

CONTENIDO

GNU-LINUX.....	2
Historia de Gnu-Linux.	2
Debian.....	6
arch linux	9
Red Hat	10
Suse.....	11
Ubuntu	12
Guadalinex.	14
Instalación de Linux.....	15
Puntos de montaje.	16
Partición swap.....	18
Gestor de Arranque.....	18
Lilo.....	19
GRUB (versión 1).	20
GRUB (versión 2).	21

GNU-LINUX.

HISTORIA DE GNU-LINUX.

En los años setenta, dos investigadores de los Laboratorios Telefónicos Bell (Bell Telephone Labs o BTL) llamados Dennis Ritchie y Ken Thompson desarrollaron un sistema operativo al que llamaron **Unix**.

Al completar el desarrollo de Unix, Ritchie y Thompson expusieron su diseño en una conferencia internacional donde varios de los participantes les pidieron una copia de este sistema. En esa época, BTL había perdido un juicio antitrust (antimonopolio), y el Juez había prohibido a BTL incorporarse a cualquier negocio que no fuera el de las telecomunicaciones. Debido a esto, a BTL le era imposible entrar en el negocio de los sistemas operativos. La presión de los investigadores en obtener una copia de Unix motivó a los ejecutivos de BTL a licenciar su uso como una herramienta de investigación. La licencia de Unix fue usada por las universidades, que consiguieron acceso a un gran sistema operativo y a su código fuente.

Una de las universidades que adquirió una licencia de Unix fue la Universidad de California en Berkeley. Al poco tiempo, la gente de Berkeley había leído el código fuente y había escrito varios programas adicionales para Unix que otros investigadores podrían encontrar útiles.

La Universidad decidió entonces distribuir este código a la comunidad y llamó a sus distribuciones **BSD** (Berkeley Software Distribution). Al principio estas distribuciones BSD consistían en el Unix de BTL y algunas herramientas propias de la Universidad, pero muy pronto comenzaron a cambiar la forma en que el propio sistema operativo funcionaba, comenzaron a modificar el código fuente del propio Unix. Entre otras cosas implementaron el manejo de memoria virtual y programaron el soporte para los protocolos de Arpanet que luego se convertiría en el conocido Internet. Todos estos cambios eran recopilados y lanzados como distribuciones BSD basadas en Unix de BTL.

A mediados de los años ochenta, Richard Stallman, entonces en el Instituto Tecnológico de Massachusetts (MIT) decidió dedicarse a la construcción de lo que denominó **software libre**. El razonamiento de Stallman era que los mayores progresos en la industria del software surgen cuando se coopera entre programadores y según Stallman, las industrias de la época estaban atentando contra la libertad de los usuarios y programadores de compartir el software, así que decidió programar un sistema parecido a Unix y regalarlo. A este sistema le llamó GNU, un acrónimo recursivo que significa Gnu's Not Unix (GNU no es Unix).



Para este entonces, varias compañías estaban ya en el negocio de Unix, que había salido del ámbito puramente universitario. Entre otras empresas, Microsoft distribuía Xenix, una versión de Unix para el procesador 80386 y Sun Microsystems utilizaba BSD como base para su SunOS.

Como resultado de la finalización del juicio antimonopolio, BTL fue desmembrada en varias compañías, una de ellas AT&T, que, de acuerdo con los nuevos arreglos legales ya podía comercializar Unix. Pero cuando AT&T quiso vender Unix comercialmente se dio cuenta que ya existían varias variaciones de su Unix que se estaban comercializando. AT&T inmediatamente lanzó una demanda legal contra todas estas compañías y sobre sus sistemas operativos.

AT&T también decidió demandar a la Universidad de California en Berkeley por distribuir código basado en el Unix de AT&T sin licencia. La Universidad de California a su vez, demandó a AT&T argumentando que BSD incorporaba muchísimas mejoras y que estas mejoras habían sido introducidas por AT&T dentro de Unix.

En el mercado convivían en ese momento el Unix de AT&T junto con muchas distribuciones basadas en BSD como el 386BSD y el FreeBSD. Todas estas distribuciones fueron detenidas por el Juez mientras se celebraban los diversos juicios que se habían abierto y se prohibió su uso.

Para las personas deseosas de correr Unix en las ahora populares PCs, quedaba únicamente una alternativa legal, Minix. Minix era un sistema operativo parecido a Unix desarrollado por el Profesor Andrew Tanenbaum para enseñar a sus alumnos el diseño de un sistema operativo. Sin embargo, debido al enfoque puramente educacional de Minix, Tanenbaum no permitía que este fuera modificado, o usado comercialmente y el sistema se encontraba muy limitado en muchísimos aspectos.

Un estudiante de Finlandia, **Linus Torvalds**, al ver que no le era posible modificar Minix, decidió escribir su propio sistema operativo compatible con Unix. Miles de personas que querían correr Unix en sus PCs vieron aquí su única alternativa debido a que a Minix le faltaban demasiadas cosas y BSD, a pesar de tener toda la funcionalidad esperada, tenía problemas legales.

En realidad, Linus Torvalds no creó un sistema operativo completo, sino su pieza más importante, el núcleo.

El proyecto GNU que Stallman había iniciado hacía ya casi diez años había producido un sistema casi completo a excepción del núcleo, que no conseguía hacer funcionar correctamente. Linus unió su núcleo con GNU formando un sistema operativo completo al que llamó Linux.

Richard Stallman insiste que el sistema operativo resultante debiera ser llamado **GNU/Linux**, ya que incluye más código del proyecto GNU que del proyecto Linux. En la actualidad, el proyecto GNU puede ser instalado sin usar el núcleo de Linux, sino con su propio núcleo conocido como Hurd. Sin embargo, dicho núcleo es excesivamente inestable y casi todas las instalaciones de GNU se realizan con el Kernel de Linux.

El éxito inmediato que tuvo Linux se basó en una variedad de factores. Por un lado, es un núcleo realmente bueno, llegó justo en el momento en que GNU necesitaba un núcleo, y coincidió con el boom de Internet, lo que permitió que se creara una comunidad alrededor de dicho núcleo, tanto para desarrollarlo como para distribuirlo, usarlo y mantenerlo.



Vemos aquí el primer mensaje que Linus envió a internet informando del desarrollo de su núcleo.

Extracto del grupo de noticias comp.os.minix de Usenet.

Mensaje enviado por Linus Torvalds

Fecha: 25 Aug 91 20:57:08 GMT

Hola a todos allá afuera usando Minix -

Estoy haciendo un sistema operativo libre (es solo un hobby, no será grande ni profesional como GNU) pero funciona en 386(486) AT clones, lo he estado cocinando desde abril y está quedando listo. Quisiera alguna retroalimentación de las cosas que a ustedes les gusta y no les gusta de MINIX, ya que mi sistema operativo se parece (la misma disposición física (debido a razones prácticas) entre otras cosas).

Ya he portado bash (1.08) y gcc (1.40) y parece que las cosas funcionan. Esto implica que tendré algo usable en unos meses, y quiero saber qué es lo que a la gente le gustaría que tenga. Cualquier sugerencia es bienvenida, aunque no prometo implementarla :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Sí, está libre de cualquier código de Minix y tiene un FS de multi hilos, NO es portable (usa el switching de tareas de 386) y probablemente nunca soportará algo más que un disco duro de AT, porque es lo único que tengo.

