soporta un número creciente de adaptadores inalámbricos. Aunque puede hacer funcionar en Ubuntu muchos adaptadores de red inalámbricos que no se soportan en el núcleo oficial de Linux.

Aún se está desarrollando en el instalador la posibilidad de utilizar una red inalámbrica durante el proceso. El hecho de que funcione (*o no*) dependerá de su tipo de adaptador y de la configuración de su punto de acceso.

Al término de la instalación del sistema operativo podrá instalar el controlador que necesite (*después del reinicio del sistema*) y hacer la configuración de su red inalámbrica manualmente.





En algunos casos el controlador que necesite especialmente para el tipo de tarjeta puede no estar disponible como paquete Debian (*.deb*). En tal caso, deberá comprobar si existe código fuente disponible en Internet y compilar el controlador usted mismo.

Existe otra alternativa que es hacer uso de una herramienta llamada *ndiswrapper*, la cual nos permite instalar el controlador hecho para Windows en nuestro sistema. Para más información sobre esta herramienta (*instalación, configuración*) dirigirse a '<http://sourceforge.net/apps/mediawiki/ndiswrapper>'

2.4.6. Periféricos y otro hardware.

Linux soporta una amplia variedad de dispositivos de hardware como pueden ser: impresoras, escáneres, cámaras, teclados, reproductores portátiles; etc. Sin embargo, muchos de estos dispositivos no son necesarios durante la instalación del sistema. Ubuntu trata que todo esté listo para que todo funcione al conectarlo.

2.5. Requisitos de memoria y espacio en disco.

Debe tener al menos 44MB de memoria y 500MB espacio en disco duro, para realizar una instalación básica, es decir, en modo consola (*sin entorno de escritorio*). Se debe tener en cuenta que éstos son los mínimos absolutos. Si quiere instalar una cantidad razonable de software, necesitará al menos 512 MB en memoria RAM; para una instalación más o menos completa de un sistema de escritorio, necesita unos 4 GB en disco duro; ya sea todo un disco duro o una partición del mismo. Muchos paquetes Debian, de hecho, se ejecutarán más rápido en sistemas más modernos como resultado de haber descartado el soporte de estos paquetes en procesadores antiguos.





Si requiere un escritorio con aceleración gráfica y efectos gráficos 3D (*para juegos o para efectos de escritorio*), por experiencia, se recomienda un mínimo de 1 GB de memoria RAM y una tarjeta gráfica compatible con al menos 128 MB de memoria (*independiente o compartida*). Esto garantizará que no tenga mayores problemas de rendimiento.

2.6. Iniciando el sistema.

Cuando encendemos una computadora lo primero que vemos aparecer son las opciones que ofrece el *BIOS (Basic Input-Output System -Sistema Básico de Entrada/Salida-)*, que es un software muy básico instalado en la placa base, que, entre otras cosas, localiza y reconoce todos los dispositivos necesarios para cargar el sistema operativo.

El BIOS busca un dispositivo de inicio (*como el disco duro, un CD, DVD o una memoria USB*) y de ahí pasa el control al Master Boot Record o MBR, que consiste en los primeros 512 bytes del disco duro, en donde se busca un gestor de arranque.

La etapa del gestor de arranque no es totalmente necesaria, determinadas BIOS pueden cargar y pasar el control a Linux sin hacer uso del cargador de arranque. Usar un cargador de arranque facilita al usuario la forma en que el kernel será cargado. Es un programa diseñado exclusivamente para cargar un sistema operativo en memoria.

El gestor de arranque muestra un menú con los sistemas operativos que están instalados en el disco duro y que fueron detectados, el que seleccionemos será el que se cargue. Ejemplos de gestores de arranque son GRUB, LILO (*Linux LOader*) y SYSLINUX. Cada distribución trae uno u otro para su instalación. Ubuntu por defecto trae GRUB.





2.6.1. GRUB.

GRUB (**GR**and **U**nified **B**ootloader -Gran Gestor de Arranque Unificado-), es un administrador o gestor de arranque múltiple, desarrollado por el proyecto GNU, que se usa comúnmente para iniciar uno de dos o más sistemas operativos instalados en una misma computadora. Se usa principalmente en sistemas operativos derivados de UNIX.

GRUB fue inicialmente diseñado e implementado por el programador Erich Stefan Boleyn, como parte del trabajo en el arranque del sistema operativo GNU/Hurd, desarrollado por la Free Software Foundation.

Una de las características más interesantes de GRUB es su flexibilidad, ya que se puede cambiar el tiempo de espera, sistema operativo por defecto, el nombre de los sistemas operativos y toda la información del arranque de cada uno de ellos; esto se puede hacer editando el archivo `'/boot/grub/grub.cfg'`.

2.6.2. El arranque.

Cuando un sistema GNU/Linux arranca, primero se carga el kernel del sistema, después se inicia el primer proceso, denominado *'init'*, que es el responsable de ejecutar y activar el resto del sistema, mediante la gestión de los niveles de ejecución (o *runlevels*).

Un nivel de ejecución es básicamente una configuración de programas y servicios que se ejecutarán orientados a un determinado funcionamiento.

Un servicio es una funcionalidad proporcionada por la máquina, normalmente basada en *daemons* (*demonios*) o procesos en segundo plano de ejecución, que controlan peticiones de red, actividad del hardware, u otros programas que provean alguna tarea.





En el arranque de un sistema GNU/Linux, se produce todo un flujo de información muy amplia; cuando el sistema arranca, suelen aparecer los datos de detección de las características de la máquina, detección de dispositivos, arranque de servicios de sistema, entre otras cosas; y se mencionan los problemas aparecidos.

En la mayoría de las distribuciones esto puede verse en la consola del sistema directamente durante el proceso de arranque. Sin embargo, o la velocidad de los mensajes o algunas modernas distribuciones que los ocultan tras carátulas gráficas pueden impedir seguir los mensajes correctamente, por lo que si queremos ver esta información podemos utilizar el comando `'dmesg'` que nos muestra los mensajes del último arranque del kernel.

2.7. Entorno de escritorio.

En GNU/Linux el entorno gráfico no forma parte del sistema operativo, se le considera como un programa más o un conjunto de programas, a diferencia de sistemas Windows en que el entorno gráfico es una capa más del sistema operativo.

Un entorno de escritorio es un conjunto de software para ofrecer al usuario una interacción amigable y cómoda.

Es una solución completa de interfaz gráfica de usuario; ofrece iconos, barras de herramientas e integración entre aplicaciones con habilidades, como por ejemplo, arrastrar y soltar con el ratón.

En general los entornos de escritorio se distinguen por su aspecto y comportamiento particulares, aunque algunos tienden a imitar características de escritorios ya existentes.





Cada entorno de escritorio trae su propio conjunto de aplicaciones. Actualmente el entorno más conocido es el ofrecido por la familia Windows, aunque existen otros como los de Macintosh (*Classic* y *Cocoa*) y otros de código abierto como GNOME, KDE, CDE, Xfce Y LXDE.

En GNU/Linux el entorno de escritorio más utilizado por las grandes distribuciones es GNOME.

2.7.1. GNOME.



Ubuntu 10.04 LTS por defecto trae como entorno de escritorio a GNOME (**GNU Network Object Model Environment**). GNOME (*logo Fig. 2.7.1.1*) es un entorno de escritorio para sistemas operativos Unix y derivados Unix como GNU/Linux, BSD o Solaris; compuesto enteramente de software libre.

Fig. 2.7.1.1. Logo GNOME

El proyecto fue iniciado por los programadores mexicanos Miguel de Icaza y Federico Mena, a finales de la década de los noventa.

El Escritorio GNOME se lanza cada seis meses con muchas características nuevas, mejoras, solución de errores y traducciones (*Fig. 2.7.1.2*), y forma parte oficial del proyecto GNU. Nació como una alternativa a KDE. Actualmente se encuentra disponible en 48 idiomas, entre ellos el español.

Versión	Fecha	Observaciones
	Agosto de 1997	Anuncio de desarrollo de GNOME.
1.0	Marzo de 1999	Primera versión importante.
1.0.55	Octubre de 1999	





Versión	Fecha	Observaciones
1.2	Mayo de 2000	
1.4	Abril de 2001	GNOME 'Tranquility'. Última versión de las series 1.x.
2.0	Junio de 2002	Importante actualización a GTK+ 2.
2.2	Febrero de 2003	Mejoras multimedia y del administrador de archivos.
2.4	Septiembre de 2003	Epiphany, mejora de accesibilidad.
2.6	Marzo de 2004	Cambio a un administrador espacial, nueva ventana de diálogo.
2.8	Septiembre de 2004	Nuevas aplicaciones, cambios en GNOME VFS y en Nautilus.
2.10	Marzo de 2005	Nuevas aplicaciones, cambios en Nautilus, GTK+.
2.12	Septiembre de 2005	Mejoras en Nautilus; mejoras en el corta/pega entre aplicaciones e integración de freedesktop.org. Nuevas aplicaciones: Evince, visor de PDF; editor de menú; nuevo sistema de autenticación y herramientas administrativas.
2.14	Marzo de 2006	Mejoras en el rendimiento, búsqueda más completa, mejoras en la gestión de ventanas, deskbar, mejoras en el inicio de sesión, cambio rápido de usuario, gedit mejorado, salvapantallas integrado, fácil configuración y más.
2.16	Septiembre de 2006	Mejoras en los efectos 3D; nuevo gestor integrado de energía; inclusión oficial de Tomboy; inclusión oficial de Alacarte como editor de menús; mejor control del uso del disco duro gracias a Baobab; acercamiento a personas con minusvalía gracias al lector Orca; mejoras en Epiphany y aplicaciones GTK+; varios cambios en Evolution; y mejoras en la usabilidad.
2.18	Marzo de 2007	Integra una inmensidad de mejoras al diseño visual, rendimiento de los componentes, y la creciente colección de aplicaciones integradas. Se han agregado un conjunto de herramientas para el desarrollo, incluyendo Devhelp y Glade.
2.22	Marzo de 2008	Mejoras en el reproductor multimedia, en Evolution, Evince y Tomboy. Nuevo programa para tomar imágenes y grabar vídeos, Cheese. Inclusión de Anjuta en la plataforma de desarrollo. Simplificación de algunas preferencias de teclado y ratón. Inicio de la migración de GnomeVFS a GIO y GVFS.





Versión	Fecha	Observaciones
2.20	Septiembre de 2007	Mejoras en la gestión de energía. Mayor velocidad y estabilidad. Reorganización de ciertas componentes en las preferencias.
2.24	Septiembre de 2008	Introduce el uso de solapas en el gestor de archivos Nautilus, nueva versión de Ekiga, se añade un organizador de tareas, se añaden infinidad de mejoras a Deskbar, mejoras en el aspecto y en los controles de gestión de configuración de pantalla.
2.26	Marzo de 2009	Incluye Brasero como software de grabación de CD y DVD, soporte para videoconferencia en Empathy, mejoras en el reproductor multimedia, integración con lectura de huellas dactilares, integración del lector de documentos empotrado en aplicaciones.
2.28	Septiembre de 2009	Incluye Gnome Bluetooth para ayudar a los usuarios a manejar sus dispositivos Bluetooth, mejoras en mensajería instantánea gracias a Empathy, mejoras en la reproducción de DVD en Totem, ahora Evince (lector de PDF) tiene la capacidad de modificar anotaciones en los documentos PDF. El Control de volumen de GNOME ha añadido la capacidad para que controle el desvanecimiento de canales y el 'subwoofer'.
2.30	Marzo de 2010	Se agregaron nuevas y avanzadas características al gestor de archivos (Nautilus); incluye mejoras al escritorio remoto, sincronización de notas de manera más fácil (Tomboy) y un uso general más suave.

Fig. 2.7.1.2. Versiones publicadas de GNOME.

Al iniciar una sesión gráfica verá el escritorio GNOME (*Fig. 2.7.1.3*). Este cuenta con iconos para acceder rápidamente a aplicaciones y con menús que le permiten iniciar programas. También le permite usar varios espacios de trabajo, cada uno como un escritorio independiente de los demás.





El botón izquierdo del ratón normalmente permite elegir una opción de un menú o activa un icono. El botón derecho tiene diversas aplicaciones de acuerdo al contexto, por ejemplo, sobre los iconos permite configurarlos; el botón del centro permite pegar el texto que se hubiera seleccionado con el ratón, si su ratón sólo tiene 2 botones, puede 'emular' el botón del centro oprimiendo simultáneamente el izquierdo y el derecho. Para seleccionar un texto se pasa por encima del mismo con el puntero del ratón mientras se mantiene presionado el botón izquierdo.



Fig. 2.7.1.3. Escritorio GNOME.

El escritorio GNOME es bastante configurable: puede configurar los menús, los iconos, las tipografías, el fondo, el protector de pantalla, el tema, el administrador de ventanas, sonido, la interacción con las ventanas y muchos otros detalles de acuerdo a su gusto.

Para hacer algunas de las configuraciones puede emplear opciones de los menús GNOME.

El gestor de ventanas empleado decorará cada ventana con botones que le permitirán cerrar, maximizar o minimizar. Las ventanas minimizadas se verán en la parte inferior del escritorio GNOME, aunque no necesariamente está en la parte inferior, porque puede reubicarse de acuerdo a su gusto.

Entre los menús de GNOME, existen múltiples herramientas: calculadora, editor de texto sencillo, calendario, procesador de textos, hoja de cálculo, reproductor multimedia y muchas otras.





2.7.2. KDE.



Fig. 2.7.2.1. Logo KDE.

El proyecto fue iniciado en 1996 por el programador alemán Matthias Ettrich, quien buscaba crear una interfaz gráfica unificada para sistemas Unix. En sus inicios imitó a CDE (*Common Desktop Environment*), un entorno de escritorio utilizado por varios Unix. KDE es un proyecto de software libre (Fig. 2.7.2.1. Logo KDE.) para la creación de un entorno de escritorio e infraestructura de desarrollo para diversos el sistemas operativo GNU/Linux, Mac.

Las aplicaciones KDE están traducidas a más de 88 idiomas y están construidas con los principios de facilidad de uso y de accesibilidad moderna en mente. Las aplicaciones de KDE 4 funcionan de forma completamente nativa en GNU/Linux.

KDE se basa en el principio de la personalización; todos los componentes de KDE pueden ser configurados en mayor o menor medida por el usuario. Las opciones más comunes son accesibles en su mayoría desde menús y diálogos de configuración. Los usuarios avanzados pueden optar por editar los archivos de configuración manualmente, obteniendo en algunos casos un mayor control sobre el comportamiento del sistema. La apariencia de KDE (Fig. 2.7.2.2) es configurable en varios niveles. Tanto el gestor de ventanas (*llamado KWin*) como los controles (*botones, menús, etc.*) utilizan 'estilos' intercambiables, que definen

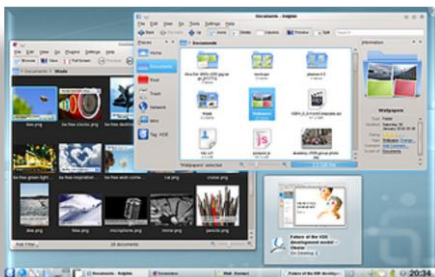


Fig. 2.7.2.2. Escritorio KDE

cada aspecto de su apariencia. Es por este motivo que KDE no mantiene una única apariencia entre versiones, sino que se opta por aquella más ampliamente aceptada en el momento de cada nuevo lanzamiento.





La intención del proyecto KDE es la de crear un entorno de escritorio que no se comporte de un modo predefinido, sino que permita al usuario adecuar el sistema a su gusto y comodidad. Esto no impide que KDE resulte fácil de usar para nuevos usuarios, detalle al que no se resta importancia.

La 'K', originariamente, representaba la palabra 'Kool', pero su significado fue abandonado más tarde. Muchas aplicaciones de KDE tienen una 'K' en el nombre, la mayoría como una letra inicial mayúscula, sin embargo, hay excepciones notables como 'amarok'. La 'K' en muchas aplicaciones de KDE se obtiene cambiando la letra, que en un principio empieza con 'C' o 'Q', por ejemplo 'Konsole' y 'Kommander'. Además, algunos tienen la 'K' como un prefijo a la palabra con que inicia el nombre, por ejemplo 'KOffice'; sin embargo, la tendencia es no tener una 'K' en el nombre del todo, como 'plasma' y 'dolphin'.

En la tabla, de la Figura 2.7.2.3 se muestra los lanzamientos más importantes que el proyecto KDE ha liberado:

Fecha de publicación	Versión liberada
12 de julio de 1998	KDE 1.010
23 de octubre de 2000	KDE 2.012
15 de agosto de 2001	KDE 2.214
3 de abril de 2002	KDE 3.015
28 de enero de 2003	KDE 3.116
3 de febrero de 2004	KDE 3.217
29 de noviembre de 2005	KDE 3.520
11 de enero de 2008	KDE 4.021
27 de enero de 2009	KDE 4.223
4 de agosto de 2009	KDE 4.324
31 de agosto de 2010	KDE 4.5.1

Fig. 2.7.2.3. Versiones de KDE.





2.8. La terminal.

Una terminal es una forma de acceder al sistema sin utilizar la interfaz gráfica, es decir, realizar todo tipo de tareas en formato texto. La forma de utilizar el sistema de este modo es mediante órdenes o comandos.

La terminal es un programa que interpreta órdenes, introducidas por medio del teclado, y los pasa al sistema operativo para su ejecución. Para algunas tareas, especialmente cosas como la configuración del sistema es muy importante utilizar la terminal.

Un comando es una instrucción o mandato que el usuario proporciona al sistema, desde la línea de órdenes o una llamada a programa, el cual generalmente está contenido en un archivo ejecutable.

La línea de comandos, consola o terminal, permite al usuario escribir instrucciones, de tal modo que, después de pulsar la tecla *'enter'* en el teclado, se ejecuten procesos internos del sistema operativo o se lancen programas externos al mismo.

Los comandos del sistema se dividen en internos y externos.

- ✓ Comandos internos: Son instrucciones que puede ejecutar el propio intérprete, es decir, son instrucciones que el propio intérprete ejecuta, bien directamente o bien haciendo una llamada al sistema operativo. Ejemplo: *'cd'* (*cambio de directorio*).
- ✓ Comandos externos: Son comandos que no sabe ejecutar el intérprete y que para su ejecución hay que localizar el ejecutable correspondiente en el disco, o los discos, del sistema; para ello, hay que introducir la ruta completa. Es decir, son ejecutables que se encuentran en alguna parte del disco de la computadora. Ejemplo: *../ejecutame.sh*





El intérprete de comandos es un programa de estructura repetitiva, cuyas funciones principales son las siguientes:

- ✓ Muestra el prompt del sistema.
- ✓ Expande los caracteres genéricos tales como el asterisco o la interrogación.
- ✓ Realiza los redireccionamientos.
- ✓ Si es un comando interno, lo ejecuta.
- ✓ Si es un comando externo, lo busca en un conjunto de directorios y si lo encuentra solicita su ejecución al sistema operativo.
- ✓ En caso contrario indica que no puede ejecutar dicho comando.

Comandos de uso frecuente. Ver Apéndice B.

2.9. Usuarios y grupos.

Cuando se instala Ubuntu inicialmente, se crea una única cuenta de usuario. Es posible crear posteriormente usuarios adicionales en el sistema.

Por razones de seguridad, las tareas administrativas en Ubuntu sólo pueden ser realizadas por usuarios con '*privilegios*' especiales. Un usuario normal puede realizar todas las tareas habituales que se necesitan a diario, pero las tareas de administración del sistema sólo podrán realizarlas usuarios autorizados.

En GNU/Linux (y *Unix en general*) existe un '*superusuario*' llamado '*root*'. Este usuario tiene acceso total al sistema, pudiendo instalar, modificar y eliminar cualquier archivo o programa, trabajar con este usuario es potencialmente peligroso, por lo que sólo es recomendable para gente experimentada, que sepa lo que hace. Además, sólo debería usarse para tareas de administración del sistema, nunca para realizar las actividades habituales de un usuario normal.





Independientemente del debate sobre las ventajas e inconvenientes de habilitar la cuenta root, lo cierto es que en Ubuntu, un usuario normal no necesita hacerlo. Se puede autorizar a los usuarios normales para que realicen tareas administrativas mediante el comando *'sudo'*.