A mediados de los años noventa AT&T vendió Unix a Novell, quién tomó como prioridad número uno resolver las demandas. El acuerdo fue que la Universidad de California eliminaría todo el código de AT&T y lanzaría una última distribución de BSD totalmente libre de problemas de licencias.

Esta distribución fue el 4.4-BSD Lite2. Quien quisiera seguir trabajando sobre BSD debería basar su distribución en 4.4-BSD Lite2 para no tener problemas legales. Hoy en día, existen varias distribuciones del BSD basadas en esa versión. Algunos grupos que distribuyen BSD son:

FreeBSD: el énfasis de este sistema operativo está en la facilidad de uso del sistema. Entre otras metas están la eficiencia del uso de recursos y el rendimiento del sistema. Usando el código de **FreeBSD** se creó el sistema operativo **Darwin** BSD, que a su vez fue usado por **Apple** para crear su **Mac OS**, que a su vez fue la base para crear el sistema **iOS** usado en la actualidad en todos los iPhone, iPod, iPad, etc.

NetBSD: el énfasis de este grupo es la portabilidad del sistema operativo. Actualmente existen sistemas NetBSD para casi cualquier plataforma.

OpenBSD: el énfasis de este grupo es en la seguridad, han hecho una auditoria de todo el código fuente buscando errores y fallas de seguridad. Incorporan sistemas criptográficos libres con una especial preocupación en que no puedan ser espiados ni forzados.

Existen numerosas distribuciones Linux (también conocidas como "distros"), ensambladas por individuos, empresas y otros organismos, conjuntando el núcleo Linux, el sistema GNU y todas las herramientas y utilidades que deseen. Ahí compañías que crean sus propias herramientas y hay compañías que se limitan a usar software libre disponible.

Los sistemas Linux funcionan sobre más de 20 diferentes plataformas de hardware, desde un PC con arquitectura x86 hasta una consola de videojuegos.



FreeBSD



Algunos ejemplos de distribuciones de [Linux](#) son:

DEBIAN

Debian o más concretamente Debian GNU/Linux es una distribución Linux que basa sus principios y fin en el software libre y en luchar contra el software propietario o cerrado.

Creado por Debian Project en el año 1993, dicha organización es la responsable de la creación y mantenimiento de la distribución, centrada en el núcleo Linux y en las utilidades GNU. También mantienen y desarrollan otros sistemas operativos GNU basados en los núcleos Hurd, llamado Debian GNU/Hurd, y NetBSD, llamado Debian GNU/NetBSD.



Debian nace como una apuesta por separar en sus versiones el software libre del software propietario. El modelo de desarrollo es independiente de empresas, creado por los propios usuarios, sin depender de ninguna manera de necesidades comerciales. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia.

Algunas de sus características principales son:

- ▶ Disponibilidad en varias plataformas hardware. Debian 7 está disponible para más de 10 plataformas distintas.
- ▶ Una amplia colección de software disponible. La versión 7 cuenta con más de 37.500 paquetes (programas).
- ▶ Un grupo de herramientas gráficas para facilitar el proceso de instalación y actualización del software en las últimas versiones.
- ▶ Su compromiso con los principios y valores involucrados en el movimiento del Software Libre. Es la distribución que más en serio se toma estos principios, llegado incluso a ser tachada de intransigente por otras distribuciones.
- ▶ No tiene preferencia sobre ningún entorno gráfico en especial ya sea GNOME, KDE, XFCE, LXDE, FluxBox... Cualquier entorno puede funcionar en Debian, dado que dicha distro le da una importancia crucial al hecho de permitir que el usuario tenga libertad para elegir sus propias interfaces.

Debian es una distribución sobre la que se han generado una gran cantidad de distribuciones propias. Algunas de ellas son Augustux, Catux, Gnoppix, Guadalinex, Knoppix, Kanotix, Linex, Linspire, MEPIS, Progeny, SkoleLinux, Ubuntu, UserLinux, Xandros, Mint, etc.

Debian va por su versión 9. Vemos en la siguiente tabla las distintas versiones de Debian.

Versión	Nombre clave	Fecha Lanzamiento	Arquitecturas	Paquetes
1.1	Buzz	17 de junio de 1996	1	474
1.2	Rex	12 de diciembre de 1996	1	848
1.3	Bo	2 de junio de 1997	1	974
2.0	Hamm	24 de julio de 1998	2	~ 1500
2.1	Slink	9 de marzo de 1999	4	~ 2250
2.2	Potato	15 de agosto de 2000	6	~ 3900
3.0	Woody	19 de julio de 2002	11	~ 8500
3.1	Sarge	6 de junio de 2005	11	~ 15400
4.0	Etch	8 de abril de 2007	11	~ 18000
5.0	Lenny	14 de febrero de 2009	12	~ 23000
6.0	Squeeze	6 de febrero de 2011	11	~ 29000
7.0	Wheezy	4 de mayo de 2013	13	~ 36000
8.0	Jessie	5 de noviembre de 2014	11	~ 43000
9.0	Stretch	17 de junio de 2017	10	~ 51000
10.0	Buster	¿?		

RAMAS DE DESARROLLO DE DEBIAN

Cada versión de Debian establece 4 fases distintas:

- Estable.
- En Pruebas.
- Inestable.
- Congelada.

Debian estable (stable), es la versión estabilizada de esta distribución. Cuenta con el apoyo del Equipo de seguridad de Debian y es la recomendada para uso en producción.

Debian en pruebas (testing). En esta versión se encuentran paquetes que han estado previamente en la versión Inestable, pero que contienen muchos menos fallos. Además, deben de poder instalarse en todas las arquitecturas para las cuales fueron construidas. Es la versión más utilizada como sistema de escritorio por aquellos que buscan tener el software más actualizado, aunque se pierde en estabilidad. De aquí saldrá la futura versión Estable.

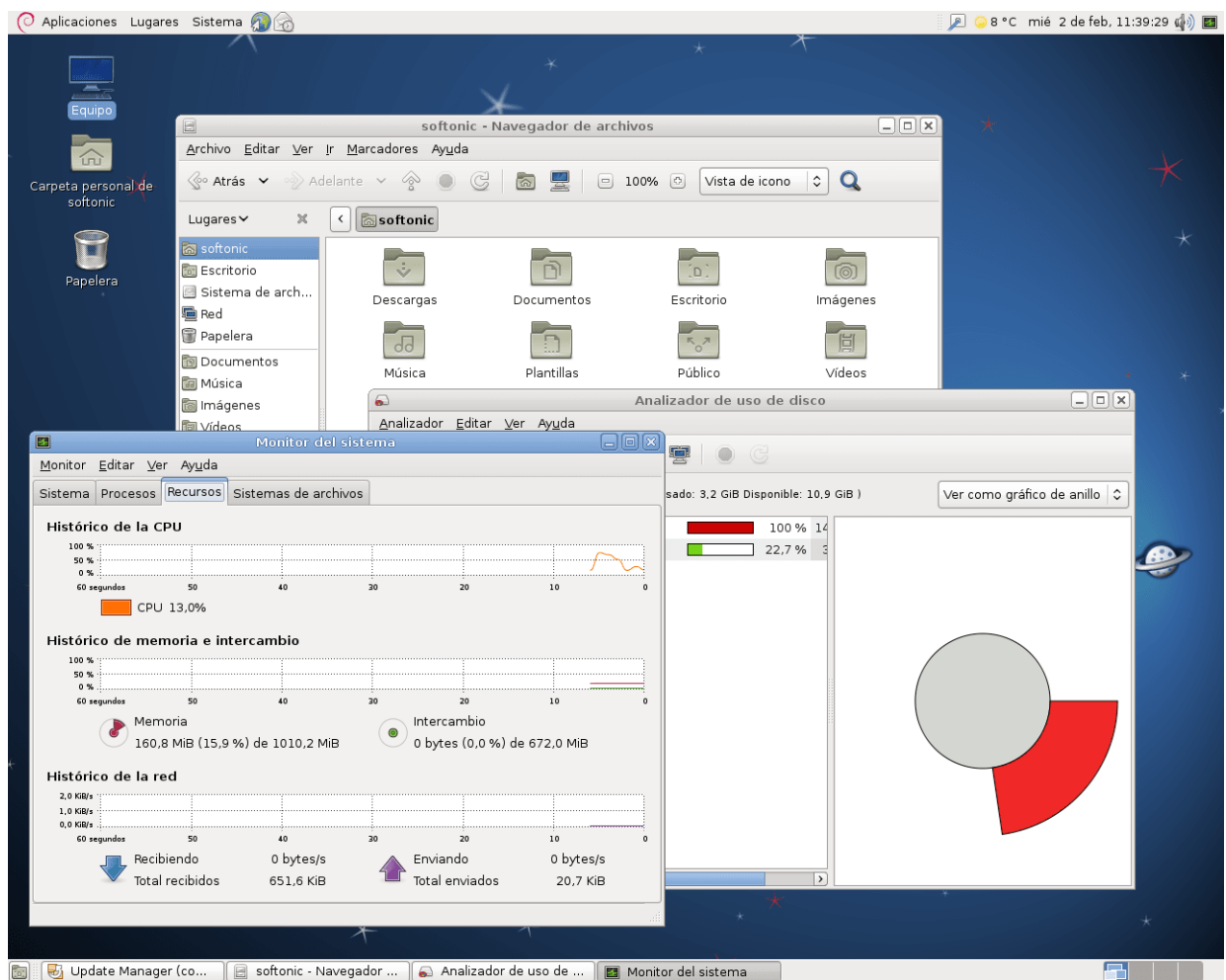
En Debian inestable (unstable), es donde tiene lugar el desarrollo activo de Debian. Es la rama que usan los desarrolladores del proyecto. La rama inestable de Debian siempre tiene como nombre en clave [Sid](#).

Cuando la versión de pruebas (testing) llega a un nivel aceptable de fallos, entonces se "congela", lo que significa que ya no se aceptan nuevos paquetes desde la versión inestable. A continuación, se trabaja para pulir el mayor número de bugs posibles, para así liberar la versión Estable. Ese periodo puede durar varios meses debido a que no se fija una fecha de lanzamiento.

Existe un repositorio de Debian llamado experimental. Esto no es una rama de desarrollo, sino un repositorio que cuenta con los últimos paquetes aparecidos.

Debian no será liberada como estable en tanto sus desarrolladores no consideren que está preparada y es estable. Esa estabilidad se mide basándose en el registro de errores de software o Bug Tracking. Cuando se alcanza un nivel aceptable se le asigna un número de versión, acordado previamente, y se libera como versión estable, solo las versiones estables cuentan con número de versión.

La anterior versión estable es clasificada como old-stable, se mantendrá soporte por un período, generalmente un año, y posteriormente será archivada.



ARCH LINUX

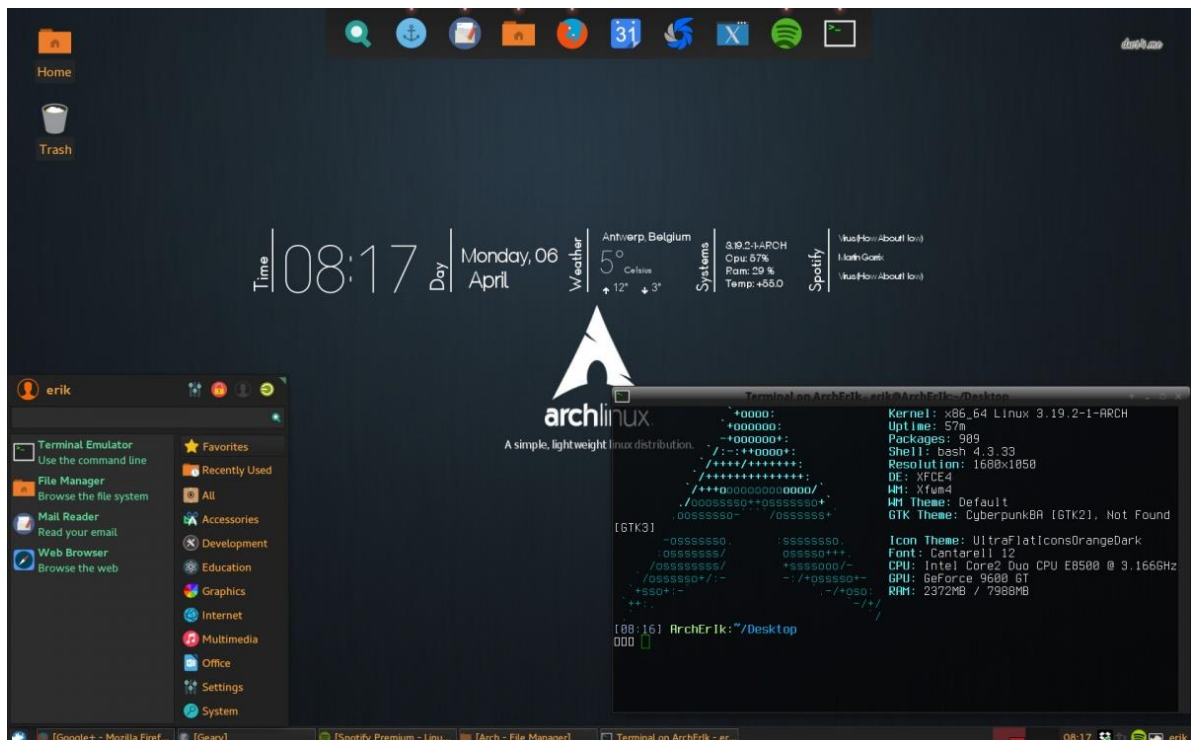
Arch Linux es una distribución Linux para computadoras x86. El enfoque de diseño se centra en la simplicidad, la elegancia, la coherencia de código y el minimalismo. Arch Linux define



simplicidad como «...una ligera estructura base sin agregados innecesarios, modificaciones, o complicaciones, que permite a un usuario individual modelar el sistema de acuerdo a sus propias necesidades». La simplicidad de su estructura no implica sencillez en su manejo. (Más bien al contrario).

Inspirado por CRUX, otra distribución minimalista, Judd Vinet creó Arch Linux en marzo de 2002. Arch Linux utiliza un modelo de rolling release, de tal manera que una actualización regular del sistema operativo es todo lo que se necesita para obtener la última versión del software; las imágenes de instalación son simplemente «capturas» de los principales componentes del sistema. No se puede hablar por tanto de una versión 1.0 o 2.3, Arch Linux se va actualizando continuamente.

Arch Linux no posee herramientas de configuración automática, compartiendo así la misma filosofía que otras distribuciones, como por ejemplo Slackware, por lo que para poder llegar a instalar y configurar el sistema se necesita un grado de conocimiento más que básico.