La primera cuenta de usuario que se crea en el sistema durante la instalación, por defecto, está capacitada para usar sudo y realizar las tareas administrativas necesarias. Además, podemos restringir o permitir el acceso a sudo al resto de usuarios del sistema, con la utilidad de *'Usuarios y Grupos'* que encontramos en el panel de administración.

Cuando ejecute una aplicación que requiera privilegios administrativos, se le pedirá que introduzca su contraseña de usuario. De esta forma se dificulta que aplicaciones maliciosas puedan dañar su sistema, y sirve como un recordatorio de que está a punto de realizar acciones administrativas que requieren que usted tenga cuidado.

Hay que notar que, siempre que usemos sudo el sistema nos pedirá nuestra clave. La contraseña se guarda por defecto durante 15 minutos, después de ese tiempo, se tendrá que volver a introducir, la contraseña no se mostrará en la pantalla, ni siquiera con los clásicos asteriscos, sin embargo, se está escribiendo.

Para administrar usuarios y grupos, estos son los comandos principales:

- ✓ Para añadir un usuario.

sudo adduser <nombreusuario>

- ✓ Para ver las opciones de añadir usuarios prueba el comando man.

man adduser

- ✓ Puedes también crear un nuevo grupo para tus usuarios.

sudo addgroup <nombregrupo>





✓ Para añadir un usuario existente a un grupo existente puedes usar:

```
sudo adduser <nombreusuario> <nombregrupo>
```

Para eliminar usuarios y grupos se emplean los comandos *'userdel'* y *'groupdel'* respectivamente.

Por ejemplo: El comando *'sudo userdel alumno'* elimina el usuario alumno. Si además se indica la opción *-r*, también se borrará el directorio personal del usuario con todo su contenido.

El comando *'sudo groupdel alumnos'* elimina el grupo alumnos.

2.10. Paquetes y administración de paquetes.

2.10.1. ¿Qué es un paquete?

Es un hecho que cada programa necesita otros recursos para trabajar. Cuando se instala software, se requieren cientos de archivos, sólo para que el programa se inicie.

Ubuntu utiliza *'paquetes'* para almacenar todo lo que un programa en particular necesita para ejecutarse. Un paquete, entonces, es esencialmente una colección de archivos construidos en un único archivo, el cual puede ser manejado mucho más fácilmente.

2.10.2. ¿Fuente o binario?

Normalmente, cuando alguien empaqueta un programa, incluye el *'código fuente'* del programa en el paquete. El código fuente está escrito por programadores y es esencialmente una lista de instrucciones para la computadora, que puede ser leída y escrita por las personas. Las computadoras pueden entender este código si está *'interpretado'* de forma que puedan utilizarlo directamente.





Una de las maneras de interpretar código para una computadora es *'traducir'* o *'compilar'* en un *'binario'*, el cual la computadora puede entender.

La gente que hace los paquetes (*llamados 'empaquetadores' (packagers)*) no los convierten desde el inicio en paquetes binarios simplemente porque, los diferentes tipos de computadoras necesitan tipos de binarios diferentes. Así que si se hace un paquete para un tipo de computadora (*como por ejemplo un PC Intel*), este no funcionará en otros (*como por ejemplo Apple Macintosh*).

Los *'paquetes fuente'* son sencillamente paquetes que incluyen código fuente, y generalmente pueden ser utilizados por cualquier tipo de máquina, si el código se compila de manera correcta.

Los *'paquetes binarios'* son los que están contruidos específicamente para algún tipo de computadora o *'arquitectura'*. Ubuntu soporta las arquitecturas x86, AMD64 y PPC. Para saber qué arquitectura estás usando, abre una *'terminal'* y escribe *'arch'* y presiona la tecla *'enter'*.

2.10.3. Dependencias de Paquetes.

Algunas aplicaciones frecuentemente utilizan los mismos archivos que otras aplicaciones. En vez de poner esos archivos en cada paquete, se puede instalar un paquete separado para proporcionar esos archivos a todos los programas que los necesiten. Cuando un paquete necesita de otro, para funcionar correctamente, se conoce como *'dependencia de paquete'*. Cuando se instala un programa, las dependencias deben ser instaladas al mismo tiempo. Normalmente, la mayoría de las dependencias requeridas estarán instaladas, pero se pueden necesitar también unos cuantos extras. Por eso, cuando se instala un paquete, no debe sorprender que otros paquetes se instalen también, son justamente dependencias para que el paquete escogido pueda funcionar correctamente.





2.10.4. Canales de software (repositorios).

Ubuntu almacena todos los paquetes en localizaciones llamadas '*canales de software*' o más comúnmente llamados '*repositorios*'. Un repositorio es una localización que almacena paquetes de tipos similares, que pueden ser descargados e instalados. El gestor de paquetes almacena un índice de todos los paquetes disponibles en ese repositorio. A veces se reconstruye este índice para asegurar que todos los datos están al día y qué paquetes han sido actualizados o añadidos al repositorio desde la última comprobación.

Hay cuatro canales de software o repositorios principales en Ubuntu para cada arquitectura:

- ✓ **Main.** Este componente contiene aplicaciones que son software libre, que pueden ser redistribuidas y que están soportadas completamente por el equipo de Ubuntu. Esto incluye las aplicaciones de código abierto más populares y más seguras disponibles, muchas de las cuales son instalados por defecto cuando instalamos Ubuntu.

El software en '*main*' incluye una colección de aplicaciones escogida a mano que los desarrolladores de Ubuntu, la comunidad y los usuarios creen que son importantes; y que el equipo de seguridad y de distribución de Ubuntu está dispuesto a dar soporte.

De este modo, cuando instalamos software desde el componente '*main*' estamos seguros que el software vendrá con actualizaciones de seguridad y soporte técnico.

- ✓ **Restricted.** El componente '*restricted*' está reservado para software que es utilizado muy comúnmente y que es soportado por el equipo de Ubuntu a pesar de que no esté disponible bajo una licencia completamente libre.





Debido a ello, es posible que no se pueda proporcionar soporte completo para este software por que el equipo de Ubuntu no está autorizado a modificar dicho software, sino simplemente a enviar informes de problemas a los autores verdaderos.

Este software está incluido por que resulta esencial para que Ubuntu funcione correctamente en ciertas máquinas (*un ejemplo de ello son los drivers binarios publicados por algunos fabricantes de tarjetas gráficas*), ya que es la única manera de que Ubuntu corra en esas máquinas. Por defecto, en Ubuntu sólo se usa software de código abierto a menos que no haya ninguna otra alternativa. El equipo de Ubuntu trabaja con esos fabricantes para acelerar la transición a código abierto de su software para asegurar que haya tanto software bajo licencia libre como sea posible.

- ✓ **Universe.** En *'universe'* podemos encontrar casi cualquier paquete de software de código abierto y de software disponible bajo una variedad de licencias, todo construido automáticamente a partir de una variedad de fuentes públicas. Todo este software es compilado, sometido a las bibliotecas y utilizando las herramientas que forman parte de *'main'*, pero viene sin ninguna garantía de obtener actualizaciones de seguridad y soporte.

El componente *'universe'* incluye miles de paquetes de software. A través de *'universe'* los usuarios pueden tener la diversidad y la flexibilidad que ofrece el vasto mundo del código abierto con el añadido de la estabilidad del núcleo Ubuntu.

Los paquetes de software que son muy populares o que tengan un buen mantenimiento son movidos de *'universe'* a *'main'*.





- ✓ **Multiverse.** El componente '*multiverse*' contiene software que es '*no libre*', lo que significa que los requisitos de licencia de este software no coinciden con la política de licencias del componente '*main*'. Es responsabilidad del usuario que instala software de este componente el verificar los derechos y responsabilidades que implica emplear este software. Este software no está soportado, y generalmente no puede ser corregido, por lo que debe ser empleado a responsabilidad del usuario.

Además de los repositorios oficiales de Ubuntu, es posible utilizar repositorios de terceros. Se tiene que ser cuidadoso al usarlos, ya que algunos no son compatibles con Ubuntu y usarlos puede causar que otros programas dejen de trabajar o pueden causar graves daños en la instalación.

Para editar los canales de software, ya sea agregar o quitar repositorios, sólo se debe editar el archivo '*/etc/apt/sources.list*'; se debe de tener privilegios administrativos para esta tarea. Un repositorio tiene la siguiente forma:

```
# deb http://us.archive.ubuntu.com/ubuntu lucide-updates main restricted
```

Un símbolo de gato (#) al inicio de la línea indica que está deshabilitado, para habilitarlo, se borra el símbolo y se guardan los cambios; esto lo podemos hacer con cualquier editor de textos.

2.11. Instalar aplicaciones.

Instalar software en Ubuntu es fácil, aquí se mostrará cómo hacerlo. Cuando se instala Ubuntu, por defecto, se instalan muchos programas útiles. Aún así, es posible que se necesite algún tipo de aplicación en particular para alguna tarea que no se pueda realizar con las aplicaciones que vienen por defecto. También es posible que se quiera probar un programa alternativo al que está instalado.





Existen diversas formas de instalar aplicaciones en Ubuntu en función de la situación en que se encuentren. Es importante enfatizar que sea cual sea el método a utilizar para instalar aplicaciones, debemos tener privilegios administrativos.

2.11.1. Instalar paquetes de los repositorios.

Esta sección explica cómo instalar paquetes desde repositorios, utilizando varias herramientas disponibles en Ubuntu. Esta es la forma más rápida, fácil y segura de instalar un programa en Ubuntu. De este modo instalamos aplicaciones comprobadas, estables y sin problemas de dependencias.

a) Métodos gráficos.

- ✓ **Centro de software de Ubuntu** (Fig. 2.11.1.1). Está en el menú Aplicaciones, permite ver qué programas se tiene en Ubuntu y qué aplicaciones se pueden añadir.



Fig. 2.11.1.1. Centro de software de Ubuntu

Este es el método más fácil de instalar programas. Simplemente ha de buscarse el programa que se desea instalar de entre las secciones (*departamentos*) que nos ofrece o bien usar el buscador. El buscador se puede usar para buscar bien el nombre del programa, o bien alguna palabra que aparezca en su descripción. Si queremos buscar un procesador de textos, por ejemplo, podemos escribir en el buscador la palabra texto y veremos cómo aparecen los programas relacionados.





Haciendo clic en la derecha, en Software instalado, podemos ver qué tenemos instalado y podemos desinstalarlo si lo deseamos.

- ✓ **Synaptic.** Instala por paquetes, te soluciona sus dependencias automáticamente, por lo que es uno de los sistemas de instalación más potentes. El gestor de paquetes Synaptic proporciona un modo más avanzado de instalar paquetes.

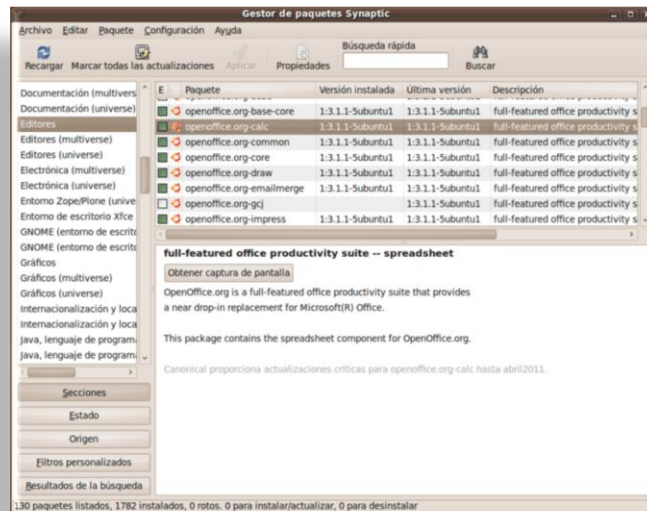


Fig. 2.11.1.2. Gestor de paquetes Synaptic.

b) Métodos basados en Texto.

Los métodos basados en texto requieren estar familiarizado con la terminal. Mediante el comando en consola `'sudo aptitude install paquete'` es posible instalar el paquete solicitado y sus dependencias. El programa `'aptitude'` es un gestor de paquetes de línea de comandos, que debe ser utilizado si las herramientas `'Centro de software de Ubuntu'` y `'Synaptic'` nos dan problemas, o si preferimos hacer uso de la terminal como herramienta de mantenimiento. Este programa nos ofrece un sistema gestor de paquetes subyacente que utiliza Ubuntu, pero es razonablemente fácil de utilizar. Los usuarios con el tiempo encontrarán que los comandos de `'aptitude'` son más rápidos de utilizar y más poderosos que la opción gráfica.





2.11.2. Instalar paquetes de otras fuentes.

Este sistema es útil si no encontramos lo que necesitamos en los repositorios, si queremos probar aplicaciones en estado de desarrollo o si queremos utilizar versiones muy nuevas que aún no existen en los repositorios.

- ✓ **Paquetes ‘*.deb’**. Cuando hacemos doble clic sobre el paquete ‘*.deb’, mediante el gestor de paquetes ‘*gDebi*’, se instala automáticamente, no soluciona dependencias automáticamente, pero avisa de las dependencias que necesita y no se instala el paquete hasta que no se hayan instalado dichas dependencias. También pueden instalarse desde consola mediante el comando ‘*dpkg -i *.deb*’.
- ✓ **Archivos ‘*.bin’**. Son los ‘*ejecutables*’ de GNU/Linux, para instalarlos sólo hay que ejecutarlos desde la terminal, esto es especificando la ruta completa del archivo y presionar ‘*enter*’.
- ✓ **Archivos ‘*.run’**. Suelen ser asistentes de instalación para GNU/Linux. Se ejecutan igual que los archivos ‘**bin*’.

2.11.3. Compilar paquetes.

Instalar aplicaciones a partir del código fuente (‘*compilar*’) es la forma más eficiente de instalar aplicaciones, con lo que la instalación queda optimizada para nuestra máquina, pero también es la más ‘*complicada*’ y no es demasiado recomendable para usuarios noveles.

Cuando se descarga el código fuente de algún paquete que se desee instalar, generalmente, en el interior vienen instrucciones de las opciones para compilar dicho paquete.

Se recomienda utilizar este método de instalación cuando se tenga algo de experiencia en Linux y en el uso de la terminal.





2.12. Actualizaciones Automáticas.

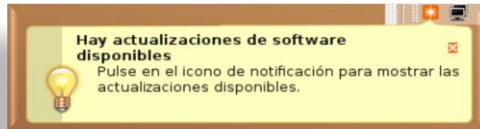


Fig. 2.12.1. Notificación de actualizaciones.

Ubuntu notificará automáticamente cuando haya actualizaciones de seguridad y de software disponibles en los repositorios. Simplemente se hace clic en el ícono que aparecerá en el área de notificación (Fig. 2.12.1.), se

escribe la contraseña de administrador y seguir las instrucciones en pantalla. No se notificará actualizaciones de paquetes instalados de otras fuentes.

Mantener al día tus datos es importante, ya que las correcciones de seguridad que protegen tu computadora se proporcionan de esta manera.

Para tener siempre las últimas actualizaciones de los paquetes instalados se debe, habitualmente, poner primero al día los repositorios.

Los comandos utilizados para esto son:

'*aptitude update*'. Actualiza tus repositorios, es decir, actualiza el índice de paquetes disponibles.

'*aptitude full-upgrade*'. Actualiza todo tu sistema.

2.13. Sistema de archivos.

El sistema de archivos de Ubuntu está basado en el File system Hierarchy Standard. Este es un estándar que con mayor o menor rigor siguen la mayoría de las distribuciones GNU/Linux y algunos otros sistemas operativos basados en UNIX.





Algunas de sus principales características son:

- ✓ Todo está organizado en una única estructura de directorios. Todo está dentro del directorio raíz, representado por el signo '/'.
- ✓ Todo es un archivo. Dentro de '/' está representado todo el sistema, los directorios y archivos presentes en los discos duros y demás dispositivos de almacenamiento montados, todo el hardware de la computadora, todos los procesos en ejecución, etc.
- ✓ Cada cosa en su sitio y un sitio para cada cosa. Todos los directorios tienen una finalidad concreta, todos los contenidos tienen una ubicación predeterminada según su naturaleza y función.

Aunque, en apariencia, todo parece formar parte de un único sistema de archivos en realidad hay varios. Además de los sistemas de archivo de disco que contienen los archivos normales, se incluye diversos sistemas de archivos virtuales que cumplen diversas funciones.

El directorio raíz es '/' de aquí 'cuelgan' todos los directorios del sistema:

/bin → Aquí tenemos los programas básicos que pueden lanzar todos los usuarios del sistema.

/boot → Aquí tenemos los archivos de configuración del arranque del sistema (como por ejemplo GRUB) y del Kernel.

/cdrom → Antiguamente aquí se montaban los CD-ROM's. Actualmente es un enlace a /media/cdrom, está ahí por compatibilidad con programas antiguos.

/dev → Aquí encontramos todos los dispositivos físicos del sistema (todo nuestro hardware). La mayoría de su contenido es generado durante el arranque.





/dev/disk → contiene los archivos que representan los discos y sus particiones.

/dev/psaux → Representa al ratón.

/dev/ram → Representa la memoria RAM.

/etc → Aquí encontramos los archivos de configuración generales del sistema y los programas.

/home → Es un directorio donde tenemos los directorios personales de los usuarios.

/home/(usuario) → Es nuestro directorio personal. Aquí es donde guardaremos nuestros documentos, música, películas, fotos y los archivos de configuración personalizados de los programas que usamos. También podemos referirnos a este directorio por su abreviatura '~'.

~/Desktop → Aquí tenemos nuestro escritorio.

/lib → Contiene las bibliotecas necesarias para que se ejecuten los programas que tenemos en /bin y /sbin.

/lost+found → Este directorio es usado por el programa fsck, cuando fsck no puede reparar completamente un sistema de archivos, guarda aquí todo lo que no sepa dónde ubicar.

/media → Aquí encontramos todas las unidades físicas que tenemos montadas: Discos duros, unidades de dvd, memorias usb, etc.





/mnt → Este ha sido el lugar tradicional para montar unidades, ha perdido gran parte de su función en favor de `/media`, pero sigue siendo útil para el montaje puntual de algunas cosas.

/opt → Es el lugar apropiado para la instalación de programas que no forman parte de la distro y que pueden no encajar bien en la estructura de `/usr`.

/proc → Su contenido representa a los procesos en ejecución. Aquí se encuentran archivos que reciben o envían información al núcleo. Es un sistema de archivos virtual, en realidad en el disco duro ese directorio está vacío.

/root → Es el directorio personal del usuario root.

/sbin → Aquí están los programas básicos que requieren privilegios de superusuario para ser usados.

/srv → Pensado para contener los archivos compartidos por los servidores. Obviamente en un sistema de escritorio no es muy usado.

/sys → Contiene información sobre el sistema y el kernel. Es un sistema de archivos virtual, en realidad en el disco duro ese directorio está vacío.

/tmp → Este directorio contiene información temporal de los programas. No se conserva su contenido, suele borrarse al reiniciar el sistema.

/usr → Este directorio contiene los archivos de los programas no esenciales. Es el directorio más grande del sistema.

/usr/bin → Aquí tenemos programas de uso general para los usuarios.





/usr/doc → Aquí tenemos documentación de los programas.

/usr/include → Aquí tenemos las cabeceras de C y C++. Son archivos con extensión *.h*. Los programadores comprenderán su significado.

/usr/sbin → Este directorio contiene los programas de administración del sistema. Para usuarios con privilegios.

/usr/lib → Aquí tenemos las bibliotecas de nuestros programas.

/usr/man → En este directorio encontramos los manuales (*man*).

/usr/src → Aquí se almacenan los códigos fuentes de nuestros programas.

/var → Este directorio contiene información variable, como registros, datos de los servidores, etc.

2.13.1 Archivos de configuración importantes.

Antes de editar cualquier archivo de configuración es importante que se realice una copia de seguridad del mismo. En ocasiones necesitamos editar un archivo y no disponemos de entorno gráfico, para ello podemos utilizar algún editor de texto para la terminal como son '*vi*' o '*nano*':

- ✓ vi <ruta archivo>
- ✓ nano <ruta archivo>

/boot/grub/menu.lst → Aquí tenemos la configuración de GRUB (gestor de arranque).