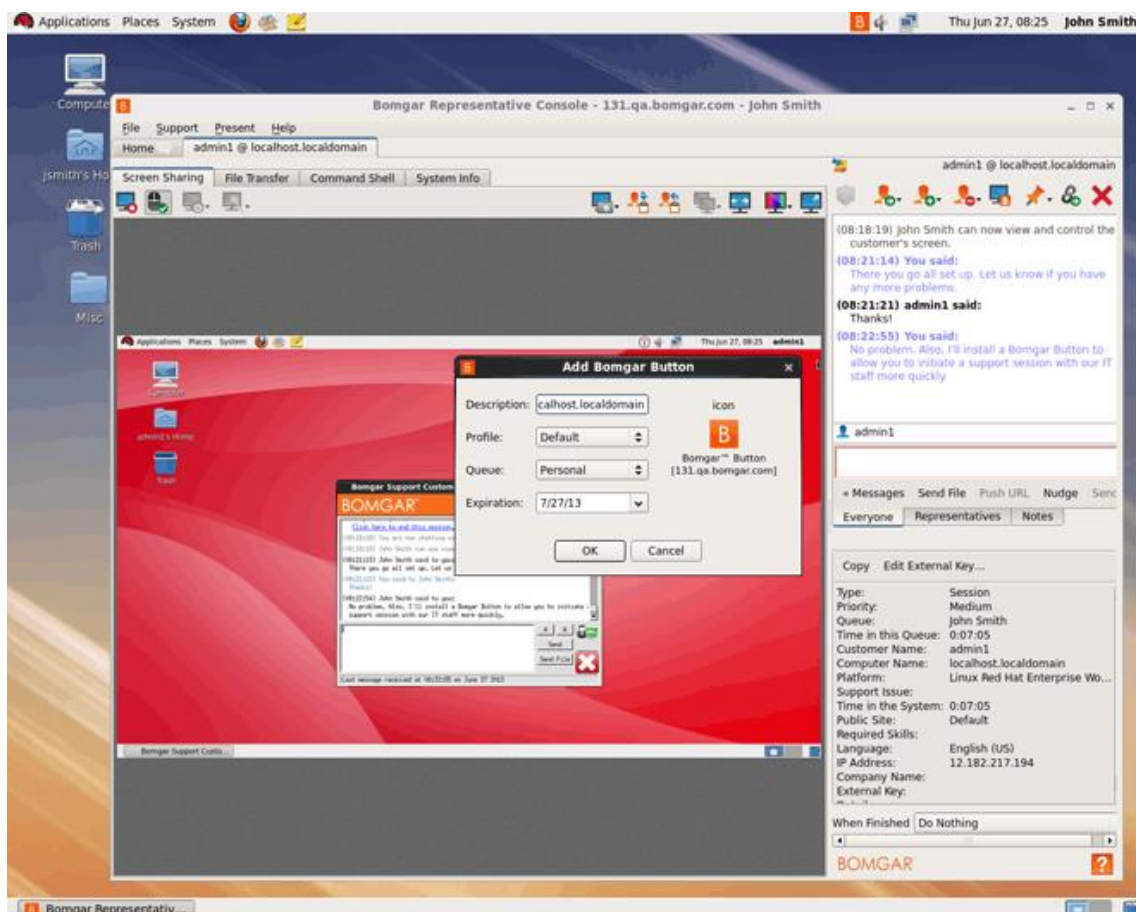
RED HAT

Red Hat Linux es una distribución Linux creada por Red Hat, la cual fue una de las más populares en sus inicios. La versión 1.0 fue presentada el 3 de noviembre de 1994. Fue la primera distribución que usó RPM como su formato de paquete, y en un cierto plazo ha servido como el punto de partida para muchas otras distribuciones, tales como Mandrake, Fedora o Yellow Dog Linux.



Desde el 2003, Red Hat ha desplazado su enfoque hacia el mercado de los negocios con la distribución Red Hat Enterprise Linux (RHEL). Esta versión ha tenido mucho éxito comercial, dado que aúna las ventajas del software libre con el soporte comercial de una gran empresa.

Ahora mismo Red Hat se encarga de los sistemas operativos Red Hat Enterprise Linux, Fedora y CentOS (Estos dos últimos mediante supervisión, apoyo y patrocinio, ya que cuentan con equipos de desarrollo propios). RHEL es su apuesta para empresas, Fedora para usuarios normales y CentOS sirve para ambos cometidos.

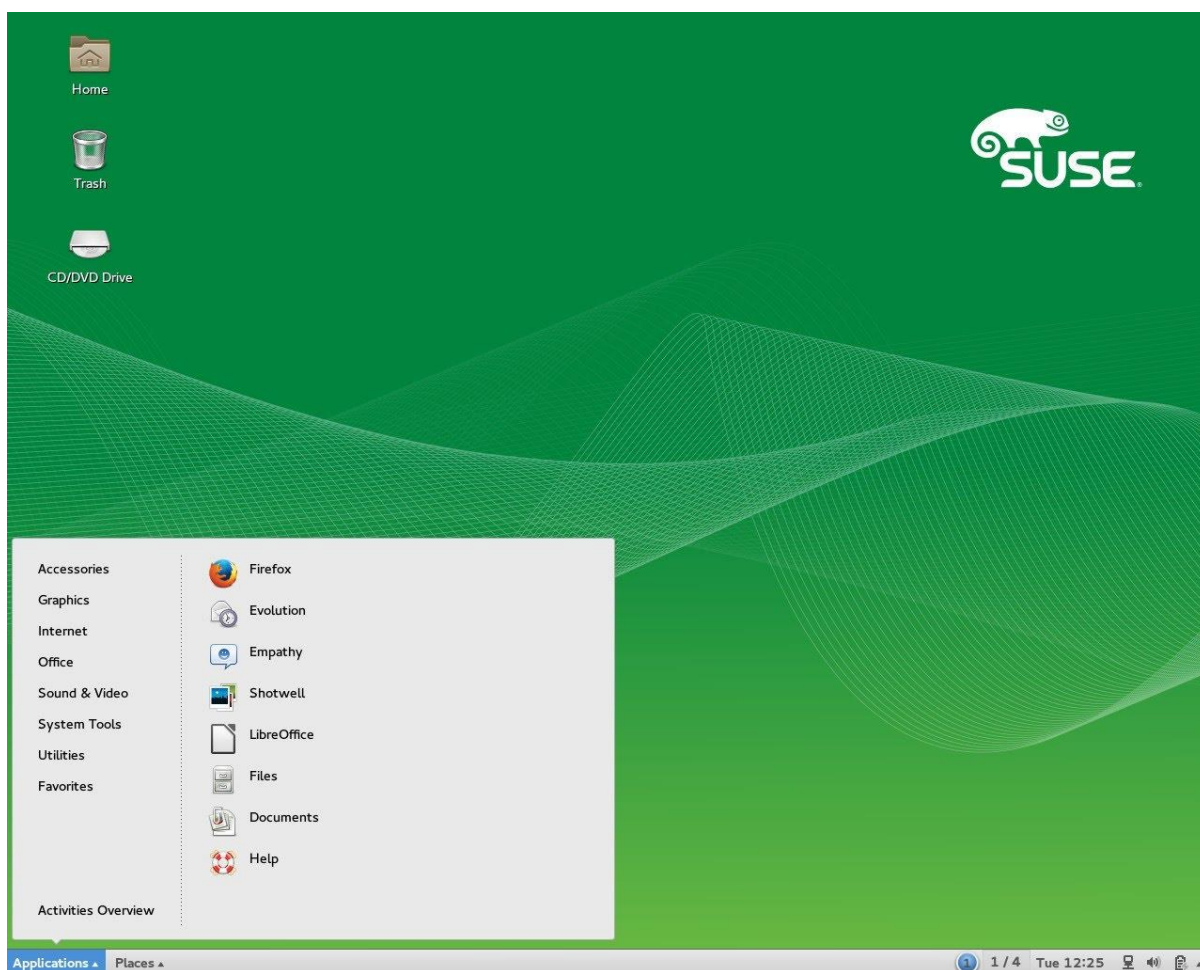


SUSE

SuSe es una de las principales distribuciones GNU/Linux existentes a nivel mundial, y su centro de producción está ubicado en Alemania.

Entre las principales virtudes de esta distribución se encuentra el que sea una de las más sencillas de instalar y administrar, ya que cuenta con varios asistentes gráficos para completar diversas tareas.

Utiliza el sistema de paquetes RPM (RedHat package manager) aunque no guarda relación con esta distribución. También, al igual que Red Hat, ha establecido una compañía dedicada a dar soporte a empresas, aunque su éxito ha sido relativo.



UBUNTU

Ubuntu es una distribución GNU/Linux basada en Debian GNU/Linux. Proporciona un sistema operativo actualizado y estable para el usuario, con un fuerte enfoque en la facilidad de uso y de instalación del sistema. Al igual que otras distribuciones se compone de múltiples paquetes de software normalmente distribuidos bajo una licencia libre o de código abierto.

Está patrocinado y financiado por Canonical Ltd., una compañía británica propiedad del empresario sudafricano Mark Shuttleworth.

Cada seis meses se publica una nueva versión de Ubuntu la cual recibe soporte por parte de Canonical durante dieciocho meses por medio de actualizaciones de seguridad, parches para bugs críticos y actualizaciones menores de programas.

Las versiones LTS (Long Term Support), que se liberan cada dos años normalmente, reciben soporte durante tres años en los sistemas de escritorio y cinco para la edición orientada a servidores.

Ubuntu soporta oficialmente dos arquitecturas de hardware: Intel i386 y AMD64. A partir de Ubuntu 9.04 (abril de 2009) se empezó a ofrecer soporte oficial para procesadores ARM.

Esta distribución ha sido y está siendo traducida a más de 130 idiomas, y cada usuario es capaz de colaborar voluntariamente a esta causa, a través de Internet.



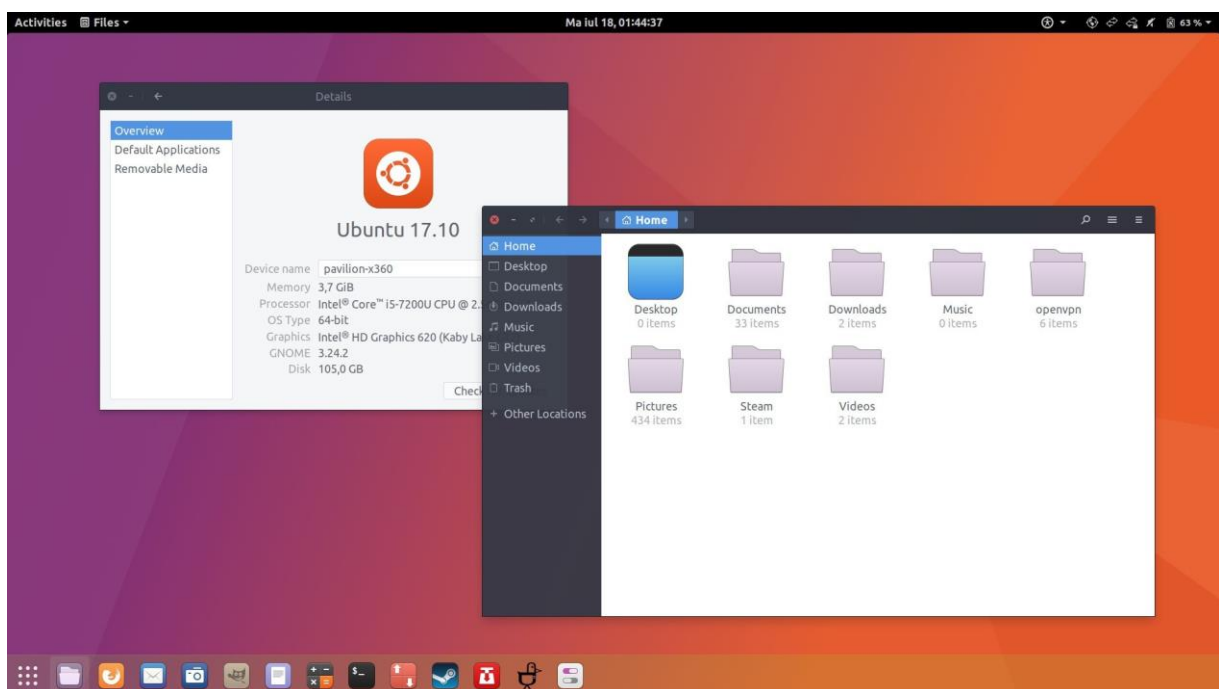
Versión	Nombre en clave	Lanzamiento	Fin de soporte
4.10	<i>Warty Warthog</i>	20/octubre/2004	2006
5.04	<i>Hoary Hedgehog</i>	8/abril/2005	2006
5.10	<i>Breezy Badger</i>	13/octubre/2005	2006
6.06 LTS	<i>Dapper Drake</i>	1/junio/2006	2011 (servidor)
6.10	<i>Edgy Eft</i>	26/octubre/2006	2008
7.04	<i>Feisty Fawn</i>	19/abril/2007	2008
7.10	<i>Gutsy Gibbon</i>	18/octubre/2007	2009
8.04 LTS	<i>Hardy Heron</i>	24/abril/2008	2013 (servidor)
8.10	<i>Intrepid Ibex</i>	30/octubre/2008	2010
9.04	<i>Jaunty Jackalope</i>	23/abril/2009	2010
9.10	<i>Karmic Koala</i>	29/octubre/2009	2011
10.04 LTS	<i>Lucid Lynx</i>	29/abril/2010	2015 (servidor)
10.10	<i>Maverick Meerkat</i>	10/octubre/2010	2012
11.04	<i>Natty Narwhal</i>	28/abril/2011	2012
11.10	<i>Oneiric Ocelot</i>	13/octubre/2011	2013
12.04 LTS	<i>Precise Pangolin</i>	26/abril/2012	2017 (servidor)
12.10	<i>Quantal Quetzal</i>	18/octubre/2012	2014
13.04	<i>Raring Ringtail</i>	25/abril/2013	2014
13.10	<i>Saucy Salamander</i>	17/octubre/2013	2014

14.04 LTS	<i>Trusty Tahr</i>	17/abril/2014	2019 (servidor)
14.10	<i>Utopic Unicorn</i>	23/Oct/2014	2015
15.04	<i>Vivid Vervet</i>	23/Abril/2015	2016
15.10	<i>Wily Werewolf</i>	22/Oct/2015	2016
16.04 LTS	<i>Xenial Xerus</i>	21/Abril/2016	2021 (servidor)
16.10	<i>Yakketi Yak</i>	13/Oct/2016	2017
17.04	<i>Zesty Zapus</i>	26/Abr/2017	Enero 2018
17.10	<i>Artful Aardvark</i>	13/Oct/2017	Julio 2018
18.04 LTS	<i>Bionic Beaver</i>		

Canonical intenta desde hace varios años que Ubuntu de beneficios. Buscando una monetización del proyecto, canonical intentó que Ubuntu fuera utilizado en teléfonos, tablets, televisores, etc.