/etc/apt/sources.list → Aquí encontramos la lista de repositorios.

/etc/fstab → Este archivo contiene información sobre los dispositivos que se montaran automáticamente durante el arranque del sistema.

/etc/network/interfaces → Este archivo contiene los datos de configuración de la red.

/etc/passwd → Este archivo controla el uso de usuarios, en contraseñas, con permisos y grupos que pertenecen a cada usuario.

/etc/X11/xorg.conf → Este archivo contiene la configuración del entorno gráfico (pantalla, teclado, ratón, tarjeta gráfica; etc.).





Capítulo III. Aplicaciones para ingeniería mecánica eléctrica en GNU/Linux.

3.1. Introducción.

Hoy en día muchos usuarios, centros educativos, empresas y gobiernos están haciendo la migración a GNU/Linux y eso ha despertado muchas expectativas en todo el mundo. Mucha gente no hace la migración por miedo a no tener aplicaciones que son utilizadas a diario en el campo profesional, especialmente para los Ingenieros (*ya sea electricistas, mecánicos, civiles, químicos; etc.*), necesitan aplicaciones avanzadas para la realización de sus actividades y la eficiencia de su trabajo. En GNU/Linux se tienen muchas ventajas debido a:

- ✓ Que generalmente son mas robustas y rentables que sus alternativas.
- ✓ Proveen un flexible y conveniente entorno de desarrollo.
- ✓ Tienen un diseño libre, extendible y modular.
- ✓ Son orientadas a redes.
- ✓ Se pueden hacer máquinas potentes y seguras con muy pocos recursos.

Y, además algo para tomar en cuenta es que en GNU/Linux, la calidad de aprendizaje sería mucho mejor, porque se tendría mejor capacidad de programación y desarrollo propio de las aplicaciones.

Aunque todavía existen aplicaciones que son necesarias y no es posible instalarlas por el momento en GNU/Linux, así como va la tendencia, muy pronto cada vez más software correrá sobre GNU/Linux.

Entonces podemos ver que para las universidades que tienen Facultad de Ingeniería y para el campo profesional y la investigación, se tiene todo para desarrollar grandes proyectos, por lo que no se ve la dificultad en la migración, es solamente quitarse la costumbre de lo que conocemos actualmente.





3.2. Aplicaciones para CAD (*Computer Aided Design - Diseño Asistido por Computadora*).

3.2.1. QCAD (*CAD 2D*).

QCAD es desarrollado por la empresa RibbonSoft. Buena parte de la interfaz y de los conceptos sobre su uso son iguales que los de AutoCAD, aunque quizás más fácil de entender en QCAD.

QCAD es una aplicación para el diseño asistido por computadora en dos dimensiones (2D). Con QCAD se puede crear dibujos técnicos como planos para edificios, interiores, piezas mecánicas o esquemas y diagramas. QCAD funciona en Windows, Mac OS X, Linux y muchos sistemas Unix. El código fuente de QCAD es liberado bajo la licencia GPL (*Open Source*).

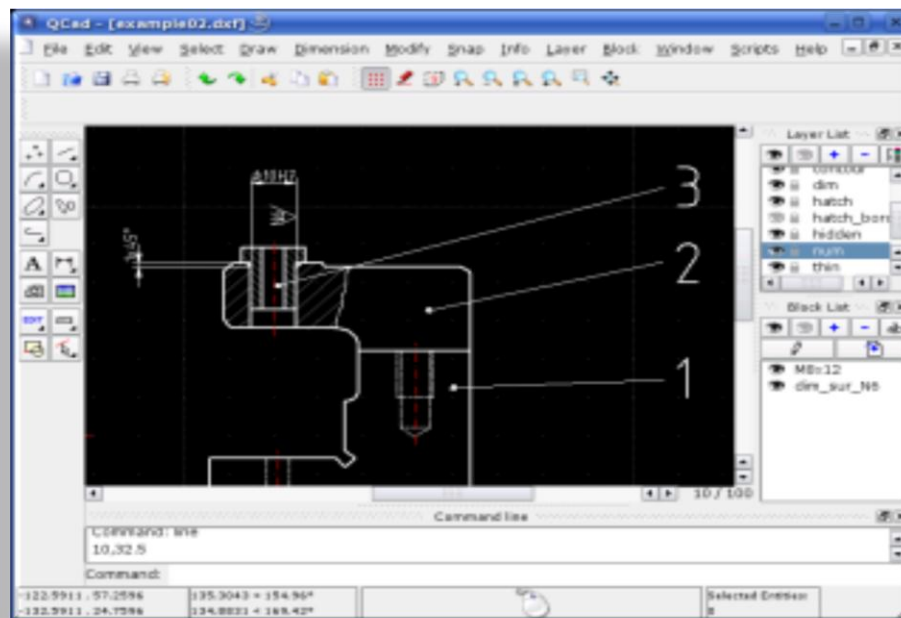


Fig. 3.2.1.1 Interfaz de usuario de QCAD.





QCAD fue diseñado con la modularidad, extensibilidad y la portabilidad en mente, pero lo que la gente nota con más frecuencia acerca de QCAD es su interfaz de usuario intuitiva. QCAD es un simple sistema de CAD 2D para todo el mundo.

No se necesita ninguna experiencia en CAD para empezar con QCAD inmediatamente (*Fig. 3.2.1.1 Interfaz de usuario de QCAD.*).

Principales características:

- ✓ Capas.
- ✓ 35 fuentes CAD incluidas.
- ✓ Sistema Métrico Internacional e Inglés.
- ✓ Formato DXF de entrada / salida.
- ✓ La impresión en escala.
- ✓ Más de 40 herramientas de construcción.
- ✓ Más de 20 herramientas de modificación.
- ✓ La construcción y modificación de puntos, líneas, arcos, círculos, elipses, polilíneas, textos, dimensiones, rellenos, imágenes de la trama.
- ✓ Referencias a objetos.
- ✓ Herramientas de medición.
- ✓ Biblioteca con más de 4800 piezas CAD.

CAM Expert es una extensión de QCAD. Cuenta con creación automática de G-Code, HP / GL y otros formatos individuales.

CAM Expert es muy flexible, puede ser configurado para crear casi cualquier texto en formato vectorial de gráficos.

El proceso de conversión desde CAD a CAM también puede incluir varias optimizaciones tales como cortar contornos del interior al exterior y la optimización de los caminos más cortos de corte (*Fig. 3.2.1.2. Proyecto en CAM Expert.*).



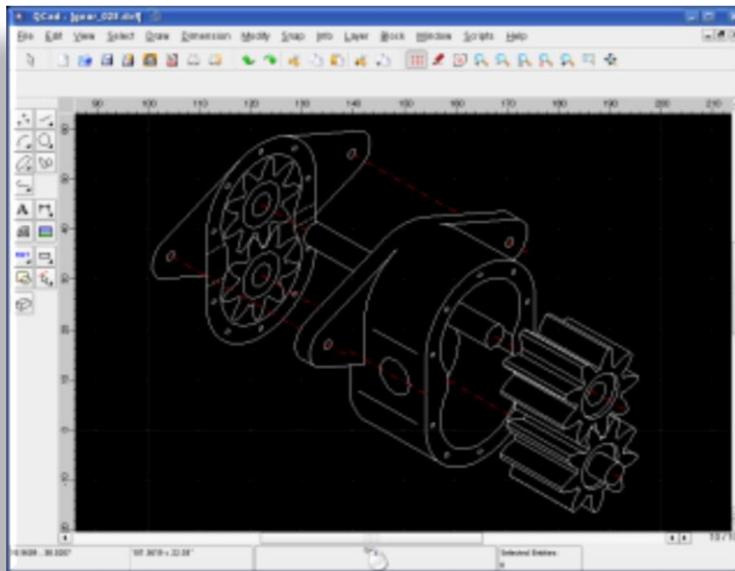


Fig. 3.2.1.2. Proyecto en CAM Expert.

vec2web es una utilidad de línea de comandos para los usuarios de QCAD que intentan automatizar la conversión de formatos. vec2web se basa en la misma tecnología que QCAD. vec2web lee dibujos en formato DXF y mapas de bits salidas y algunos otros formatos: BMP, GIF, JPEG, PNG, XPM, XBM,

PBM, PGM, PPM, Postscript y DXML. vec2web también puede utilizarse para imprimir directamente en la impresora predeterminada.

dxflib es una biblioteca de código abierto de C++, principalmente para analizar e importar archivos DXF. También puede escribir archivos DXF, pero hay que tener buen conocimiento del formato DXF para producir una salida válida.

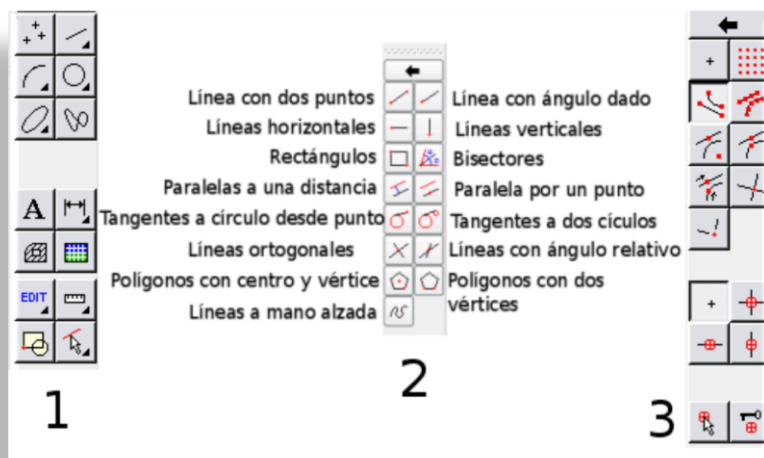


Fig. 3.2.1.3. Comandos para dibujar líneas.





Como sabemos, cualquier dibujo es la suma de muchas líneas, en la figura 3.2.1.3. Se muestran botones con los comandos principales para dibujar líneas. Vamos a ver la forma de dibujar una línea:

1. Seleccionar línea del panel izquierdo.
2. Seleccionar cómo dibujar las líneas, normalmente definida como:
 - ✓ Línea con dos puntos, definir punto inicial y final con el ratón.
 - ✓ Línea con un ángulo dado: definir el punto inicial, después su longitud y ángulo.
 - ✓ Línea horizontal: definir punto inicial y la longitud.
 - ✓ Línea vertical: definir punto inicial y la longitud.
3. Definir cómo seleccionar los puntos: puntos finales, intermedios... (*equivalente a SNAP*) y si corresponden a líneas horizontales y verticales.

Qcad es realmente fácil de usar con lo que se puede acostumbrar rápidamente a su uso. Se accede a Qcad desde el menú de aplicaciones o desde consola ejecutando la orden 'qcad'. Como ejemplo de las funciones básicas se explicará cómo hacer un hexágono donde haya un círculo dentro:



Fig. 3.2.1.4. Barra de coordenadas de Qcad

1. Abrir la aplicación y presionar el icono de 'Nuevo', saldrá la ventana principal de Qcad. En la parte inferior hay un espacio donde podemos introducir coordenadas en el

área de trabajo y esto nos permite tener más precisión a la hora de hacer un plano (Fig. 3.2.1.4. Barra de coordenadas de Qcad).





2. Lo primero que hay que hacer es posicionar el punto de inicio en el centro del área de trabajo, esto es introduciendo el comando de coordenadas '0,0'.
3. La primer línea será, partiendo del origen '0,0', escribir el comando de coordenadas '@100<0'. El símbolo de arroba (@) indica que la orden de coordenadas se realiza desde la última posición, seguido por la magnitud (en mm) y ángulo del vector (Fig. 3.2.1.5. Primera línea de 100 mm y ángulo de 0°).

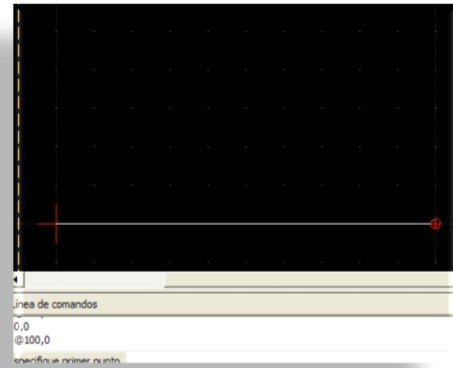


Fig. 3.2.1.5. Primera línea de 100 mm y ángulo de 0°.

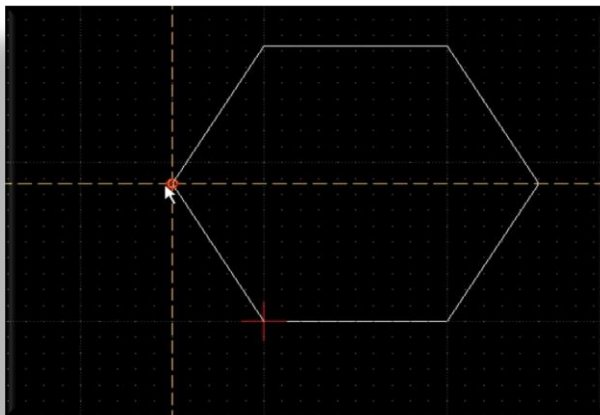


Fig. 3.2.1.6. Hexágono terminado.

4. Se sabe que un hexágono tiene sus ángulos internos de 60°. A partir de aquí, en la línea de comandos de coordenadas, se escribe '@100<60', para hacer otra línea de 100 mm, pero con un ángulo de 60°; después escribir '@100<120', y así hasta completar los 360°; el resultado final será el hexágono (Fig. 3.2.1.6. Hexágono terminado).
5. Para realizar el círculo dentro del hexágono, se accede al menú de círculos, seleccionar 'Dibujar círculos a 2 puntos', se da clic en el ícono 'seleccionar puntos medios' (Fig. 3.2.1.7. Ícono para hacer selección de puntos medios.).



Fig. 3.2.1.7.





6. Se mueve el cursor a la mitad de alguna de las líneas y se da clic ahí, posteriormente se da clic en la parte media de la línea contraria y se dibujará un círculo, ocupando la mayor área posible del hexágono (Fig. 3.2.1.8. *Hexágono con un círculo dentro.*).

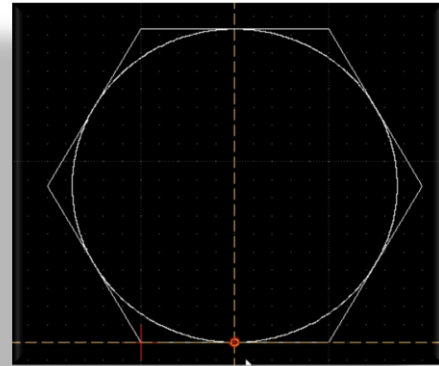


Fig. 3.2.1.8. *Hexágono con un círculo dentro.*

Instalación.

QCAD está en los repositorios de Ubuntu, lo único que hay que hacer es instalarlo por el medio que más se nos facilite (Ver 2.11. *Instalar aplicaciones.*). En consola, `'sudo aptitude install qcad'`; nos instalará todas las dependencias necesarias.

La interfaz de usuario de QCAD se integra mejor en KDE, aunque no hay ningún inconveniente en instalarlo en otros entornos de escritorio como GNOME, XFCE, LXDE; etc. Eso sí, se instalaran algunas librerías propias de KDE con las dependencias.

Sólo está desarrollada y soportada una versión para arquitecturas de 32 bits, por ahora no hay versiones propias para sistemas de 64 bits.

3.2.2. BlenderCAD (*Modelado 3D*).

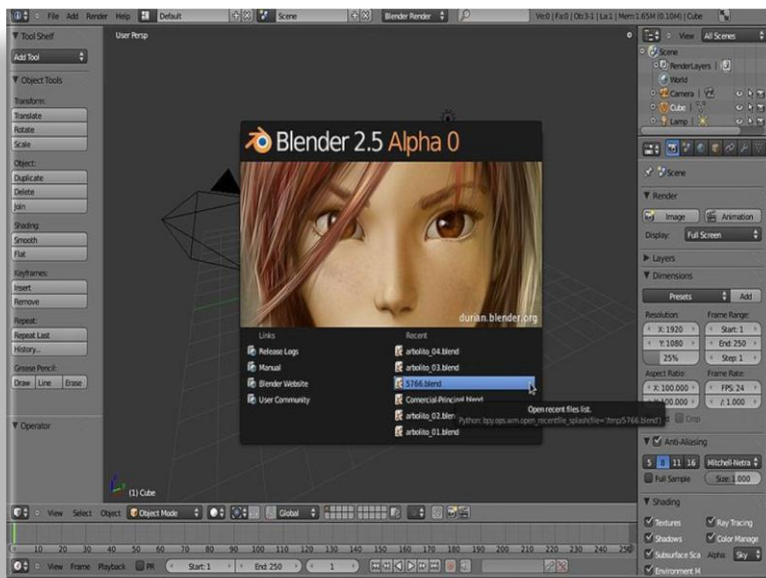
Blender es un programa informático multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales. El programa fue inicialmente distribuido de forma gratuita pero sin el código fuente, con un manual disponible para la venta, aunque posteriormente pasó a ser software libre.





Actualmente es compatible con todas las versiones de Windows, Mac OS X, Linux, Solaris, FreeBSD e IRIX.

Tiene una muy peculiar interfaz gráfica de usuario (*GUI*), que se critica como poco intuitiva, pues no se basa en el sistema clásico de ventanas (*Fig. 3.2.2.1. GUI de Blender.*); pero tiene a su vez ventajas importantes sobre éstas, como la configuración personalizada de la distribución de los menús.



Siendo una herramienta relativamente nueva, ha gozado de la aceptación de muchos animadores independientes. En la industria de Generación de gráficos avanza como un proyecto prometedor. Existen proyectos actuales que han empezado a usarlo profesionalmente:

Fig. 3.2.2.1. GUI de Blender.

El 18 de febrero de 2010 se estrenó el primer largometraje animado realizado íntegramente con software libre, usando a Blender como principal herramienta; se trata de '*Plumíferos*', proyecto que está impulsando el desarrollo de Blender aún más, sobre todo a nivel de animación y manejo de librerías a gran escala.

Películas tales como *SpiderMan 2* que lo ha usado para hacer una previsualización de escenas (*Screen-Board Test*), han usado de manera incipiente las capacidades del popular programa GNU/GPL.





Blender tiene un largo camino por recorrer, tiene bases que garantizan su desarrollo, se espera que en un futuro no muy lejano, dentro de los ámbitos profesionales más renuentes al uso de software libre, este programa sea un embajador muy adecuado del mismo.

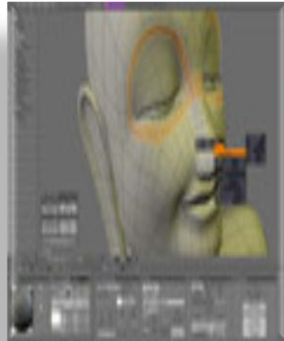
Algunas de sus muchas características son:

- ✓ Multiplataforma, libre, gratuito y con un tamaño de origen realmente pequeño comparado con otros paquetes de diseño 3D.
- ✓ Capacidad para una gran variedad de primitivas geométricas, incluyendo curvas, mallas poligonales, vacíos, NURBS, metaballs.
- ✓ Junto a las herramientas de animación se incluyen cinemática inversa, deformaciones por armadura o cuadrícula, vértices de carga y partículas estáticas y dinámicas.
- ✓ Edición de audio y sincronización de video.
- ✓ Características interactivas para juegos como detección de colisiones, recreaciones dinámicas y lógicas.
- ✓ Posibilidades de renderizado (*proceso de generar una imagen desde un modelo*) interno versátil e integración externa con potentes trazadores de rayos o '*raytracer*' libres como kerkythea, YafRay o Yafrid.
- ✓ Lenguaje Python para automatizar o controlar varias tareas.
- ✓ Blender acepta formatos gráficos como TGA, JPG, Iris, SGI, o TIFF. También puede leer ficheros Inventor.
- ✓ Motor de juegos 3D integrado, con un sistema de ladrillos lógicos. Para más control se usa programación en lenguaje Python.
- ✓ Simulaciones dinámicas para cuerpos suaves, partículas y fluidos.
- ✓ Modificadores apilables, para la aplicación de transformación no destructiva sobre mallas.