Un problema que esta política ocasionó es que Canonical intentó unificar el escritorio de todos estos dispositivos mediante un interfaz de control conocido como Unity. Este interfaz es bastante incomodo de utilizar en PC y tiene unos requerimientos de proceso bastante elevados.

En la actualidad Ubuntu ha abandonado este enfoque debido a su nula aceptación por el mercado y ha vuelto a un escritorio más adecuado al PC, eliminado el Unity y utilizando Gnome, que es el entorno de escritorio habitual en la mayoría de las distribuciones Linux. Ubuntu en la actualidad es una opción muy interesante para desarrollos especiales como “cloud computing” o “big data” donde cuenta con soluciones automatizadas bastante interesantes, pero ha perdido bastante empuje en su versión escritorio.



GUADALINEX.

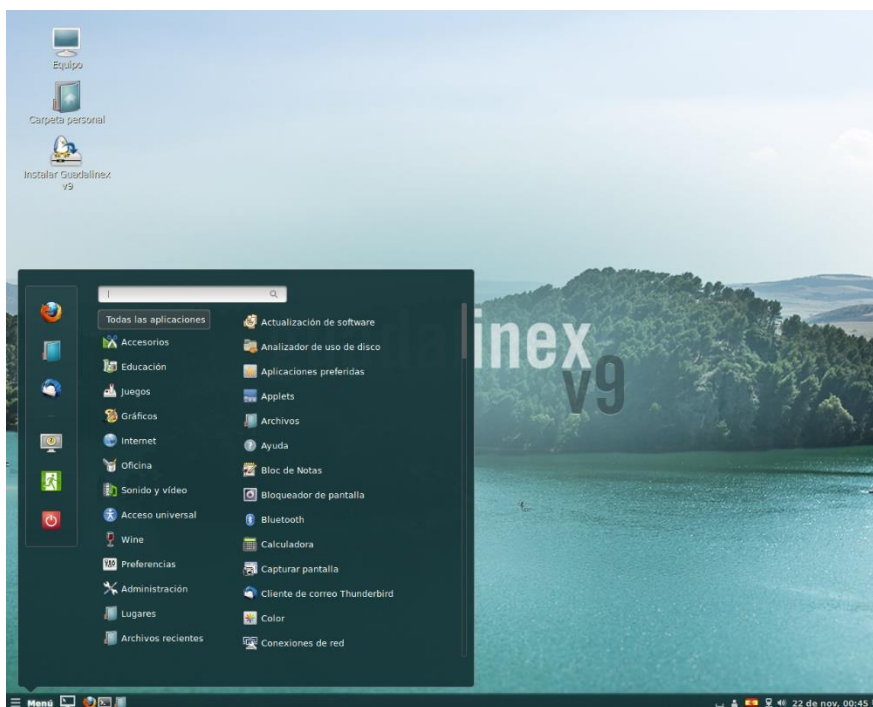
Guadalinex es una distribución Linux promovida por la Junta de Andalucía para fomentar el uso del software libre en su comunidad autónoma. Está inspirada en Gnu LinEx, un proyecto similar de la Junta de Extremadura basada en Debian. Inicialmente por lo tanto Guadalinex estuvo basada en Debian GNU/Linux debido al acuerdo inicial entre la Junta de Andalucía y la de Extremadura, pero desde la versión 3.0 se basa en Ubuntu.

Existen varios «sabores» de Guadalinex, según a qué público esté orientada:

- Guadalinex Base: de propósito general, se publica una vez al año.
- Guadalinex BIB: para Bibliotecas (en desarrollo).
- Guadalinex CDM: para los Centros de Día de Mayores.
- Guadalinex EDU: para los centros educativos, mantenida por el CGA (Centro de Gestión Avanzado).
- Guadalinex Guadalinfo: para los centros Guadalinfo.
- Guadalinex Mini: para ordenadores antiguos, lleva IceWM como gestor de ventanas.
- Guadalinex UCA: para la Universidad de Cádiz.



La última versión liberada es la Guadalinex 9.



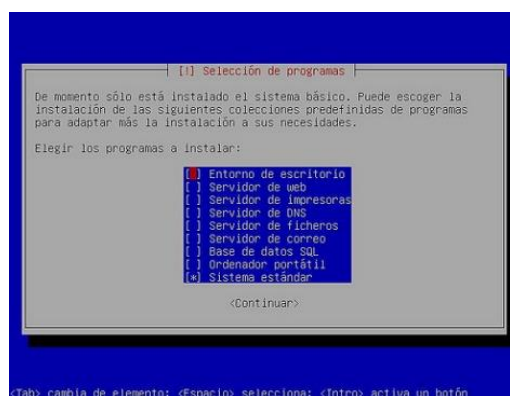
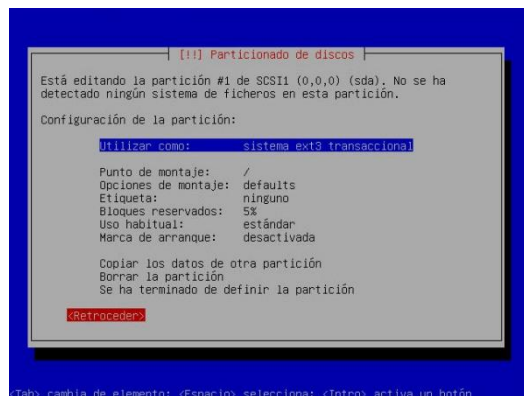
INSTALACIÓN DE LINUX.

La instalación de Linux es muy parecida a la que hemos visto en temas anteriores para otros sistemas operativos. La mejor manera de instalar Linux es arrancar directamente la máquina con Linux, bien mediante un CD o una unidad USB.

Un problema que nos vamos a encontrar, es que no existe una “instalación normal” de Linux. Cada distribución (distro) de Linux dispone de su instalador propio, siendo distintos los instaladores de SuSe o Red Hat, por poner un ejemplo. Incluso varían los instaladores de una versión a otra de la misma distribución.

En general, veremos que los pasos básicos de la instalación serán más o menos estos:

- 1) Iniciar la máquina con el arranque de Linux.
- 2) Detección del hardware básico para la instalación (CD, HD, teclado, pantalla, ratón, tarjeta de red, etc.)
- 3) Elegir el tipo de instalación (normalmente, siempre es conveniente escoger personalizada, manual o experta).
- 4) Crear una partición para instalar en ella Linux, y seleccionar dicha partición.
- 5) **Montar** la partición creada en un punto de montaje como raíz. Montar otras particiones si es necesario.
- 6) Crear una partición para utilizarla como **swap** o memoria de intercambio.
- 7) Escoger los paquetes que queremos instalar.
- 8) Copiar archivos.
- 9) Instalar el gestor de arranque.
- 10) Reiniciar la máquina para ejecutar ya nuestro Linux.

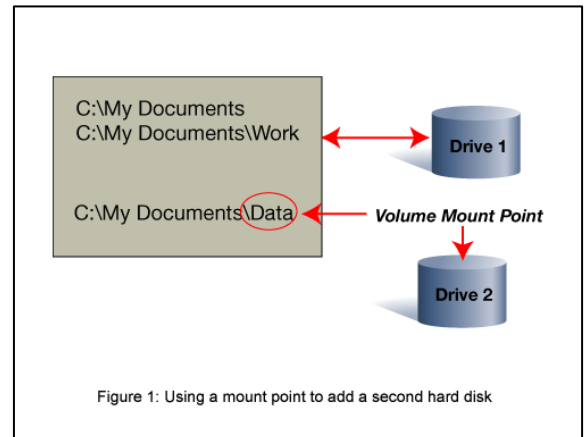


En estos pasos, he hablado de tres puntos que no hemos realizado cuando instalamos sistemas operativos tipo Windows; puntos de montaje, partición swap y gestor de arranque.

PUNTOS DE MONTAJE.

Normalmente estamos acostumbrados a la forma en que los sistemas operativos de Microsoft denominan a los medios de almacenamiento secundarios, asignando una letra a cada volumen, de modo que la disquetera es la A: la primera partición del primer disco duro es la C: la siguiente la D: etc. En Linux todo esto cambia.

En primer lugar, veamos como Linux referencia a los propios discos duros. Así, el primer disco duro de nuestra maquina en Linux se conoce como `/dev/hda` (si es paralelo) o `/dev/sda` (si es serial).



/ indica el root o raíz del árbol de Linux (En Linux solo existe un árbol)

dev nos indica el directorio donde se almacenan todos los dispositivos (devices)

/hda nos indica que nos referimos al Hard Disk (hd paralelo) con la letra a, es decir, el 1º.

`/dev/hda` – Dispositivo maestro en la IDE 1.
`/dev/hdb` – Dispositivo esclavo en la IDE 1.
`/dev/hdc` – Dispositivo maestro en la IDE 2.
`/dev/hdd` – Dispositivo esclavo en la IDE 2.
`/dev/sda` - Dispositivo serie en el bus serial 1.
`/dev/sdd` – Dispositivo serie en el bus serial 4.

Cuando referenciamos particiones, se utiliza un número a continuación del nombre del disco duro. Este número representa la partición. Así, `/dev/hda2` nos indica que nos referimos a la segunda partición del disco duro maestro de IDE 1. Como en un disco duro no pueden existir más de cuatro particiones primarias, estas reciben números del 1 al 4. Si creamos una partición extendida, esta no recibe ningún número (igual que en Windows no se le asigna una letra) y a las unidades lógicas que se crean dentro de dicha partición extendida se le asignan números a partir del 5. Veamos algunos ejemplos.

`/dev/hdb1` - Primera partición primaria del disco duro 2 (esclavo en el bus IDE 1).
`/dev/hda5` - Primera unidad lógica del disco duro 1 (maestro en el bus IDE 1).
`/dev/sdc7` - Tercera unidad lógica del disco duro 3 (SATA en el bus 3).

Como comentamos anteriormente, Linux no utiliza “letras” para acceder a las particiones que creamos, así que ¿Cómo podemos grabar un archivo por ejemplo en `/dev/hdb6`? En Windows nos limitaríamos a grabar el archivo en D: o E: o la letra que asignemos a esa partición, pero esto no se hace así en Linux.

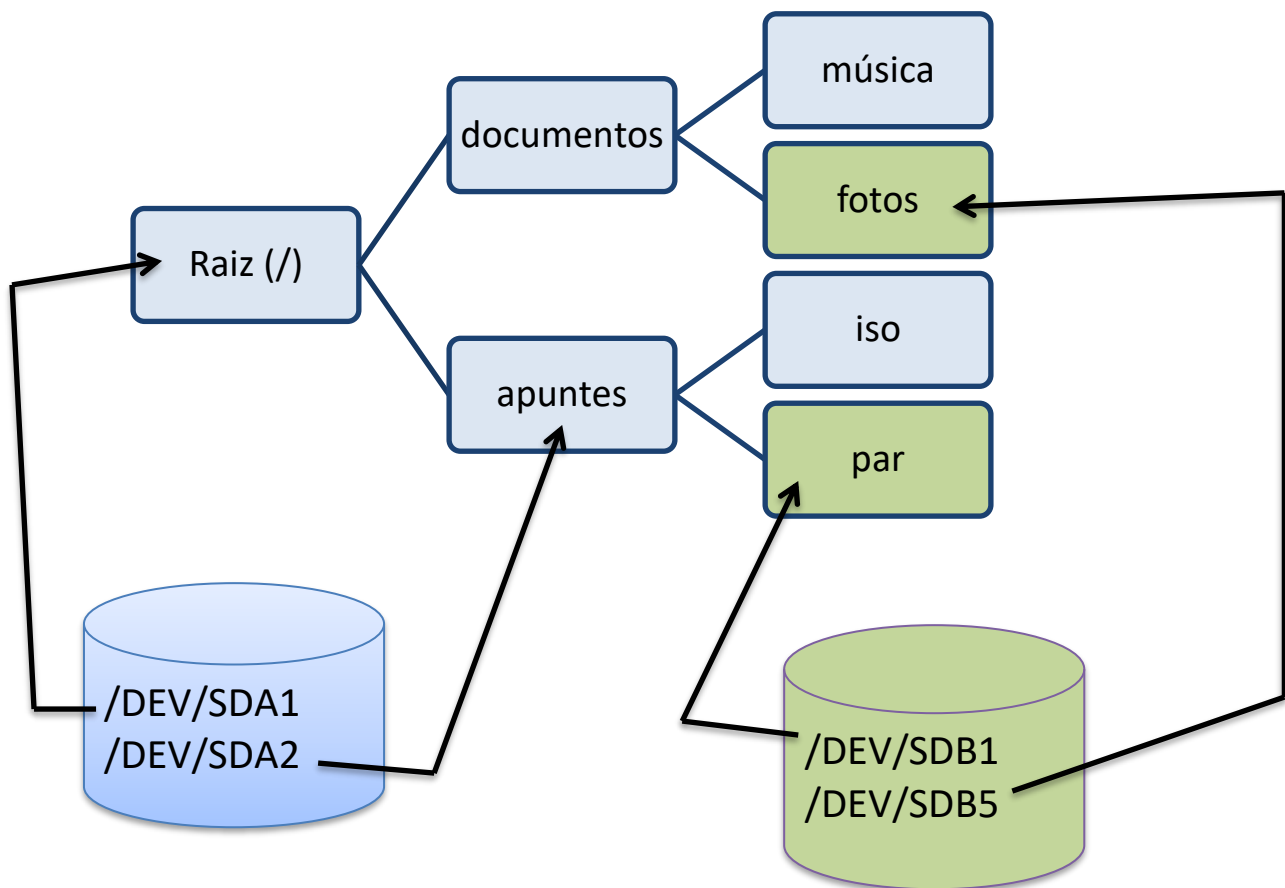
En Linux, cada dispositivo de almacenamiento (partición, disquete, CD) que utilizemos debe ser **montado** en nuestro árbol mediante un punto de montaje. En Linux solo existe un espacio de almacenamiento, un único árbol que empieza en la raíz (root) y que contiene todo lo que tenemos en nuestro sistema. Esto se consigue asociando cada partición a un directorio mediante un proceso denominado montaje.

Montar una partición hace que su espacio de almacenamiento se encuentre disponible accediendo al directorio especificado (conocido como punto de montaje).