En las figuras 3.2.2.2 a), b), c); se aprecian algunas de los trabajos que se pueden hacer en Blender:



a)



b)



c)

Figs. 3.2.2.2. a), b), c). Modelado 3D, animación y renderizado de imágenes.

- ✓ Blender es completamente flexible y configurable, diseño de la ventana con muchas configuraciones como usted prefiera.
- ✓ Temas personalizados.
- ✓ Animación.
- ✓ Armadura (esqueleto) con la deformación de avance y cinemática inversa.
- ✓ Sistema de limitación de animación.
- ✓ Editor de Animación de personajes.

Blender puede leer y escribir en muchos formatos de imágenes en 2D y 3D:

- ✓ **2D.** TGA, JPG, PNG, OpenEXR, DPX, Cineon, Radiance HDR, Iris, SGI Movie, IFF, GIF, TIFF, PSD, MOV (*Windows and Mac OS X*).
- ✓ **3D.** 3D Studio, AC3D, COLLADA, FBX Export, DXF, Wavefront OBJ, DEC Object File Format, DirectX, Lightwave, MD2, Motion Capture, Nendo, OpenFlight, PLY, Pro Engineer, Radiosity, Raw Triangle, Softimage, STL, TrueSpace, VideoScape, VRML, VRML97, X3D.





Blender es soportado en los sistemas operativos:

- ✓ Windows XP, Vista, 7. (32/64 bits)
- ✓ Mac OS X (PPC e Intel).
- ✓ Linux (x86-32/64 bits y PPC) .
- ✓ FreeBSD (i386).
- ✓ SGI Irix.
- ✓ Sun Solaris (sparc y x86).

Especificaciones de hardware mínimos:

- ✓ 300 MHz CPU.
- ✓ 128 MB RAM.
- ✓ 20 MB de espacio libre en disco duro.
- ✓ Pantalla 1024 x 768 px con 16 bit color.
- ✓ Tarjeta gráfica Open GL con al menos 16 MB RAM.

Especificaciones de hardware recomendadas:

- ✓ 2 Ghz dual CPU.
- ✓ 2 GB RAM.
- ✓ Pantalla 1920 x 1200 px con 24 bit color.
- ✓ Tarjeta gráfica Open GL con 128 ó 256 MB RAM.

Especificaciones de hardware para sistemas de producción profesional:

- ✓ 64 bits, Quad core CPU.
- ✓ 8 GB RAM.
- ✓ Pantalla 1920 x 1200 px con 24 bit color.
- ✓ Tarjeta gráfica Open GL con 768 MB RAM, ATI FireGL o Nvidia Quadro.





Instalación.

El paquete de Blender está en los repositorios de Ubuntu, lo único que tenemos que hacer es instalarlo por el medio que más se nos facilite (*Ver 2.11. Instalar aplicaciones.*).

En consola, *'sudo aptitude install blender'*; nos instalará todas las dependencias necesarias.

Otra forma de realizar la instalación, si es que queremos la última versión de desarrollo, es descargar el código fuente de la página oficial de Blender y compilarlo para nuestra arquitectura.

El archivo lo podemos obtener tecleando en la consola el comando:

```
'wget http://www.blender.org/dl/http://ftp.nluug.nl/pub/graphics/blender/release//Blender2.54beta/blender-2.54-beta-linux-glibc27-i686.tar.bz2', esto para un sistema de 32 bits.
```

En el caso que tengamos un sistema de 64 bits tecleamos el comando:

```
'wget http://www.blender.org/dl/http://ftp.nluug.nl/pub/graphics/blender/release//Blender2.54beta/blender-2.54-beta-linux-glibc27-x86_64.tar.bz2'.
```

Una vez descargado el archivo fuente, lo desempacamos en el directorio que queramos y dentro de la carpeta principal viene un archivo llamado *'README'*, en este archivo se indica el proceso y los requerimientos antes de compilarlo e instalarlo.

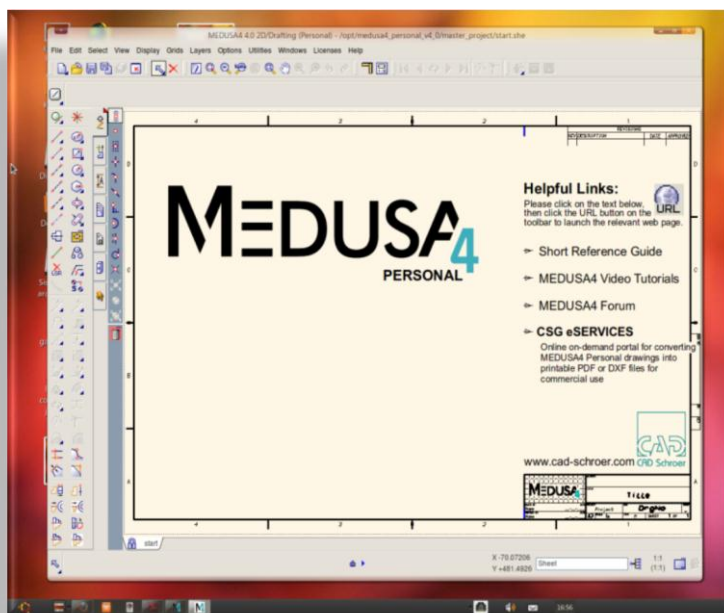




3.2.3. Medusa4 (2D y 3D).

De sobra nos es conocido que el 'rey' indiscutible del CAD es AutoCad, ningún programa ha sido capaz de hacerle sombra, aunque los precios del software son realmente abusivos. Además tiene el problema que no presentan versiones en Linux, aunque el futuro está en la adaptación del software a Linux (*por su creciente demanda en el mercado*) y en la virtualización. No obstante, nos encontramos con otros programas CAD libres con los que podemos desarrollar nuestros proyectos, es el caso de Medusa4.

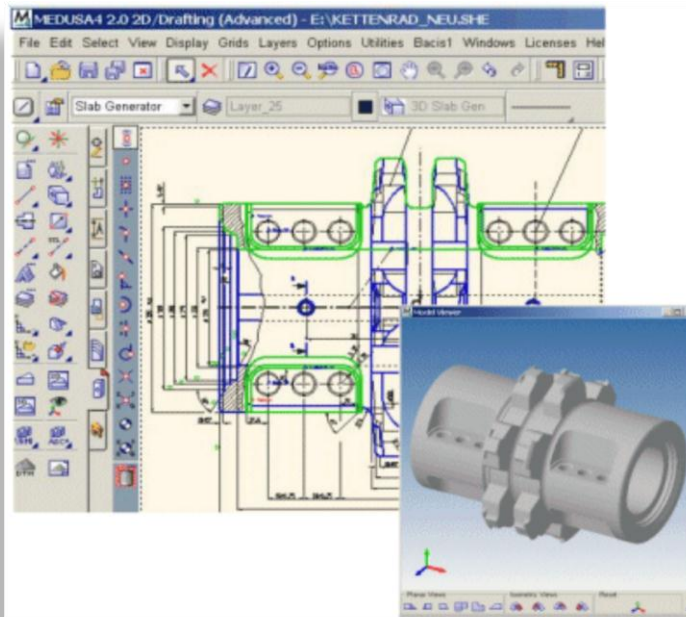
Medusa4 es un software para diseño asistido por computadora, creado por la empresa CAD Schroer. Se ofrece en una versión libre para uso personal, de cuyas características e instalación se describen a continuación. Medusa4 permite trabajar tanto con archivos en formato 'dxf' como en 'dwg' y es posible tanto dibujos en 2D como en 3D.



Los comandos de dibujo se realizan seleccionando entre los menús desplegables desde el lado izquierdo y sólo tenemos que trabajar de una forma un poco diferente a AutoCad, aunque las posibilidades del programa son muy buenas (Fig. 3.2.3.1. *Interfaz de usuario.*).

Fig. 3.2.3.1. Interfaz de usuario.





La cuarta generación de la suite de software de diseño mecánico MEDUSA ofrece un diseño avanzado dentro de un entorno de sistemas altamente flexibles. Sus potentes herramientas de fueron desarrollados para funcionar de la manera que necesitan los ingenieros (*En la Fig. 3.2.3.2. Se muestra un ejemplo de lo que podemos hacer.*).

Fig. 3.2.3.2. Ejemplo de diseño en 2D y 3D.

MEDUSA4 está disponible en todas las principales plataformas, incluyendo Windows, Solaris y Linux, y es altamente personalizable para adaptarse a las necesidades específicas de la industria (*Fig.3.2.3.3.*).

La amplia familia del producto incluye un potente sistema de diseño 2D/3D. Es altamente compatible con sistemas de terceros. Las piezas estándar e interfaces de programación CAD están también disponibles.

Instalación.

Para instalar el programa se debe registrar en la siguiente página: '<https://calltracking.cad-schroer.com/portal/com/Log3>'. Una vez registrado, el sistema permitirá descargar el archivo de instalación que está en formato '.sh'.





Características	Beneficios
Fácil integración con sistemas de terceros – completo soporte para la interfaz: Pro/ENGINEER [®] , Windchill [®] PDMLink [™] , mySAP PLM [®] , Teamcenter, STHENO/PRO [®] , AutoCAD [®] DXF/DWG, DXF 3D, VDA, STEP, IGES, STL, COLLADA [™] (Google Earth) VRML, VDA.	<ul style="list-style-type: none">✓ Lograr mayores niveles de servicio y de productividad a través de una fácil colaboración con clientes y proveedores.✓ Reducir los tiempos muertos del proyecto.✓ Lograr eficiencia en el diseño final.✓ Reducir los costos mediante la consolidación de sistemas y bases de datos.✓ Acelerar el tiempo de mercado
Altamente personalizable.	✓ Obtener ventaja en su nicho de mercado a través de la utilización de aplicaciones específicas.
Eficiente e intuitiva interfaz de usuario.	✓ Reducir los periodos de entrenamiento de los usuarios al nuevo software.

Fig. 3.2.3.3. Algunas características de Medusa4 y sus respectivos beneficios.

Hay que recordar que este tipo de archivos no instala las dependencias necesarias, así que antes de instalarlo, se debe instalar las dependencias que nos indica el desarrollador:

- Se abre una terminal.
- Escribir `'sudo aptitude install build-essential csh libstdc++5 libg2c0 gcc-3.4-base'`
- Después de que las dependencias se han instalado, ya se puede comenzar a instalar el programa, ejecutando, desde el lugar en que se descargó el archivo de instalación, el siguiente comando: `'sudo sh medusa4_v4_0_0_linux_personal.sh'`.

Ahora comienza la instalación del programa, contestando la información requerida. Cuando se llega a la pantalla de *'License Information'* y pregunta: *'Do you already have a valid MEDUSA4 Personal license?'* seleccionar **No**.





En la pantalla siguiente, copiar el primer número de MAC Address y pulsando el botón 'Next' se continúa con la instalación, hasta finalizar. Para ejecutar el programa, se necesita una licencia que se obtiene de forma gratuita accediendo a la página de Medusa4:

'<http://www.cad-schroer.com/index.php?screen=1.3&ziel=Products-MEDUSA-M4Personal&land=com>'

Y ahora se pulsa en STEP 1: 'Request your FREE personal license' y se rellena la información requerida, en MAC Address se pone el valor que se copió. De esta forma se recibirá un correo electrónico, que se responderá al remitente sin añadir nada, en seguida, se obtiene como respuesta un correo que incluye el archivo con la licencia, del tipo PLXXXXX.lic, que se descargas y se guarda.

Se otorgará una licencia por 6 meses totalmente gratis, al término se podrá renovar la licencia y seguirá siendo gratis. Ahora se abre la terminal y se ejecuta:

```
$ sudo nautilus
```

Con esto, se abre el navegador de archivos como 'root', se dirige al lugar donde se guardó el archivo de licencia y cambiar el nombre a 'csglmd.lic', se copia y se dirige, con el navegador, a 'opt/medusa4_pers_v4_0/medsys/med/license', se encuentra en ese lugar otro archivo llamado 'csglmd.lic', que se borra y se pega el nuevo.

Ya se tiene todo instalado, ahora para ejecutarlo de forma cómoda, dirigirse a 'Sistema > Preferencias > Menú principal', en el panel izquierdo seleccionar 'Gráficos' y pulsar 'Elemento nuevo'. Se abre la configuración del lanzador (Fig. 3.2.3.4.) y rellenar con la siguiente información:



Fig. 3.2.3.4. Crear Lanzador para el Menú.





- ✓ Tipo: Aplicación en terminal
- ✓ Nombre: Medusa4
- ✓ Comando:
`/opt/medusa4_pers_v4_0/master_project/startmedusa`

Pulsar el botón de la parte superior izquierda para seleccionar el icono que se desee, para identificar a la aplicación en el menú.

Al terminar pulsar el botón 'Cerrar'. De esta forma, en el menú Aplicaciones > Gráficos seleccionar Medusa4, para utilizar el programa.

3.2.4. Otras herramientas CAD.

3.2.4.1. BricsCAD.

Es una alternativa que ofrece una característica completa y potente, principalmente dirigida para los usuarios profesionales. Bricscad lee y escribe en formato DWG y ofrece una alta compatibilidad con AutoCAD. El software es desarrollado por la empresa Bricsys, la cual está plenamente comprometida a servir a la creciente comunidad de Linux. Una versión nativa para Linux de la tecnología V10 BricsCAD está en su etapa final de desarrollo. '<http://www.bricsys.com/>'.

3.2.4.2. FreeCAD.

FreeCAD es un básico 3D CAD con capacidades avanzadas de simulación de movimiento. A los educadores, estudiantes y nuevos ingenieros les resultará ideal para la enseñanza y el aprendizaje de la geometría, cinemática, dinámica, vibraciones, mecanismos, las levas, el diseño de máquinas y la física.





El programa se ejecuta en Windows, Macintosh PPC, PC Linux y Unix.
'<http://askoh.com/freecad/>'.

3.2.4.3. VariCAD.

VariCAD es un sistema CAD para ingeniería mecánica. Incluye funciones para el modelado 3D, dibujo 2D, edición, transformaciones, trabajar con objetos definidos por el usuario, bloque, grupo y la creación de capacidades símbolo. Todo el sistema es muy personalizable y abierto ampliamente. '<http://www.varicad.com/>'.

3.3. Aplicaciones EDA (*Electronic Design Automation – Diseño Electrónico Automatizado*).

3.3.1. gEDA.

El proyecto gEDA se inició debido a la falta de herramientas EDA libres para sistemas UNIX, con el propósito principal de promover el estado de '*hardware libre*' o '*hardware de código abierto*'. La suite ha sido principalmente desarrollada en la plataforma GNU/Linux, con un poco de esfuerzo de desarrollo, se ha tratado de portar a otras plataformas también.

gEDA es una suite de aplicaciones de software libre EDA (*Electronic Design Automation*) usada para diseño de circuitos eléctricos, captura esquemática, simulación, creación de modelos y producción.

Actualmente, el proyecto gEDA ofrece un conjunto maduro de aplicaciones de software libre para diseño electrónico, incluye aplicaciones de software para varios propósitos en el diseño de circuitos eléctricos, entre las que se encuentran:





- ✓ Administrador de proyectos.
- ✓ Capturador esquemático.
- ✓ Administrador de atributos.
- ✓ Un generador de netlist.
- ✓ Una herramienta para el diseño de circuitos impresos (*PCB*).
- ✓ Simuladores de circuitos análogos y digitales.
- ✓ Compiladores/simuladores Verilog HDL.
- ✓ Visualizadores de formas de onda.
- ✓ Generador de lista de materiales (BOM - Bill Of Materials).

Para entender cómo están integradas las aplicaciones en la suite gEDA debemos tener claro que gEDA no funciona como un entorno integrado de desarrollo, en una sola ventana, gEDA está diseñada con la tradición de las aplicaciones de los sistemas operativos tipo UNIX, cada una hace una tarea en específico y la hace bien. Las principales aplicaciones asociadas a gEDA son:

Administrador de proyectos.

geda es el administrador de proyectos gráfico, es necesario aclarar que la palabra gEDA (*g en minúscula y EDA en mayúscula*) se refiere a la suite en general y geda se refiere al administrador de proyectos gráfico.

Por medio de esta herramienta es posible organizar los archivos y llamar algunas otras herramientas.

En un principio el administrador de proyectos va a ser útil para ubicarse en la forma como se trabaje con gEDA.





Capturador esquemático.

gschem (Fig. 3.3.1.1.) es el programa de captura esquemática de la suite, por medio de él se puede introducir componentes gráficos para luego realizar distintas operaciones sobre el archivo que el 'capturador' genera.

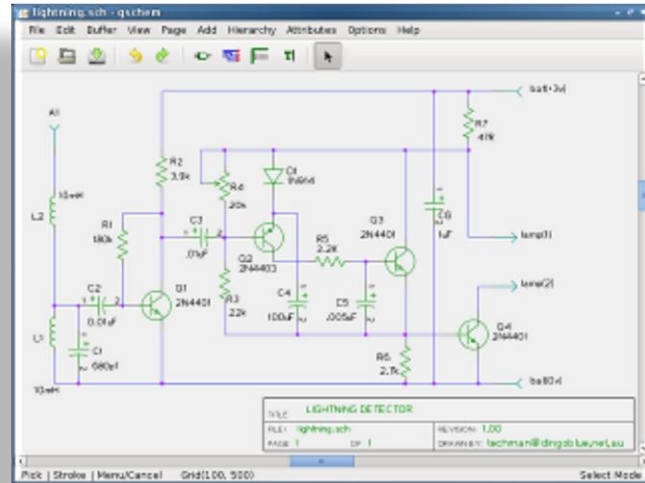


Fig. 3.3.1.1. Capturador esquemático – gschem.

Al referirnos a captura esquemática hablamos del diseño y dibujo de circuitos usando la computadora. Así por medio de dibujos se representa el funcionamiento de cierto componente, compuerta o un circuito complejo.

gEDA cuenta con una biblioteca de símbolos con más de 1400 símbolos, todos bajo GPL, que hace que gschem sea muy poderoso.

Editor de atributos.

gattrib es el editor de atributos de la suite gEDA. Su función es crear, a partir de los archivos esquemáticos (.sch), la lista de componentes del diseño. Así es posible agregar, modificar o eliminar los atributos de los componentes sin tener que estar en el capturador esquemático para luego guardar esas modificaciones.

Por lo tanto su utilidad se encuentra cuando se necesita cambiar los atributos de muchos componentes a la vez ya que evita la molestia de cambiar atributos componente a componente.





Generador de netlist.

gnetlist es un programa que genera los *'netlist'* (forma de cómo se encuentran conectados los componentes dentro de una estructura y las señales que propicia esa interconexión) tomando como argumentos el archivo del capturador esquemático (.sch) y los archivos de los símbolos empleados (.sym).

Actualmente el programa puede generar *'netlist'* en formatos como spice, verilog o VHDL entre otros.

Simuladores.

gnucap es un simulador de circuitos de propósito general, puede realizar análisis no lineal en CD y transitorio, análisis de Fourier y análisis CA.

El simulador es compatible con modelos de spice. GNUCAP es acrónimo de GNU Circuit Analysis Package.

ngspice es un simulador de circuitos basado en los simuladores de código abierto Spice3f5, Cider1b1 y Xspice.

PCB (Printed Circuit Board - Tarjeta de circuito impreso).

PCB (Fig. 3.3.1.2.) es un editor de PCB. En electrónica, PCB es un medio para sostener mecánicamente y conectar eléctricamente componentes electrónicos, a través de rutas o pistas de material conductor, grabados en hojas de cobre laminadas sobre un sustrato no conductor, comúnmente baquelita o fibra de vidrio.



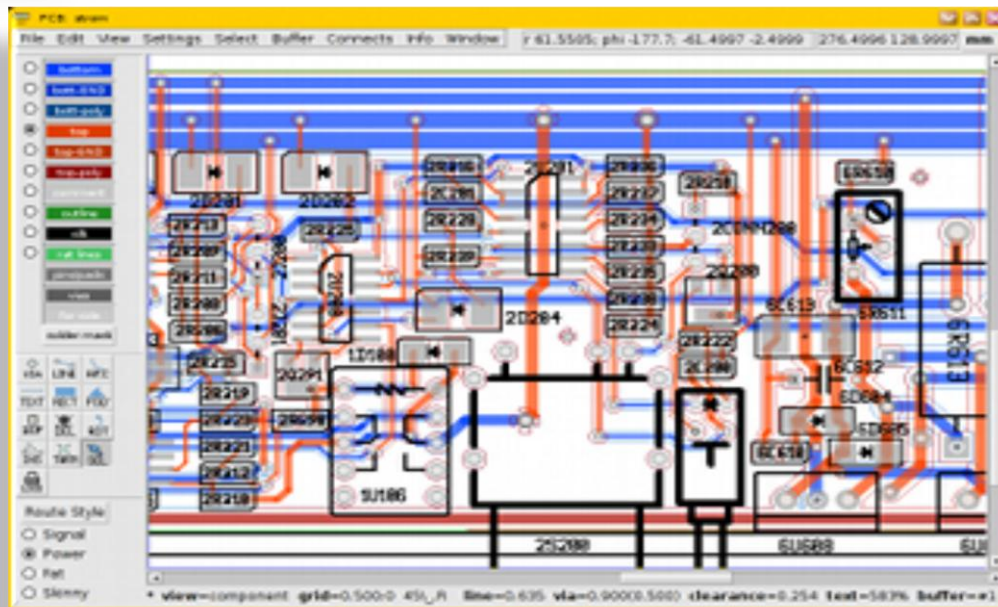


Fig. 3.3.1.2. PCB Editor.

Gerbv (Fig.3.3.1.3.) es un visor de archivos 'Gerber'. El formato Gerber contiene la información necesaria para la fabricación de la placa de circuito impreso o PCB. Se pueden crear con distintos programas de diseño electrónico como Eagle, Protel o Altium Designer.

Instalación.

El paquete de gEDA está en los repositorios de Ubuntu, lo único que se tiene que hacer es instalarlo por el medio que se desee (Ver 2.11. Instalar aplicaciones.).

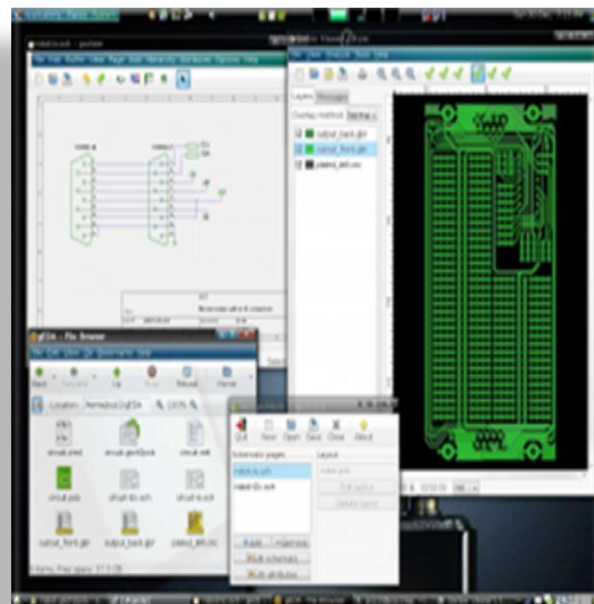


Fig. 3.3.1.3. gerv – Visor de archivos gerber.





En consola, '*sudo aptitude install geda geda-utils geda-xgsch2pcb pcb*'; instalará la suite completa de gEDA, con todas las dependencias necesarias.

3.3.2. Alliance (Diseño VLSI.).

Alliance. Es un conjunto de herramientas y bibliotecas para el diseño de sistemas electrónicos VLSI. Incluye un compilador y simulador de lenguaje VHDL, herramientas de síntesis; así como un completo conjunto de bibliotecas CMOS, basadas en MOSIS, que proporciona los estándares en los procesos de fabricación de circuitos integrados.

Alliance es el resultado de un esfuerzo de doce años de investigación y desarrollo en el departamento de ASIM de la Pierre et Marie Curie University, en París.

El programa Alliance contiene:

- ✓ Más de 60 bibliotecas escritas en lenguaje C. Estas bibliotecas contienen la definición de estructuras de datos y métodos que permiten describir los circuitos.
- ✓ Más de 650 páginas describiendo la interface de programación de esas bibliotecas.

Alliance proporciona herramientas CAD y EDA que cubren la mayoría de los pasos en el flujo de diseño digital:

- ✓ VHDL Compilación y Simulación.
- ✓ El chequeo de modelo y la prueba formal.
- ✓ RTL y Síntesis Lógico.
- ✓ Compilación Data-Path.
- ✓ Generación de Macro-cells.





- ✓ Place and route (lugar y ruta).
- ✓ Edición de Layout (disposición o acomodo de componentes).
- ✓ Extracción y verificación del Netlist.
- ✓ DRC (Design Rules Checking – Verificación de las reglas de diseño).

Algunas características de Alliance son:

Independencia del proceso. El objetivo principal es el diseño digital. La Independencia del proceso se basa en un enfoque de diseño simbólico y un mapeo de proceso totalmente automático, desde un proveedor de silicio a otro. Las bibliotecas existentes portables están diseñadas para el proceso estándar CMOS con un poli y dos metales.

Portabilidad del software. El paquete de Alliance ha sido diseñado para ejecutarse en estaciones de trabajo de bajo costo, basados en PCs bajo Linux con fines educativos, pero se puede aprovechar la enorme capacidad de memoria de hoy en día en supercomputadoras Unix para proyectos de investigación avanzada. La mayoría de programas se pueden utilizar en una terminal de texto.

Modularidad. Cada herramienta puede funcionar como un programa independiente. Puede, por ejemplo, utilizar sólo el simulador VHDL o el compilador. La facilidad y sencillez se han preferido a los enfoques sofisticados. (Fig. 3.3.2.1. Editor de Layout.).

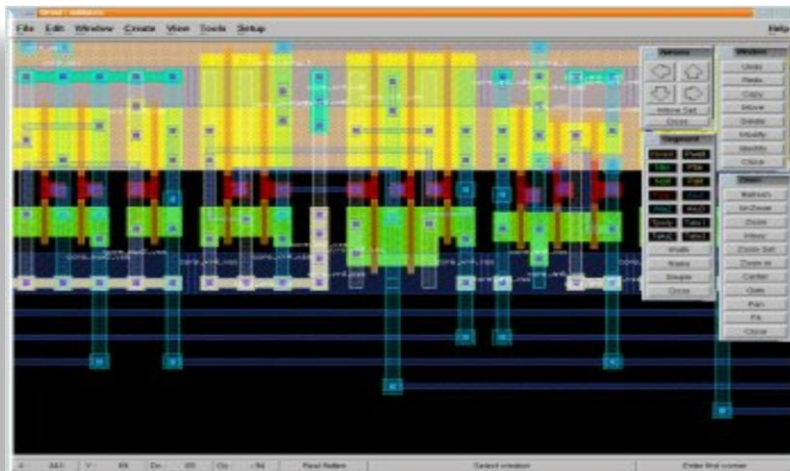


Fig. 3.3.2.1. Editor de Layout.



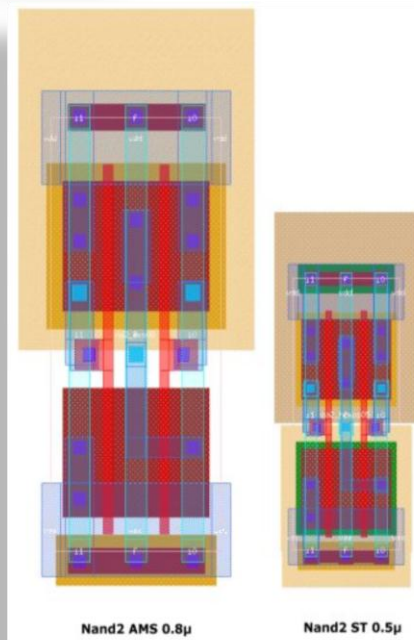


Fig.3.3.2.2. Layout de una compuerta NAND

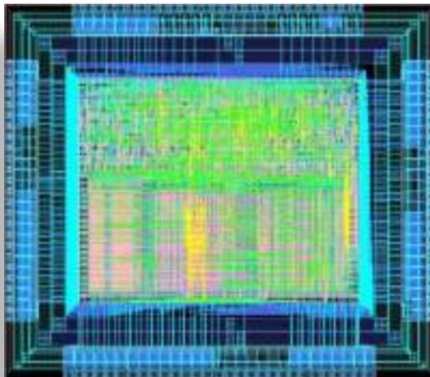


Fig. 3.3.2.3. Layout de un Circuito Integrado.

Interoperabilidad: Gracias a los formatos estándar utilizado para los archivos externos, Alliance es un software de código abierto. Puede utilizar la mayoría de las herramientas con otras herramientas de diseño de VLSI, tales como Cadence. En las figuras: 3.3.2.2. Y 3.3.2.3. Se aprecian los layouts de una compuerta NAND de dos entradas y de un circuito integrado completo, respectivamente.

El conjunto de herramientas soporta un flujo de diseño '*arriba abajo*' desde cero, con no sólo herramientas de construcción (*editor de diseño, síntesis lógica, place and route automática*), sino también las herramientas de validación de diseño.

Instalación.

Para GNU/Linux se ofrece el programa para arquitecturas de 32 bits y 64 bits. Los cuales se pueden descargar de su página de internet: '[http:// www-asim.lip6.fr / pub / alliance / distribution / 5.0 /](http://www-asim.lip6.fr/pub/alliance/distribution/5.0/)'.

Hasta el momento no se ofrece un paquete '*debian*' para su instalación en Ubuntu, por consiguiente, tampoco se tiene en los repositorios; por lo que hay que descargarlo de internet y compilarlo.





Para instalar se hace lo siguiente:

- ✓ Se descarga el código fuente de la página.
- ✓ Se desempaqueta en el directorio que se quiera.
- ✓ Mediante la consola se sitúa en el directorio del paquete fuente, esto es mediante el comando:
 - `'cd /home/usuario/dir_de_la_carpeta'`.
- ✓ Situados en el directorio escribir en consola:
 - `export ALLIANCE_TOP=/opt/Alliance`
 - `./configure --prefix=$ALLIANCE_TOP`
 - `gmake install`

3.3.3. KICAD.

Kicad es un programa de código libre (*GPL*) para la creación de esquemas electrónicos y circuitos impresos.

Concebido y creado por Jean Pierre Charras, investigador en el LIS (*Laboratorio de Imágenes y Señales*) y profesor de electrónica en el Instituto Universitario de Tecnología de Saint Martin d'Hères, en Francia. La suite Kicad es un conjunto de cuatro programas y un gestor de proyectos para realizar circuitos electrónicos:

- ✓ Eeschema: Creación de esquemas.
- ✓ PcbNew: Realización de circuitos impresos.
- ✓ Gerbview: Visualización de documentos generados en formato Gerber.
- ✓ Cvpcb: Utilidad de selección de las huellas físicas de los componentes electrónicos utilizados en el esquema.
- ✓ Kicad: Gestor de proyectos.



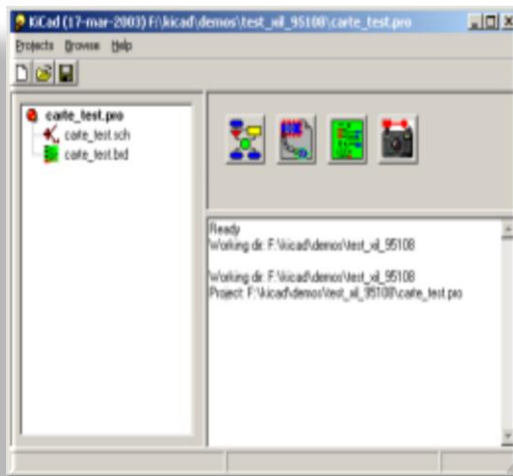


Fig. 3.3.3.1. Kicad (gestor de proyectos).

El gestor de proyectos, Kicad (Fig. 3.3.3.1), permite seleccionar un proyecto y abrir la herramienta deseada (Eeschema, PcbNew; etc.).

Esta suite de programas se distribuye gratuitamente en open source, con licencia GPL.

Es útil para cualquier persona que desee crear circuitos impresos.

Estos programas son multiplataforma, funcionan tanto en LINUX como en Windows y son actualizados regularmente. También ha sido probada en FreeBSD.

Eeschema (Fig. 3.3.3.2.) permite crear esquemas simples o en jerarquía (multihoja), el control de reglas eléctricas (ERC) y generar netlists para PcbNew, así como para la simulación con Spice.

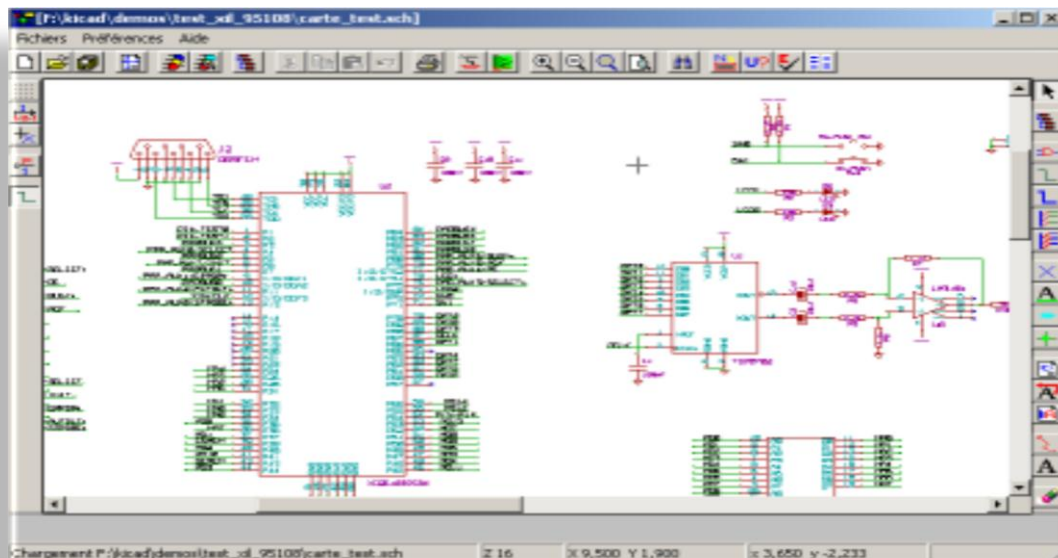


Fig. 3.3.3.2. Eeschema – Editor de esquemáticos.





El programa gestiona igualmente el acceso directo e inmediato a la documentación de componentes.

Pcbnew (Fig. 3.3.3.3.) es el programa de realización de circuitos impresos, trabaja con 1 a 16 capas y genera automáticamente todos los documentos necesarios para realizar los circuitos (*archivos GERBER de fototrazado, taladrado y colocación de componentes, así como los archivos de trazado PostScript para realizar prototipos*).

Pcbnew permite visualizar los circuitos y los componentes en 3 dimensiones (*utilizando OpenGL*).

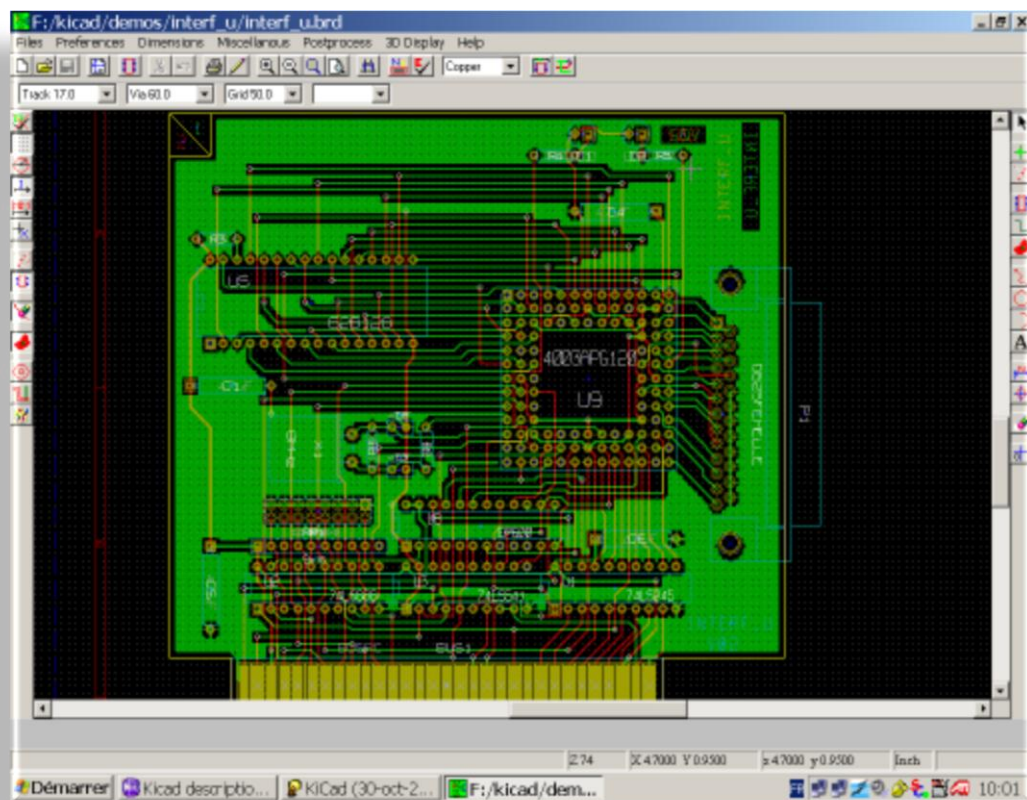


Fig. 3.3.3.3. PcbNew – Editor de PCB.



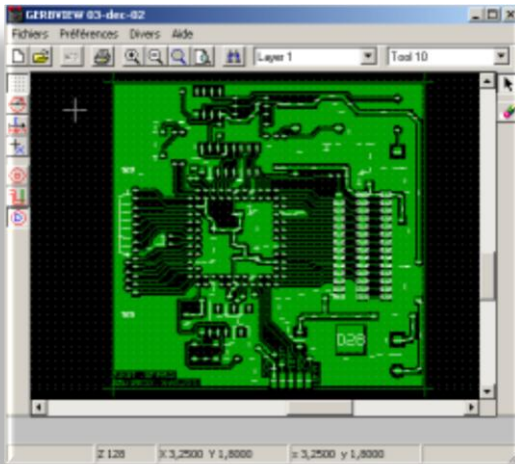


Fig. 3.3.3.4. Gerbview – Visor de archivos gerber.

Gerbview (Fig. 3.3.3.4.) es un visor de archivos 'Gerber'. El formato o tipo de archivo Gerber contiene la información necesaria para la fabricación de la placa de circuito impreso o PCB.

Kicad es un paquete muy fácil de usar (*para quien tiene conocimientos de electrónica*). Para ilustrar más claramente cómo usarlo, se presenta un ejemplo de circuito eléctrico.

Para ejecutar kicad, se hace clic en el menú '*aplicaciones*' > '*electrónica*' > '*kicad*' o desde la consola se escribe el comando '*kicad*'.

Aparece la pantalla principal de Kicad con cuatro menús desplegables (*en la barra de menús*) y cuatro botones en la parte superior del centro de la pantalla (*corresponden a los módulos de la suite: Eeschema, Cvpcv, PcbNew y Gerbview*) (Fig. 3.3.3.5. Pantalla principal Kicad.).

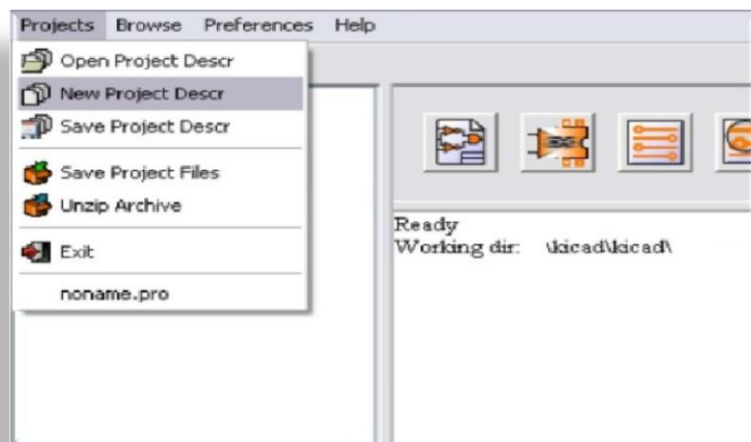


Fig. 3.3.3.5. Pantalla principal y menú de proyectos.