Por ejemplo, si montamos la partición `/dev/hda5` en `/usr`, significa que todos los ficheros y directorios a partir de `/usr` residen físicamente en `/dev/hda5`.

Por lo tanto, el fichero `/usr/doc/FAQ/txt/Linux-FAQ` estará almacenado en `/dev/hda5`, cosa que no ocurre con el fichero `/etc/X11/gdm/sessions/gnome`.

Es absolutamente obligatorio montar al menos el root o raíz (`/`) durante la instalación.



PARTICIÓN SWAP.

Vimos cuando tratamos los temas sobre conceptos de Sistemas Operativos, que existía una técnica conocida como paginación de memoria, que nos permitía ofrecer a los programas más memoria de la que existe físicamente en la máquina, usando para ello una memoria virtual que en realidad existía en el disco duro.

En Windows esta técnica utiliza un archivo de intercambio que es gestionado directamente por Windows, y se suele llamar `pagefile.sys`. Linux no crea ningún archivo de intercambio (a menos que le obliguemos), sino que utiliza una partición entera para este fin, conocida como partición Swap. Simplemente tenemos que crearla en la instalación de Linux, y el sistema se encarga de usarla, sin tener nosotros que montarla ni nada por el estilo.

El tamaño que se le suele dar a una partición Swap, es el doble de la memoria RAM que tengamos instalado en nuestro sistema, sin exceder nunca los 2 GB de Swap. Esta es una regla general, aunque en cada caso particular puede que el tamaño ideal de Swap sea distinto.

En linux también tenemos la posibilidad de no utilizar una partición para este cometido y establecer un fichero tal como lo hace Windows. Lo veremos más adelante.

GESTOR DE ARRANQUE.

Vimos en apuntes anteriores como en el proceso de inicio de Windows el encargado final de cargar el SO era el gestor de arranque de Windows. Estudiamos cómo funcionaban estos gestores de arranque y como podían configurarse.

GNU-Linux sin embargo no tiene un gestor de arranque predeterminado como en el caso de Windows, sino que puede utilizar cualquier gestor de arranque de otras compañías. Existen muchos de estos gestores, el más usado hace un tiempo era el Lilo que hoy en día ha sido sustituido por el GRUB y aún más recientemente por el GRUB versión 2.

LILO.

Lilo (Linux LOader) es un gestor de arranque que permite elegir el sistema operativo a cargar al momento de iniciar un equipo con más de un sistema operativo disponible. No es capaz únicamente de arrancar Linux, sino que también puede arrancar otros sistemas operativos.

LILO funciona en una variedad de sistemas de archivos y puede arrancar un sistema operativo desde el disco duro o desde un disco flexible externo. LILO permite seleccionar entre 16 imágenes en el arranque. Al iniciar el sistema LILO solamente puede acceder a los drivers del BIOS para acceder al disco duro. Por esta razón en BIOS antiguas el área de acceso está limitada a los cilindros numerados de 0 a 1023 de los dos primeros discos duros. En BIOS posteriores LILO puede utilizar sistemas de acceso de 32 bits permitiéndole acceder a toda el área del disco duro.

En las primeras distribuciones de Linux, LILO era el gestor de facto utilizado para arrancar el sistema. En la actualidad es una segunda opción en favor del gestor de arranque GRUB.

El archivo lilo.conf se localiza típicamente en el directorio /etc y es la forma de configurar el gestor lilo. Dentro de lilo.conf se pueden encontrar dos secciones. La primera sección, que define opciones globales, contiene parámetros que especifican atributos sobre la localización del cargador. La segunda contiene parámetros asociados a las imágenes del sistema operativo que van a ser cargadas.

La información específica sobre su definición se puede encontrar en página del Manual de lilo.conf. Veamos un ejemplo del contenido de un fichero lilo.conf

```
boot = /dev/hda      # la partición de la que se arranca.
delay = 10           # tiempo durante el que aparecerá el menú.
image = /boot/vmlinuz # El fichero con el núcleo de Linux.
root = /dev/hda1     # La partición donde se monta la raíz del árbol.
label = Linux        # Nombre que aparece en el menú.
read-only            # opciones de montaje.
other = /dev/hda4    # Otros sistemas operativos en el sistema.
label = windows      # Nombre que aparece en el menú.
```

GRUB (VERSIÓN 1).

GRUB se carga y se ejecuta en 4 pasos:

1. La primera etapa del cargador es muy pequeña y se almacena en el MBR del disco duro, desde donde es leída por el BIOS.
2. La primera etapa carga el resto del cargador (segunda etapa). Si la segunda etapa está en un dispositivo grande, se carga una etapa intermedia (llamada etapa 1.5), la cual contiene código extra que permite leer cilindros mayores que 1024.
3. La segunda etapa muestra el menú de inicio de GRUB. Aquí se permite elegir un sistema operativo junto con parámetros del sistema.
4. Cuando se elige un sistema operativo, GRUB carga en la CPU el principio de este SO. (Este sistema operativo puede ser un Linux y entonces carga en la CPU el núcleo de Linux, o bien un Windows que se carga ya que GRUB le cede el control al gestor de arranque de Windows).

GRUB soporta tres interfaces: un menú de selección, un editor de configuración y una consola de línea de comandos.

GRUB no presenta el problema que presentaba LILO de depender exclusivamente del BIOS del sistema, pero a cambio tiene que ser capaz de trabajar con los sistemas de ficheros de los volúmenes de datos. GRUB en esta versión 1 por ejemplo no puede trabajar con el sistema de ficheros ext4 sino solo con los sistemas ext2 y ext3.

GRUB versión 1 se configura desde el fichero menu.lst, que suele estar almacenado en /boot/grub/menu.lst. Veamos un ejemplo de dicho fichero:

```
default      0
timeout     15
#Debian Etch
title        Debian GNU/Linux Etch, kernel 2.6.18-4-686 (on /dev/hdb1)
root         (hd1,0)
kernel       /boot/vmlinuz-2.6.18-4-686 root=/dev/hdb1 ro
initrd       /boot/initrd.img-2.6.18-4-686
boot
#Microsoft Windows 7 on /dev/hda1
title        Microsoft Windows XP Home Edition
root         (hd0,0)
makeactive
chainloader  +1
```

GRUB (VERSIÓN 2).

La versión 1 de GRUB tenía varios problemas arrastrados desde su inicio, siendo el principal de ellos depender de un archivo de configuración. Se pensó en realizar un programa con una configuración modular y automática, que pudiera adaptarse más fácilmente a los cambios de sistemas de archivos, instalación de nuevos sistemas operativos, borrados de los mismos, etc.

El archivo `/boot/grub/grub.cfg` reemplaza el antiguo `/boot/grub/menu.lst` pero a diferencia de este último el archivo de configuración para la nueva versión es generado automáticamente a partir del archivo `/etc/default/grub` y los scripts ubicados en `/etc/grub.d` que son, si no hemos agregado alguno:

- 00_header Carga las opciones del archivo `/etc/default/grub`
- 05_debian_theme Configuración del tema: imagen de fondo y color de texto
- 10_hurd Para kernel Hurd
- 10_linux Para kernel Linux
- 30_os-prober Genera entradas para otros sistemas operativos instalados
- 40_custom Para agregar entradas a mano

Dada la importancia de este gestor GRUB 2 y lo complicado de entenderlo sin verlo en funcionamiento, le dedicaremos un tema exclusivo más adelante.