El procedimiento para trabajar es el siguiente:





1. Crear un nuevo proyecto, yendo al menú: 'projects' > 'New Project Descr' (Fig. 3.3.3.5.). Se introduce el nombre del proyecto que se desee con extensión '*.pro'. Automáticamente se crea un archivo con el mismo nombre, pero con extensión '*.sch' y aparece con estructura de árbol en el panel izquierdo de la pantalla principal.

2. Se hace doble clic sobre el nuevo archivo '*.sch'. Esto llama al módulo Eeschema (*editor de esquemáticos*).

3. A partir de aquí se realiza la edición del esquemático que se necesite. En la parte lateral derecha de la ventana de Eeschema se encuentra la barra de herramientas. Se hace clic en 'Add components' (Fig. 3.3.3.6. *Agregar componentes*).

4. Se hace clic en cualquier parte del área de trabajo en donde se quiera colocar el componente, la ventana 'component selection' aparece. Seleccionar 'List all' (Fig. 3.3.3.7. *Selección de componente*).

5. Aparece la ventana 'Select Lib'. Se tienen una gran cantidad de bibliotecas de componentes. Se hace doble clic sobre 'device' (Fig. 3.3.3.8. *Selección de biblioteca*).

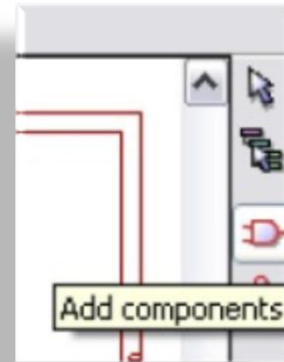


Fig. 3.3.3.6. Agregar componentes.



Fig. 3.3.3.7. Selección de componente

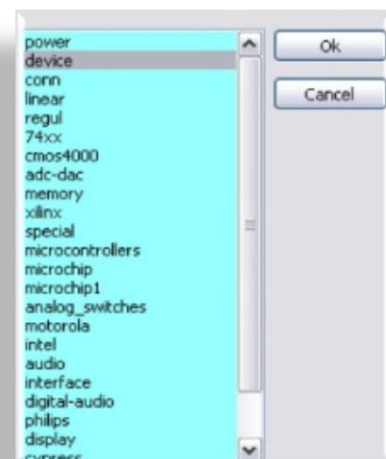


Fig. 3.3.3.8. Selección de biblioteca.





6. Aparece la ventana 'Select Part', en la que seleccionamos el tipo de dispositivo. Se puede desplazar la barra hacia abajo hasta encontrar, por ejemplo, 'R' (corresponde a una resistencia) (Fig. 3.3.3.9. Selección de dispositivo.).
7. La resistencia fue colocada en el lugar que se escogió, lo siguiente es editar su valor de resistencia. Hacer clic derecho en el centro del componente y seleccionar 'Edit component' > 'Edit' (Fig. 3.3.3.10. Editar componente.).



Fig. 3.3.3.9. Selección de dispositivo.

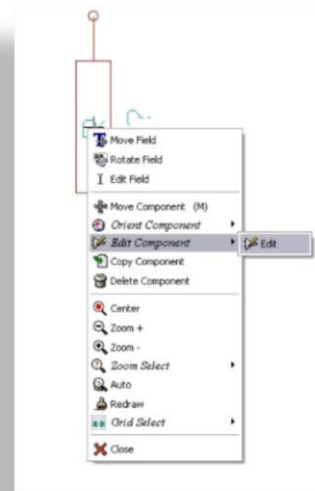
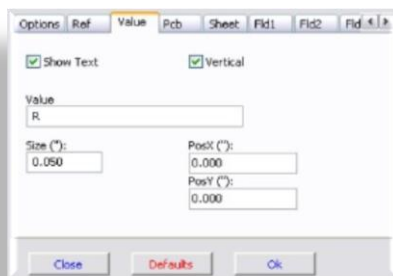


Fig. 3.3.3.10. Editar componente.



a)

b)

8. La ventana 'Component properties' aparece, seleccionar la pestaña 'Value'; cambiar 'R' por '1K' (Figs. 3.3.3.11. a) Asignación de valor del componente, b) Componente con valor asignado.).

Figs. 3.3.3.11. a) Asignación de valor del componente, b) Componente con valor asignado.





9. Para cualquier otro componente se siguen los mismos pasos, por ejemplo para agregar un LED 'Add component' > 'List All' > 'device' > 'LED'; para agregar un microcontrolador 'Add component' > 'List All' > 'microcontrollers' > 'PIC12C508A'; VCC 'Add component' > 'List All' > 'power' > 'VCC'; GND 'Add component' > 'List All' > 'power' > 'GND' (Fig. 3.3.3.12).

Resistencia, LED, microcontrolador, VCC, GND.).

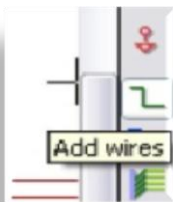


Fig. 3.3.3.13. Agregar cables

Fig. 3.3.3.12. Resistencia, LED, microcontrolador, VCC, GND.

10. Lo que hace falta es 'alambrar' el circuito, para ello se hace uso de la opción 'Add Wires' que está en la barra de herramientas a la derecha de la ventana (Fig. 3.3.3.13.

Agregar cables.).

Lo que se hace es dar clic sobre el pequeño círculo de cada pin y arrastrar, sin soltar, hasta el otro punto del componente que se quiere unir y entonces soltar (Fig. 3.3.3.14. Circuito interconectado.).

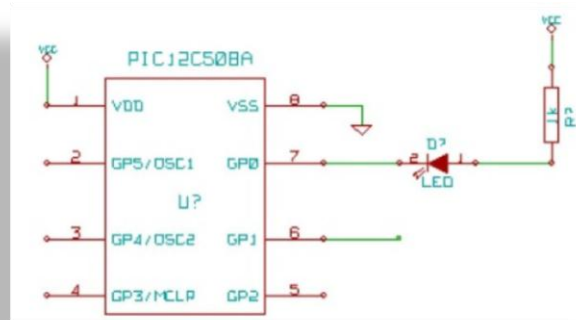


Fig. 3.3.3.14. Circuito interconectado.





11. Por último se verifica que no haya errores en el diseño, se hace clic en el botón 'Schematic Electric Rules Check' (Fig. 3.3.3.15. Verificación de las reglas eléctricas.).



Fig. 3.3.3.15. Verificación de las reglas eléctricas

Esto genera un reporte que informa acerca de cualquier error o advertencia, como cables que no estén conectados o en corto, si llegara a tener errores indica dónde y tipo de error.

Instalación.

El paquete de instalación de KICAD está en los repositorios de Ubuntu, se debe instalar por el medio que más se facilite (Ver 2.11. Instalar aplicaciones.).

En consola, '`sudo aptitude install kicad`'; esto instalará la suite completa de KICAD, con todas las dependencias necesarias.

Adicionalmente se pueden obtener muchas bibliotecas y ejemplos que pueden ser útiles en la siguiente página: '<http://www.kicadlib.org/>'.

3.3.4. Otras aplicaciones para electrónica.

3.3.4.1. GHDL.

Es una herramienta que permite compilar y simular la ejecución de código VHDL. Se trata de una herramienta muy sencilla por lo que la hace ideal para pruebas con el lenguaje o para tener un diseño previo antes de utilizarlo en otra herramienta como simulink. Sin embargo una desventaja grande es que no permite sintetizar código VHDL.





GHDL implementa el lenguaje VHDL de acuerdo con el estándar IEEE 1076-1987, el IEEE 1076-1993 y las normas posteriores. GHDL compila VHDL archivos y crea un archivo binario que simula su diseño. '<http://ghdl.free.fr/>'.

3.3.4.2. Piklab.

Piklab es un entorno de desarrollo integrado (*IDE*) para aplicaciones basadas en microcontroladores PIC y microcontroladores dsPIC similar al entorno MPLAB.

Se integra con el compilador y varias cadenas de herramientas ensamblador (como *gputils*, *sdcc*, *C18*) y con el simulador *gpsim*.

Es compatible con los programadores más comunes (*serial*, *paralelo*, *ICD2*, *Pickit2*, *PICStart* +), el depurador *ICD2*, gestores de arranque. '<http://piklab.sourceforge.net/>'.

3.3.4.3. VIPEC.

ViPEC es una poderosa herramienta para el análisis de alta frecuencia y de las redes eléctricas. Se necesita una descripción esquemática de la red eléctrica y realiza un análisis lineal en el dominio de la frecuencia.

La salida es de la forma de parámetros de puerto (*S*, *Y* o *Z*) con resultados que se presentan en una cuadrícula definida por el usuario. También se puede utilizar para calcular otras características como factores de estabilidad y retardo de grupo.

ViPEC admite varios elementos generales (*condensadores*, *resistencias*; *etc.*), así como las redes distribuidas como líneas de transmisión. '<http://vipec.sourceforge.net/>'.





3.4. Aplicaciones para matemáticas.

3.4.1. GNU Octave.

Octave o GNU Octave es un programa libre para realizar cálculos numéricos. Como indica su nombre es parte del proyecto GNU.

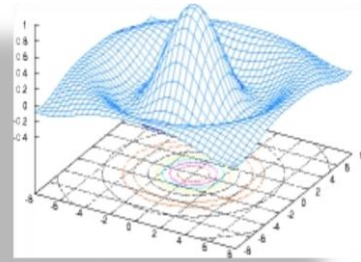


Fig. 3.4.1.1. Ejemplo de la aplicación Octave.

MATLAB es considerado su equivalente comercial. Entre varias características que comparten se puede destacar que ambos ofrecen un intérprete permitiendo ejecutar órdenes en modo interactivo.

El proyecto fue creado alrededor de 1988, pero con una finalidad diferente: ser utilizado en un curso de diseño de reactores químicos. Posteriormente en 1992, se decide extenderlo y comienza su desarrollo. La primera versión 'alpha' fue lanzada el 4 de enero de 1993. Un año más tarde, en 1994 aparece la versión 1.0.

GNU Octave tiene amplias herramientas comunes para resolver problemas de álgebra numérica lineal, encontrar las raíces de ecuaciones no lineales, integrar funciones ordinarias, manipular polinomios, e integrar ecuaciones diferenciales

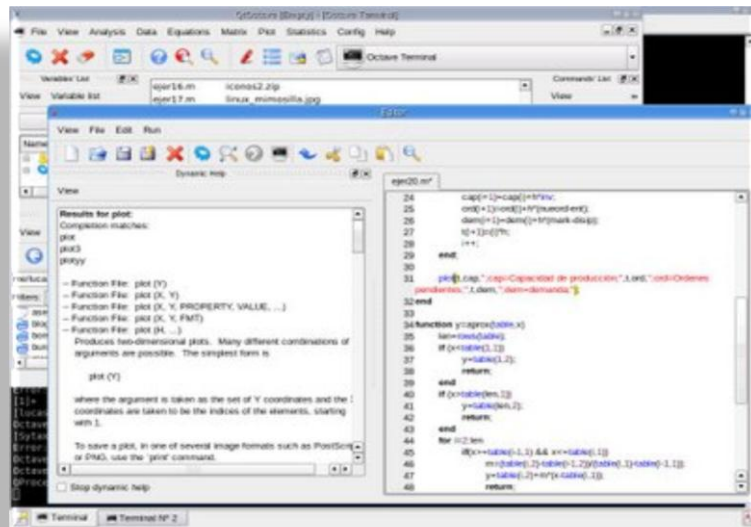


Fig. 3.4.1.2. GUI de Octave en un ejemplo de la aplicación.

ordinarias (Fig. 3.4.1.1. Ejemplo de la aplicación Octave y Fig. 3.4.1.2. GUI de Octave).





Es fácilmente extensible y personalizable a través de funciones definidas por el usuario por escrito en el idioma propio de Octave, o usando módulos cargados dinámicamente, escritos en C + +, C, Fortran, u otros idiomas.

Claramente, Octave es ahora mucho más que un simple paquete de cursos con una utilidad limitada más allá del aula. Hoy en día, miles de personas en todo el mundo están usando Octave en la enseñanza, investigación y aplicaciones comerciales.

Detalles técnicos del software Octave:

- ✓ Octave está escrito en C++ usando la biblioteca STL.
- ✓ Tiene un intérprete de su propio lenguaje (de sintaxis similar a Matlab), y permite una ejecución interactiva o por lotes.
- ✓ Puede extenderse el lenguaje con funciones y procedimientos por medio de módulos dinámicos.
- ✓ Utiliza otros programas GNU para ofrecer al usuario crear gráficos para luego imprimirlos o guardarlos.
- ✓ Dentro del lenguaje también se comporta como una consola de órdenes (*shell*). Esto permite listar contenidos de directorios, por ejemplo.
- ✓ Además de correr en plataformas Unix también lo hace en Windows.
- ✓ Puede cargar archivos con funciones de Matlab de extensión '*.m*'.
- ✓ Ayuda en español.

El lenguaje Octave:

- ✓ La sintaxis es similar a la utilizada en MATLAB.
- ✓ Es un lenguaje interpretado.
- ✓ No permite pasar argumentos por referencia. Siempre se pasan por valor.





- ✓ No permite punteros.
- ✓ Se pueden generar scripts.
- ✓ Soporta gran parte de las funciones de la librería estándar de C.
- ✓ El lenguaje está pensado para trabajar con matrices y provee mucha funcionalidad para trabajar con éstas.
- ✓ Soporta estructuras similares a los '*struct*'s de C.

GNU Octave es muy compatible con Matlab. Sin embargo, el programa de análisis de Octave va más allá de Matlab, por lo que los programas escritos para Octave no se ejecutan en Matlab.

Por ejemplo, Octave soporta el uso de comillas simples y dobles, Matlab sólo es compatible con comillas simples y las comillas dobles dan como resultado errores de análisis.

Los usuarios de Octave que deben colaborar con los usuarios de Matlab necesitan tomar nota de estas cuestiones.

Podemos tener acceso a una guía de Octave en la siguiente página: '<http://softwarelibre.unsa.edu.ar/docs/descarga/2003/curso/htmls/octave/x24.html>'.

Instalación.

El paquete de instalación de Octave está en los repositorios de Ubuntu, lo único que tenemos que hacer es instalarlo por el medio que más se nos facilite (*Ver 2.11. Instalar aplicaciones.*).

En consola, '*sudo aptitude install octave3.2 qtoctave gnuplot*'; esto nos instalará Octave y una interfaz gráfica para facilitar el trabajo.





3.4.2. Scilab.

Scilab es un programa para cálculo científico, interactivo de libre uso y disponible en múltiples sistemas operativos (Mac OS X, GNU/Linux, Windows) desarrollado por INRIA (Institut National de Recherche en Informatique et Automatique) y la ENPC (École Nationale des Ponts et Chaussées) desde 1990. Scilab es ahora desarrollado por Scilab Consortium dentro de la fundación Digiteo.

Scilab fue creado para hacer cálculos numéricos aunque también ofrece la posibilidad de hacer algunos cálculos simbólicos como derivadas de funciones polinomiales y racionales.

Posee cientos de funciones matemáticas y la posibilidad de integrar programas en los lenguajes más usados (*FORTRAN*, *Java* y *C* y *C++*). Scilab fue hecho para ser un sistema abierto donde el usuario puede definir nuevos tipos de datos y operaciones entre los mismos.

Scilab viene con numerosas herramientas: gráficos 2-D y 3-D, animación, álgebra lineal, matrices dispersas, polinomios y funciones racionales, simulación: programas de resolución de sistemas de ecuaciones diferenciales (*explícitas* e *implícitas*), control clásico y robusto, tratamiento de señales, grafos y redes.

Además se pueden agregar numerosas herramientas o toolboxes hechas por los usuarios como Grocer una herramienta para Econometría u Open FEM (*Una caja de Herramientas para Elementos Finitos*), hecha por INRIA.

En tanto las necesidades industriales y los avances científicos, Scilab cubre un amplio espectro de ámbitos: el sector aeroespacial, automoción, energía, defensa, finanzas, química, biología, medicina; etc.





Scilab incluye cientos de funciones matemáticas. Tiene un lenguaje de programación de alto nivel que permita el acceso a estructuras de datos avanzadas, 2-D y funciones gráficas en 3-D.

Características principales de Scilab:

- ✓ Cientos de funciones matemáticas.
- ✓ Lenguaje de alto nivel de programación.
- ✓ Gráficos 2-D y 3-D.
- ✓ Estructuras de datos avanzada y los tipos de datos definidos por el usuario.
- ✓ Xcos: modelador de sistemas dinámicos híbridos y simulador.

Visualización en 2D y 3D de (Fig.3.4.2.1.):

- ✓ Líneas
- ✓ Gráficos de exportación en muchos formatos: GIF, BMP, JPEG, SVG, PDF.
- ✓ Gráficos circulares
- ✓ Histogramas
- ✓ Superficies
- ✓ Animaciones

Computación numérica
(Fig.3.4.2.2):

- ✓ Álgebra lineal.
- ✓ Escaso en matrices.
- ✓ Polinomios y funciones racionales.
- ✓ Simulación: sistemas explícitos e implícitos de solución de ecuaciones diferenciales.
- ✓ Control clásico y robusto.

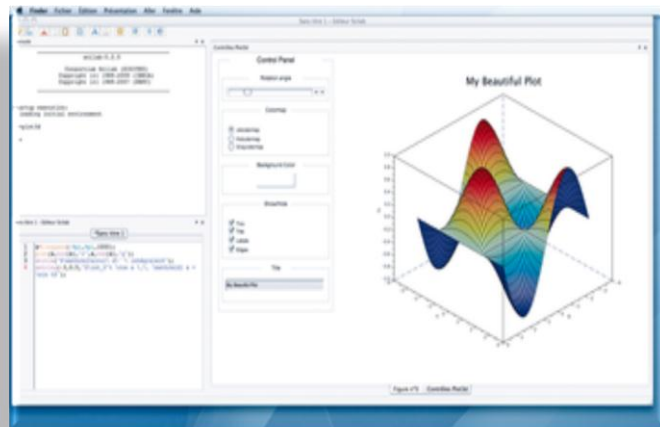


Fig. 3.4.2.1. Visualización de una gráfica.





- ✓ Optimización diferenciable y no-diferenciable.

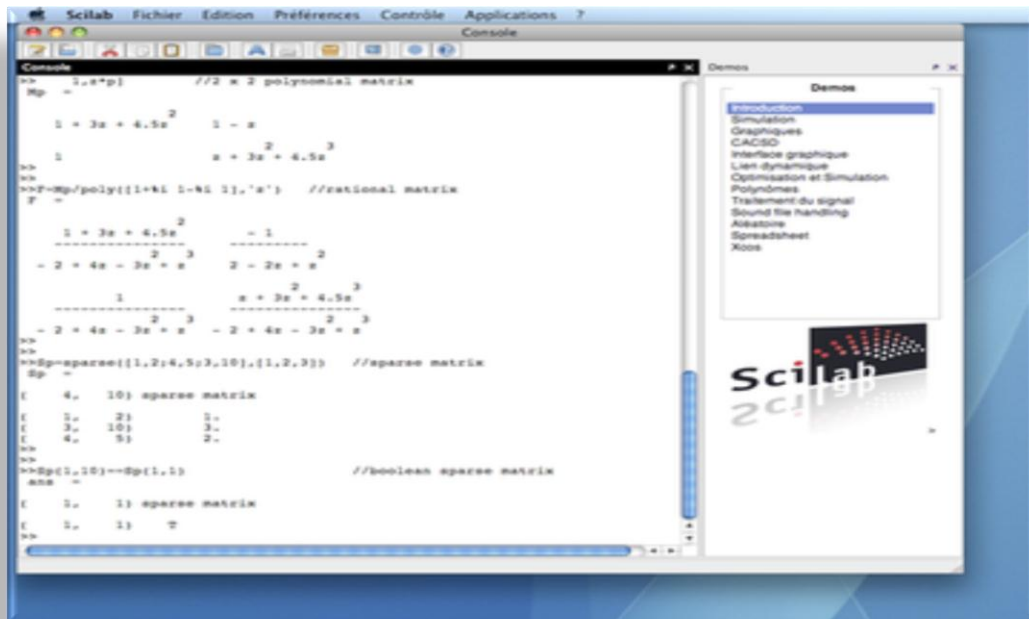


Fig. 3.4.2.2. Computación numérica.

Análisis de datos (Fig.3.4.2.3):

- ✓ Aproximación e interpolación.
- ✓ Procesamiento de Señales.
- ✓ Estadística.

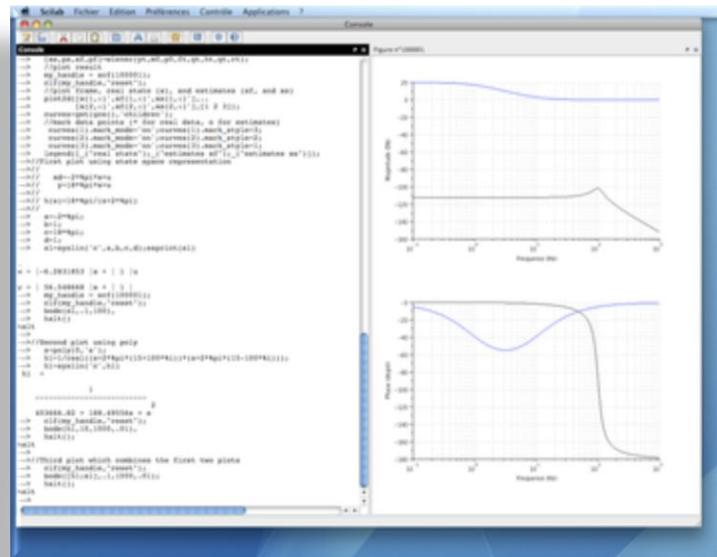


Fig. 3.4.2.3. Análisis de datos.





Instalación.

El paquete de instalación de Scilab está en los repositorios de Ubuntu, lo único que se debe hacer es instalarlo por el medio que más se facilite (Ver 2.11. *Instalar aplicaciones.*).

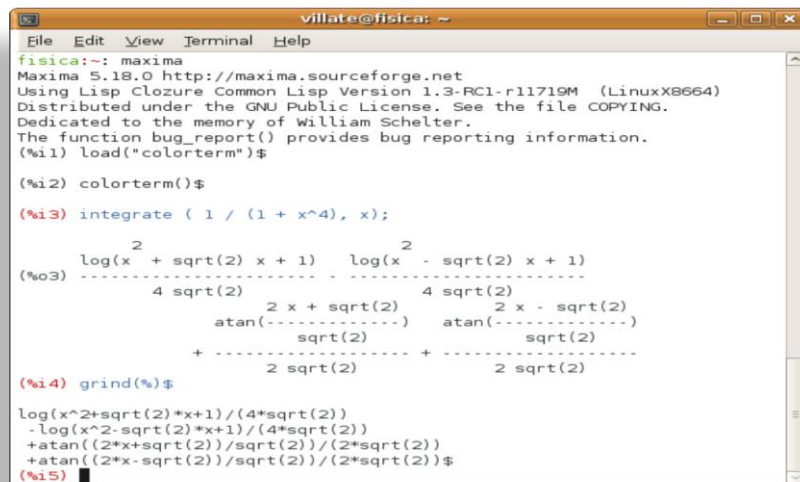
En consola, `'sudo aptitude install scilab'`; esto instala Scilab y todas sus dependencias.

3.4.3. Maxima.

El sistema de álgebra computacional Maxima es un motor de cálculo simbólico escrito en lenguaje Lisp, publicado bajo licencia GNU GPL.

Cuenta con un amplio conjunto de funciones para hacer manipulación simbólica de polinomios, matrices, funciones racionales, integración, derivación, manejo de gráficos en 2D y 3D, manejo de números de coma flotante muy grandes, expansión en series de potencias y de Fourier, entre otras funcionalidades.

Además tiene un depurador a nivel de fuente para el código de Maxima (Fig. 3.4.3.1. *Maxima corriendo en Linux en modo de línea de comandos.*)



```
villate@fisica: ~
File Edit View Terminal Help
fisica:~: maxima
Maxima 5.18.0 http://maxima.sourceforge.net
Using Lisp Clozure Common Lisp Version 1.3-RC1-r11719M (LinuxX8664)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) load("colorterm")$
(%i2) colorterm()$
(%i3) integrate ( 1 / (1 + x^4), x);
(%o3) -----
      2      2      2      2
      log(x + sqrt(2) x + 1)  log(x - sqrt(2) x + 1)
      -----
      4 sqrt(2)              4 sqrt(2)
      atan(-----) atan(-----)
      sqrt(2)              sqrt(2)
      + ----- + -----
      2 sqrt(2)              2 sqrt(2)
(%i4) grind(%)$
Log(x^2+sqrt(2)*x+1)/(4*sqrt(2))
-log(x^2-sqrt(2)*x+1)/(4*sqrt(2))
+atan((2*x+sqrt(2))/sqrt(2))/(2*sqrt(2))
+atan((2*x-sqrt(2))/sqrt(2))/(2*sqrt(2))$
(%i5) █
```

Fig. 3.4.3.1. Maxima corriendo en modo de línea de comandos.





Maxima está basado en el sistema original de Macsyma desarrollado por el MIT (*Instituto Tecnológico de Massachussets*) en los años 70.

Maxima funciona en modo consola, sin embargo incluye las interfaces gráficas xMaxima y wxMaxima para facilitar su uso.

Como está escrito en Common Lisp es fácilmente accesible para la programación, desde la capa inferior de Lisp puede llamarse a Maxima.

Como la mayoría de sistemas algebraicos, Maxima se especializa en operaciones simbólicas. Además, ofrece capacidades numéricas especiales como son los números enteros y racionales que pueden crecer en tamaño sólo limitado por la memoria de la máquina, y números reales en coma flotante cuya precisión puede ser arbitrariamente grande.

Para cálculos intensivos en reales de coma flotante, Maxima ofrece la posibilidad de generar código en otros lenguajes de programación, como Fortran, que quizá se ejecuten de manera más eficiente.

Maxima es un sistema de propósito general; como tal los cálculos especiales como la factorización de números grandes, la manipulación de polinomios extremadamente grandes, etc; son normalmente realizados de forma más eficiente y rápida en sistemas especializados. Maxima es un sistema en donde se permite el manejo de expresiones simbólicas y numéricas, además produce resultados con alta precisión.

Maxima es un programa para la manipulación de expresiones simbólicas y numéricas, incluyendo diferenciación, integración, serie de Taylor, transformadas de Laplace, ecuaciones diferenciales ordinarias, sistemas de ecuaciones lineales, polinomios y conjuntos, listas, vectores, matrices y tensores.





Maxima produce resultados numéricos de alta precisión utilizando fracciones exactas, enteros de precisión arbitraria, y de precisión variable números de punto flotante. Maxima puede dibujar funciones y datos en dos y tres dimensiones (Fig. 3.4.3.2.).

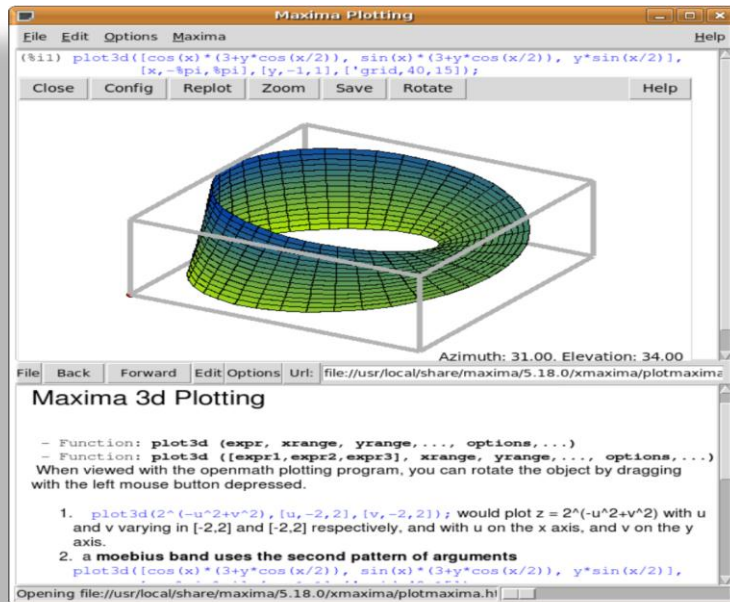


Fig. 3.4.3.2. Gráfica de una función en Maxima.

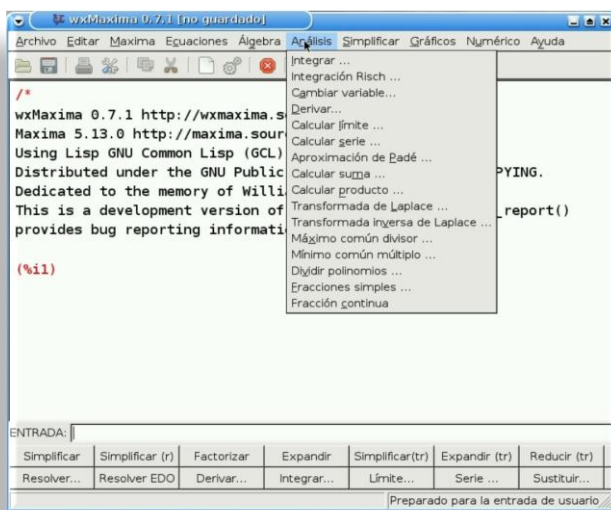


Fig. 3.4.3.3. Pantalla principal de Maxima.

Esta es la pantalla principal de wxmaxima, la cual es muy intuitiva, tiene menús desplegables que facilitan el trabajo, no siendo necesario aprender tantos comandos. Dependiendo de lo que se quiera hacer, se explora los menús para encontrar la opción adecuada (Fig. 3.4.3.3. Pantalla principal de Maxima.).





Si se quiere, por ejemplo, resolver una integral, realizamos los siguientes pasos:

1. Ejecutar wxmaxima desde el menú de aplicaciones o bien tecleando en la consola 'wxmaxima'.
2. Ir al menú desplegable 'Análisis' > 'Integrar...'
3. Sale una ventana donde se debe ingresar la información (Fig. 3.4.3.4. Captura de datos.).
4. En la opción Integrar se introduce la función que se quiere integrar, en este caso será la función x^3 .
5. En la opción para la variable, se especifica cuál es la variable que se va a integrar (x).
6. En el caso que sea una integral definida se selecciona la opción 'Definite integration', y se especifica los límites de la función.

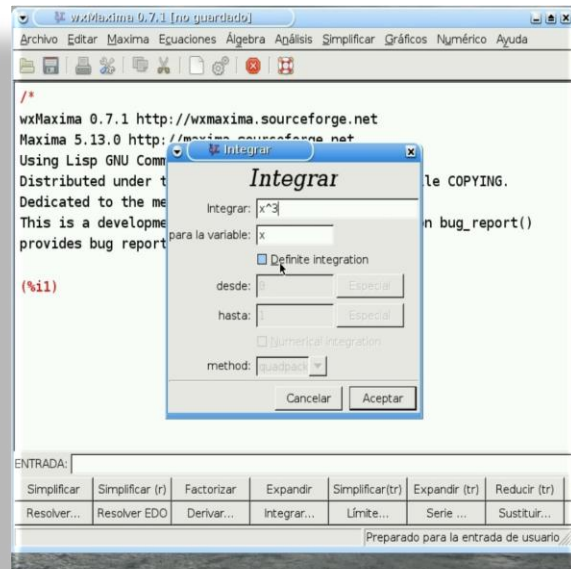


Fig. 3.4.3.4. Captura de datos.

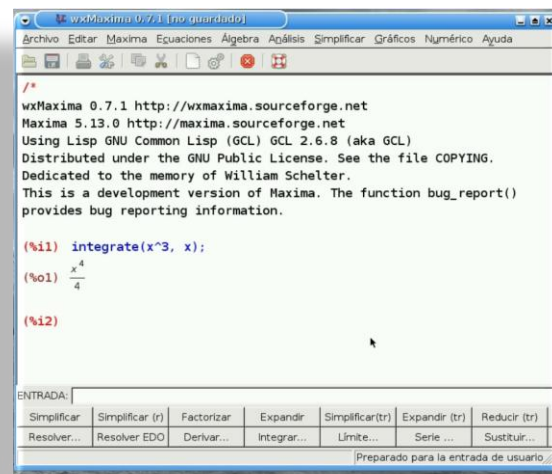


Fig. 3.4.3.5. Resultado de integrar la función x^3





7. Las demás opciones se dejan como están, al hacer clic en aceptar mostrará el resultado en la ventana principal. El resultado de hacer la integración indefinida de la función ' x^3 ' es ' $\frac{1}{4} x^4$ ' (Fig. 3.4.3.5. Resultado de integrar la función x^3).

Instalación.

El paquete de instalación de Maxima está en los repositorios de Ubuntu, lo único que se debe hacer es instalarlo por el medio que más se facilite (Ver 2.11. Instalar aplicaciones.).

Al instalar sólo el paquete '*maxima*' se tiene el programa en modo de línea de comandos; si se requiere una interfaz gráfica de usuario, se debe instalar también el paquete '*wxmaxima*'.

En consola, '*sudo aptitude install máxima wxmaxima*'; esto instalará el programa Maxima con una interfaz de usuario y todas sus dependencias.

3.4.4. Otras aplicaciones para matemáticas.

3.4.4.1. Euler.

Euler sirve para hacer cálculos, usar vectores y matrices, dibujar funciones, programar algoritmos y mucho más. Euler es un programa de computación rápido e interactivo para trabajar con números reales y complejos y matrices, en el estilo de Matlab y Octave. Puede dibujar y animar sus funciones en dos y tres dimensiones. '<http://euler.sourceforge.net>'.





3.4.4.2. Genius.

Genius es una aplicación avanzada para hacer cálculos y algoritmos complejos (*similar a Matlab*). Permite resolver sistemas de ecuaciones, realizar cuentas, cálculos y programar algoritmos a medida. Genius es útil tanto como una simple calculadora y como una herramienta educativa o de investigación. La sintaxis es muy intuitiva y está diseñado para imitar como las matemáticas se suelen escribir. Más información en '<http://linexedu.educarex.es/index.php/Genius>.'

3.4.4.3. Termograf.

Termograf, no es propiamente una aplicación para matemáticas, pero puede ser útil para el estudiante de ingeniería. Usado en la rama de la termodinámica, permite dibujar ciclos termodinámicos de trabajo y frigoríficos, calcular el calor generado, la energía obtenida, entropía; etc. Además tiene una amplia base de datos con todos los líquidos y sustancias de trabajo que se emplean habitualmente. Descárgalo en '<http://termograf.unizar.es>.'

3.5. Utilidades para la productividad diaria.

3.5.1. OpenOffice.org.

OpenOffice.org (*frecuentemente escrito OOO para abreviar*) es una suite ofimática libre (*código abierto y distribución gratuita*) que incluye herramientas como procesador de textos, hoja de cálculo, presentaciones, herramientas para el dibujo vectorial y base de datos. Está disponible para varias plataformas, tales como Microsoft Windows, GNU/Linux, BSD, Solaris y Mac OS X.





Soporta numerosos formatos de archivo, incluyendo como predeterminado el formato estándar ISO/IEC OpenDocument (*ODF*), entre otros formatos comunes, así como también soporta más de 110 idiomas. Viene instalada por default en Ubuntu.

OpenOffice.org tiene como base inicial a StarOffice, una suite ofimática, de pago, desarrollada por StarDivision y adquirida por Sun Microsystems en agosto de 1999.

Hoy en día, el desarrollo de OpenOffice.org está liderado por Oracle Corporation (*inicialmente por Sun Microsystems*), con el apoyo de otras compañías como Novell, RedHat, RedFlag, IBM, Google; entre otras.

Principales herramientas de la suite OpenOffice.org:

- ✓ **OpenOffice.org Writer** es un procesador de texto. El formato nativo para guardar documentos es *.odt*. Además de otros formatos estándares y ampliamente utilizados de documentos, puede abrir y grabar el formato propietario *.doc* y *.docx* de Microsoft Word.

El formato nativo para exportar documentos es XML. También puede exportar a archivos PDF nativamente sin usar programas intermedios.

Puede proteger documentos con contraseña, guardar versiones del mismo documento, insertar imágenes, objetos OLE, admite firmas digitales, símbolos, fórmulas, tablas de cálculo, gráficos, hiperenlaces, marcadores, formularios; etc.





Writer es también un potente editor HTML tan fácil de usar como un documento de texto. Sólo con entrar en el menú Ver y seleccionar '*Diseño para internet*' cambia el formato del cuadro de texto, asemejándose a una página web, que se puede editar de la misma forma que si fuera un procesador de textos.

Con él también se pueden hacer etiquetas, así como tarjetas de presentación fácilmente, sin tener que modificar el formato de un documento de texto para ello. También tiene una galería de imágenes, texturas y botones. Y un gran servicio de ayuda.

- ✓ **OpenOffice.org Calc** es una hoja de cálculo avanzada similar a Microsoft Excel, con un rango de características más o menos equivalente.

Su tamaño es mucho menor y proporciona un número de características no presentes en Excel, incluyendo un sistema que automáticamente define series para representar gráficamente basado en la disposición de los datos del usuario.

Calc también es capaz de exportar hojas de cálculo como archivos PDF, cuenta con filtros, auto filtros y puede hacer agrupaciones en tablas dinámicas.

Dado que, desde el punto de vista de la programación de macros, Calc no es compatible con el modelo de objetos de Excel (*aunque sí con prácticamente todo su repertorio de funciones BASIC*), no es tan vulnerable a los virus de macros (*o macrovirus*) como el producto de Microsoft.





Además, dado que su difusión es bastante menor, no hay tanto interés por parte de los creadores de software malicioso (*malware*) de atacarlo específicamente.

Calc puede abrir y guardar las hojas de cálculo en el formato de archivos de Microsoft Excel *‘.xls’* y *‘.xlsx’*. El formato por defecto de Calc es *‘.ods’*, se puede fijar para que sea el de Microsoft Excel.

- ✓ **OpenOffice.org Impress** es un programa de presentación similar a Microsoft PowerPoint, el formato nativo para guardar las presentaciones es *‘.odp’*; puede leer y escribir en los formatos *‘ppt’* y *‘pptx’*. Puede exportar presentaciones como archivos *‘.swf’* de Adobe Flash permitiendo que sean ejecutados en cualquier computadora con Adobe Flash Player instalado. También incluye la capacidad de crear archivos PDF.

Impress sufre de la carencia de diseños de presentación listos para usarse. Sin embargo, se pueden obtener fácilmente en Internet plantillas de terceros.

Los usuarios de Impress también pueden instalar la Open Clip Art Library (*Biblioteca Abierta de Clip Art*), que agrega una enorme galería de banderas, logos, iconos, estandartes y pancartas para presentaciones generales y proyectos de dibujo. Algunas distribuciones como Debian y Ubuntu han proporcionado un paquete llamado *‘openclipart’* listo para usar y fácil de bajar desde sus repositorios, instalando una galería de imágenes y sonidos para el OpenOffice.org.





3.5.2. Firefox.

Mozilla Firefox es un navegador web libre y de código abierto, descendiente de Mozilla Application Suite y actualmente desarrollado por la Corporación Mozilla, la Fundación Mozilla y un gran número de voluntarios externos. Mozilla Firefox es el segundo navegador más utilizado para Internet, con una cuota de mercado del 22,93% a finales de agosto de 2010, según la firma Net Applications.

Para visualizar páginas web, Firefox usa el motor de renderizado Gecko, que implementa estándares web actuales además de otras funciones, algunas de las cuales están destinadas a anticipar probables adiciones a los estándares web.

Incluye navegación por pestañas, corrector ortográfico, búsqueda progresiva, marcadores dinámicos, un administrador de descargas, navegación privada, navegación con georreferenciación y un sistema de búsqueda integrado que utiliza el motor de búsqueda que desee el usuario. Además se pueden añadir funciones a través de complementos desarrollados por terceros, entre los que hay una amplia selección, característica que ha atraído a muchos de los usuarios actuales del navegador.

Firefox es un navegador multiplataforma y está disponible en varias versiones de Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas basados en Unix. Su código fuente es software libre, publicado bajo una triple licencia GPL/LGPL/MPL. Viene instalado por default en Ubuntu.





Conclusiones.

Para que nuestro equipo computacional funcione necesita un sistema operativo, algunos programadores se negaron a que ciertas empresas desarrolladoras, sobre todo Microsoft, desarrollaran sistemas operativos sin entregar al mismo tiempo el código de fabricación que permitiera adaptarse a nuestras propias necesidades. Con este trabajo de tesis ahora conocemos las bases sobre las cuales fue creado todo el movimiento del software libre.

Cuando Linus Torvalds lanzó a la red (*web*) un nuevo sistema operativo para computadoras, rápidamente un '*ejército*' de programadores alrededor del mundo trabajó con él para hacer evolucionar y dar una salida a este sistema que compite de frente con otros sistemas operativos y que recibe el nombre de GNU/Linux.

Todos estos programadores, como '*hormigas obreras*', unidos entre sí por la red, han hecho evolucionar la proposición base de Linus Torvalds, adaptándola, modificándola, enriqueciéndola y creando así uno de los sistemas operativos más seguros y estables.

Ningún otro proyecto de creación software ha hecho trabajar jamás a tanta gente repartida por todo el mundo. Lo más innovador de GNU/Linux no es su código fuente, sino toda la maquinaria social que se ha creado a su alrededor. Se creó una comunidad de individuos en la que todos tienen la sensación de que su contribución es válida y útil.

Para un estudiante de Ingeniería (*en general, de cualquier carrera y cualquier nivel*); para quien es un gasto enorme contar con software privativo, necesario en sus estudios; las alternativas aquí presentadas pueden serle de mucha de utilidad, sin hacer un gasto que le sería excesivo.





Puede ser que las alternativas que se han presentado no sean de gran utilidad a muchos estudiantes que son ajenos a la ingeniería, pero la intención de este trabajo es que sepan que hay alternativas de software a las que habitualmente usan.

Algunas conclusiones muy particulares que puedo compartir son:

- ✓ Conocer cuales son los orígenes del sistema operativo GNU/Linux.
- ✓ Utilizar GNU/Linux, y en particular la distribución Ubuntu 10.04, para actividades cotidianas en equipo computacional (*aplicaciones ofimáticas, música, videos, navegar en Internet, etc.*) y para las actividades de ingeniería el CAD, el EDA, y otros programas de aplicación.
- ✓ Instalar aplicaciones, administrar archivos, el uso básico de la consola y sus comandos más frecuentes.
- ✓ Lo más importante ha sido el empleo de este sistema operativo, que es muy flexible y adaptable a las necesidades de cualquier estudiante de Ingeniería, principalmente en la carrera de Ingeniería Mecánica Eléctrica.





Apéndice A. Instalación de Ubuntu.

Esta guía proporciona instrucciones para instalar Ubuntu CD Desktop en su equipo.

El modo de instalación por defecto ha cambiado desde las anteriores versiones de Ubuntu; ahora se utiliza un entorno live CD que proporciona un instalador gráfico.

Como en cualquier instalación de software (*especialmente en la de un sistema operativo completo*) existe la posibilidad de que algo vaya mal, por eso **es importante hacer una copia de seguridad** de todos los datos importantes antes de continuar.

Conseguir un disco de instalación

Hay varias maneras de conseguir un disco de instalación de Ubuntu. La forma más recomendable es hacerlo desde la página oficial de Ubuntu (*'<http://www.ubuntu.com/desktop/get-ubuntu>'*), en ella encontrará diferentes opciones:

- ✓ Instalador para sistemas Windows. Se descarga un pequeño ejecutable (*.exe) de 1.5 MB y mediante él se realiza toda la descarga e instalación de Ubuntu.
- ✓ Descargar la imagen 'ISO', para 'quemarla' en un CD. Se puede descargar vía ftp o vía torrent.
- ✓ Pedir el CD de instalación a Canonical, totalmente gratis; lo cual puede tardar varias semanas en llegar.
- ✓ Comprar un CD de instalación con alguno de los distribuidores.





Iniciar desde CD.

Se coloca el CD dentro de la unidad óptica y se reinicia la computadora. Verá un menú de inicio (Fig. A.1.), antes de empezar, presionar F2 y escoger el idioma con el que se quiere iniciar Ubuntu. Tras escoger idioma, presiona 'enter' para seleccionar la opción de 'Probar Ubuntu sin alterar su equipo'.

Opciones de accesibilidad. Para activar las opciones de accesibilidad, presiona F5. Esto hará que aparezca la lista de opciones de accesibilidad.

- ✓ Ninguna (seleccionado por defecto)
- ✓ Alto contraste
- ✓ Lupa
- ✓ Lector de pantalla
- ✓ Modificadores del teclado
- ✓ Teclado en pantalla



Fig. A.1. Pantalla de Inicio.

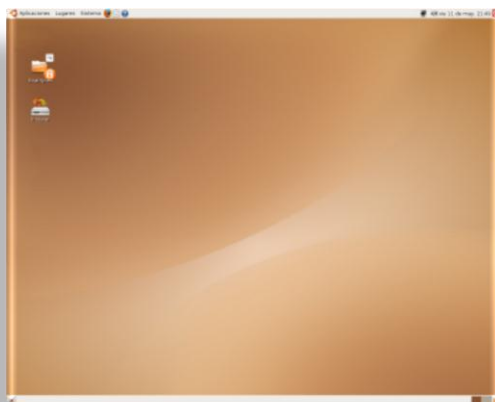


Fig. A.2. Ubuntu Live CD.

permanentemente en el disco duro.

Una vez Ubuntu ha acabado de cargar (puede tomar tiempo) se encontrará dentro de un sistema Ubuntu ejecutado desde el CD (Fig. A.2.). En este punto, puede probar el sistema y decidir si le gusta antes de instalarlo

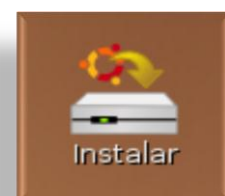


Fig. A.3. Instalador





Probar conexión a internet. Si no tiene una conexión cuando instale el sistema (este requiere descargar archivos, para que Ubuntu esté completamente en español) puede que salga un error cuando la barra de progreso este al 82%. Lo único que ocasionaría este error es que algunos paquetes no estén en español.

Instalar. Cuando haya probado el sistema y esté listo para instalar, se da doble clic en el icono de instalación del escritorio. Esto ejecutará el instalador gráfico de Ubuntu (Fig. A.3).



Fig. A.4. Opciones de Idioma.

Idioma. La primera pantalla es la de selección de idioma (Fig. A.4), en la cual puede escoger el idioma por defecto en la instalación. Cuando se haya seleccionado, hacer clic en el botón ‘Adelante’.



Fig. A.5. Selección de zona horaria.

Zona horaria. Se hace clic en el mapa para acercarse y seleccionar más fácilmente la localización (Fig. A.5). También puede hacer clic en la caja desplegable para escoger entre una lista de opciones disponibles. Cuando la hora sea correcta, dar clic en ‘Adelante’.

Distribución de Teclado. Selecciona la distribución de teclado (Fig. A.6), es buena idea escribir algo en la caja de texto de abajo para comprobar que funciona correctamente si no está seguro de la distribución. Cuando sea el correcto dar clic en ‘Adelante’.



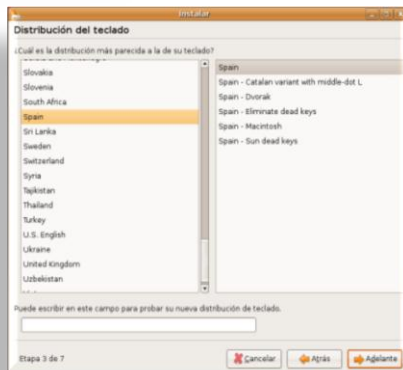


Fig. A.6. Distribución del teclado.

¿Quién eres? Introducir su nombre, nombre de usuario, contraseña y el nombre del equipo (Fig. A.7). (nota: no utilizar espacios, mayúsculas o caracteres extraños en el nombre de usuario.)



Fig. A.7. Creación de usuario

Selección de un disco. Para usar la partición guiada seleccionar uno de ellos y dar clic a 'adelante', la siguiente pantalla preguntará qué quiere hacer con el disco, para hacer espacio a Ubuntu. La acción por defecto es redimensionar una partición existente. La barra deslizante permite seleccionar un nuevo tamaño para la partición existente (Fig. A.8. Partición de disco.).



Fig. A.8. Particionar Disco.

La segunda opción es para utilizar automáticamente todo el disco y la tercera opción es para editar manualmente la tabla de particiones (con una herramienta gráfica).

El espacio libre al que se refiere es un área de disco sin ninguna partición en él. No es lo mismo un espacio que no contiene ningún archivo que un espacio que no contiene ninguna partición (es decir, está sin formatear).





Confirmación. Este es el momento de comprobar que todo está bien antes de que el instalador empiece a hacer cambios en el sistema. Revisar todas las opciones y si hay algún error deberá dar clic al botón 'Atrás' hasta llegar a donde esté el error. (Fig. A.9. *Confirmación para la instalación.*).



Fig. A.9. *Confirmación para la instalación*

Instalación. El instalador hará los cambios en las particiones e instalará Ubuntu. Esto puede durar algún tiempo, especialmente el redimensionado de particiones ya que todos los datos han de ser movidos a la partición que va a mantenerse.



Fig. A.10. *Instalación completa.*

Finalización. Cuando el instalador acabe, tendrá la opción de continuar en el Live CD o reiniciar la computadora (Fig. A.10).

Cuando reinicie la computadora, cargará Ubuntu, si es el único sistema operativo instalado. Si se tiene más de un sistema operativo, le permitirá elegir con cuál de ellos se quiere iniciar sesión.

Ahora ya puede ingresar con el usuario y contraseña que se eligió durante la instalación. Tiene documentación sobre el uso de Ubuntu instalada en el equipo, accesible desde Sistema -> Ayuda -> Documentación del sistema.



Fig. A.11. *Documentación de ayuda.*





Apéndice B. Comandos más utilizados en GNU/Linux.

A continuación se presenta un extracto de los comandos que se utilizan con más frecuencia en los sistemas GNU / Linux.

addgroup

Se utiliza para crear un grupo nuevo.

Sintaxis: *addgroup nom_grupo*

adduser

Se utiliza para añadir un usuario. No sólo se creará la cuenta del usuario, sino también su directorio de trabajo y añadirá una serie de archivos de configuración al directorio de trabajo del nuevo usuario.

Sintaxis: *adduser nom_usuario [nom_grupo]*

alias

Se utilizan comandos difíciles de recordar o que son demasiado extensos, pero en UNIX existe la posibilidad de dar un nombre alternativo a un comando, con el fin de que cada vez que se quiera ejecutar, sólo se use el nombre alternativo.

Sintaxis: *alias nom_alias = 'comando'*

aptitude search (texto)

Muestra una lista de los paquetes y una descripción relacionado con el texto.

aptitude update

Actualiza la lista de paquetes disponibles para instalar.

aptitude dist-upgrade

Función adicional a la anterior que modifica las dependencias por la de las nuevas versiones de los paquetes, es decir actualiza todo el sistema.

aptitude install (nombre_paquete)

Instala paquetes y sus dependencias.

aptitude purge (nombre_paquete)

Borra paquetes, así como sus dependencias y archivos de configuración.



**bg**

Manda un proceso a segundo plano.

Sintaxis: *bg PID*

cal

Muestra el calendario.

Sintaxis: *cal [[mes] año]*

cat

Permite concatenar uno o más archivos de texto. Muestra el contenido del archivo en pantalla en forma continua, el prompt retornará una vez mostrado el contenido de todo el archivo.

Sintaxis: *cat archivo1 archivo2 > archivo_final*

cd

Cambia de directorio.

Sintaxis: *cd nom_directorio*

chgrp

Cambia el grupo al que pertenece el archivo.

Sintaxis: *chgrp nom_grupo nom_archivo*

chmod

Utilizado para cambiar la protección o permisos de accesos a los archivos.

r:lectura w:escritura x:ejecución

+: añade permisos -:quita permisos

u:usuario g:grupo del usuario o:otros

Sintaxis: *chmod permisos nom_archivo*

chown

Cambia el propietario de un archivo.

Sintaxis: *chown nom_propietario nom_archivo*

clear

Limpia la pantalla y coloca el prompt al principio de la misma.

Sintaxis: *clear*

cp

Copia archivos en el directorio indicado.

Sintaxis: *cp nom_archivo nom_directorio*



**crontab**

Realizar una tarea programada de forma regular.

Sintaxis: *minuto(0-59) hora(0-23) dia_mes(1-31) mes(1-12) dia_semana(0-6) comando*

date

Retorna el día, fecha, hora (con minutos y segundos) y año.

Sintaxis: *date*

delgroup

Se utiliza para eliminar un grupo.

Sintaxis: *delgroup nom_grupo*

deluser

Elimina una cuenta de usuario.

Sintaxis: *deluser nom_usuario*

df

Muestra los sistemas de archivos montados.

Sintaxis: *df*

dmesg

Muestra los mensajes del kernel durante el inicio del sistema.

Sintaxis: *dmesg*

dpkg -reconfigure (paquete)

Volver a reconfigurar un paquete ya instalado

du

Sirve para ver lo que ocupa cada directorio dentro del directorio actual.

Sintaxis: *du*

echo

Muestra un mensaje en pantalla.

Sintaxis: *echo 'Cadena'*

eject

Expulsar la unidad de CD, siempre y cuando no esté en uso.

Sintaxis: *eject*



**exit**

Cierra las ventanas o las conexiones remotas establecidas.

Sintaxis: *exit*

fg

Manda un proceso a primer plano.

Sintaxis: *fg PID*

file

Determina el tipo del o los archivo(s) indicado(s).

Sintaxis: *file nom_archivo*

free

Muestra datos sobre el estado de la memoria del sistema, tanto swap como memoria física.

Sintaxis: *free*

gzip

Comprime un archivo utilizando la extensión '*.gz'.

Sintaxis: *gzip nom_archivo*

id

Numero id de un usuario.

Sintaxis: *id*

ifconfig

Obtener información de la configuración de red.

Sintaxis: *ifconfig*

insmod

Carga en memoria un módulo.

Sintaxis: *insmod*

job

Lista los procesos que se están ejecutando en segundo plano.

Sintaxis: *job*

kill

Termina un proceso y Kill -9 (*pid*) fuerza a terminar un proceso.

Sintaxis: *kill [opciones] PID*



**ln**

Sirve para crear enlaces a archivos, es decir, crear un archivo que apunta a otro. Puede ser simbólico si usamos -s o enlace duro.

Sintaxis: *ln -s nom_archivo*

logout

Las sesiones terminan con el comando logout.

Sintaxis: *logout*

ls

Lista los archivos y directorios dentro del directorio de trabajo.

Sintaxis: *ls*

lsmod

Muestra los módulos cargados en memoria.

Sintaxis: *lsmod*

make

Es una herramienta que controla la creación de ejecutables y otros archivos de un programa a partir de los archivos fuente.

Sintaxis: *make*

man

Ofrece información acerca de los comandos o tópicos del sistema UNIX, así como de los programas y librerías existentes.

Sintaxis: *man comando-paquete-librería*

mkdir

Crea un nuevo directorio.

Sintaxis: *mkdir nom_directorio*

mv

Este comando sirve para cambiar la ubicación de un archivo o también para renombrarlo.

Sintaxis: *mv lugar1 lugar2 / nom_archivo1 nom_archivo2*

mount

Comando para 'montar' un dispositivo, esto es, asignarle un lugar dentro del árbol de directorios.

Sintaxis: *mount -t sistema_de_archivos dispositivo nom_directorio*

passwd

Se utiliza para establecer la contraseña a un usuario.

Sintaxis: *passwd nom_usuario*



**poweroff**

Apagar la computadora.

Sintaxis: *poweroff*

ps

Muestra información acerca de los procesos activos.

Sintaxis: *ps*

pwd

Muestra el directorio actual de trabajo.

Sintaxis: *pwd*

rm

Elimina un archivo.

Sintaxis: *rm nom_archivo*

rmmod

Quita de la memoria un módulo, pero sólo si no está siendo usado.

Sintaxis: *rmmod*

startx

Inicia el entorno gráfico (servidor X).

Sintaxis: *startx*

su (o sudo en ubuntu)

Con este comando accedemos al sistema como root.

Sintaxis: *su*

top

Muestra los procesos que se ejecutan al momento, así como los recursos que 'consumen' (Memoria,CPU...).

Sintaxis: *top*

umount

Sirve para desmontar un dispositivo, previamente montado.

Sintaxis: *umount dispositivo*

unalias

Borra un alias.

Sintaxis: *unalias nom_alias*





vi

Permite editar un archivo. Es uno de los más usado en los sistemas UNIX.
Sintaxis: *vi nom_archivo*

view

Es similar al vi, sólo que no permite guardar modificaciones en el archivo.
Sintaxis: *view nom_archivo*

wc

Cuenta los caracteres, palabras y líneas del archivo de texto.
Sintaxis: *wc nom_archivo*

whereis

Devuelve la ubicación del archivo especificado, si existe.
Sintaxis: *whereis nomb_archivo*

who, w

Lista quienes están conectados al servidor.
Sintaxis: *who / w*

whoami

Escribe su nombre de usuario en pantalla.
Sintaxis: *whoami*





Bibliografía.

- ✓ **The Official Ubuntu Book.**
 - Autores: *Benjamin Mako Hill, Jono Bacon, Corey Burger.*
 - Editorial: *Prentice Hall.*

- ✓ **Moving to Ubuntu Linux.**
 - Autor: *Marcel Gagné.*
 - Editorial: *Addison Wesley Professional.*

- ✓ **Linux diferencia visual.**
 - Autor: *José Andrés Martínez.*
 - Editorial: *McGraw Hill.*

- ✓ **Administración avanzada del sistema operativo GNU/Linux.**
 - Autores: *Remo Suppi Boldrito, Josep Jorba Esteve.*
 - Editorial: *FUOC.*

- ✓ **Beginning Ubuntu Linux.**
 - Autor: *Keir Thomas.*
 - Editorial: *Apress.*

- ✓ **De Windows a Linux.**
 - Autor: *Michel Martin.*
 - Editorial: *Alfa Omega – Marcombo.*

- ✓ **Sistemas operativos: Conceptos y Diseño.**
 - Autor: *Milan Milenkovic.*
 - Editorial: *McGraw Hill.*





Mesografía.

- ✓ Página oficial DistroWatch: <http://distrowatch.com/>
- ✓ Documentación Linux: <http://www.linux.org/>
- ✓ Página oficial del proyecto GNU: <http://www.gnu.org/>
- ✓ Página oficial de la fundación del software libre: <http://www.fsf.org/>
- ✓ Página oficial del proyecto GNOME: <http://www.gnome.org/>
- ✓ Página oficial del proyecto KDE: <http://www.kde.org/>
- ✓ Página oficial de Ubuntu: <http://www.ubuntu.com/>
- ✓ Wikipedia: <http://www.wikipedia.org/>
- ✓ Página oficial del proyecto QCAD: <http://www.qcad.org/>
- ✓ Página oficial del proyecto Blender: <http://www.blender.org/>
- ✓ Página oficial del proyecto Medusa4: <http://www.medusa4.com/>
- ✓ Página oficial del proyecto Octave: <http://www.gnu.org/software/octave/>
- ✓ Página oficial del proyecto Scilab: <http://www.scilab.org/>
- ✓ Página oficial del proyecto MAXIMA: <http://maxima.sourceforge.net>
- ✓ Página oficial del proyecto gEDA: <http://www.gpleda.org/index.html>
- ✓ Página oficial del proyecto KICAD: http://www.lis.inpg.fr/realise_au_lis/kicad/
- ✓ Página oficial del proyecto Alliance: <http://www-asim.lip6.fr/recherche/alliance/>
- ✓ Página oficial del proyecto OpenOffice: <http://www.openoffice.org/>
- ✓ Página oficial del proyecto Firefox: <http://www.mozilla.com/firefox/>

