

**Software Libre:
Un estudio detallado**

(07/05/2003)

- Autores -

Sergio Blanco Cuaresma

<sergio at marblestation dot com>

<http://www.marblestation.com>

Jonathan Hernández Velasco

<jonathanhernandez at iespana dot es>

Tabla de contenidos

| | |
|---|----|
| Panorama actual..... | 2 |
| Historia..... | 3 |
| Definición..... | 4 |
| Anécdota universitaria..... | 5 |
| La calidad del Software Libre: mito..... | 6 |
| Licencias..... | 6 |
| Licencias Propietarias..... | 7 |
| Licencias Libres..... | 7 |
| Licencia GPL (Versión 1 y 2)..... | 7 |
| LGPL (Versión 2 y 2.1)..... | 8 |
| FDL..... | 8 |
| BSD..... | 8 |
| Open Source (Código Abierto) vs Free Software (Software Libre)..... | 8 |
| Distribuciones GNU/Linux..... | 9 |
| Distribuciones comerciales..... | 9 |
| Red Hat..... | 9 |
| Mandrake..... | 10 |
| UnitedLinux..... | 10 |
| Distribuciones no comerciales..... | 10 |
| Debian..... | 10 |
| LinEX..... | 11 |
| Slackware..... | 11 |
| Gentoo..... | 11 |
| Alternativas Software Libre..... | 12 |
| Software libre en las Universidades (notas finales)..... | 12 |
| Presentación del proyecto GPL URV (07/05/2003)..... | 13 |

Panorama actual

Imaginemos que en un parking cualquiera, el 90% de los coches fuesen de la misma marca, pongamos por ejemplo SEATs.

Imaginemos que cada vez que las líneas de las carreteras volviesen a pintarse, tendríamos que comprar otro coche que fuera compatible con el nuevo trazado.

Imaginemos que en principio sólo se puede sentar el dueño del coche, pero se podría abonar una tarifa extra por cada pasajero que llevásemos.

Imaginemos que si pusierais algún complemento al coche que no fuese de la marca SEAT, sino de la competencia, el coche se volviese automáticamente un 50% más lento.

Imaginemos que las puertas del coche se bloqueen con frecuencia sin razón aparente. Y para poder abrirlas tuviésemos que utilizar algún truco como apretar el pito al mismo tiempo que giramos la llave de contacto y agarramos la antena de la radio.

Imaginemos que todos esos coches mientras están en funcionamiento, cada X horas deciden pararse obligando al conductor a apartarse a un lado y arrancar de nuevo. Por supuesto, sin explicación alguna.

Imaginemos que en ocasiones, al intentar realizar ciertas maniobras como por ejemplo girar a la izquierda, el coche se para y no podemos volver a arrancarlo. Y que la única solución sería reinstalar de nuevo todo el motor.

Imaginemos que un buen día decidimos arreglar nuestro coche, lo llevamos al mecánico de nuestro barrio y este nos dice: "lo siento, yo no puedo hacer nada, si abro el capot estoy infringiendo la ley, debes llevarlo al taller oficial de SEAT".

Imaginemos que lo llevamos al taller oficial de SEAT para que lo arreglen y allí nos dicen: "Tendremos en cuenta sus problemas, lo estudiaremos y quizás para el siguiente modelo que saquemos, y solo quizás, pondremos alguna solución a su problema".

Resignados, imaginemos que conseguimos encontrar un taller alternativo no oficial donde hay un mecánico dispuesto a saltarse la ley para solucionar nuestro problema, abre el capot y se encuentra en lugar de encontrar el motor, la batería, bujías, etc.. se encuentra con una caja negra donde todo está integrado y de la cual no dispone ningún esquema, viéndose imposibilitado a la hora de intentar encontrar o solucionar cualquier problema.

Y ahora llega el momento de reflexionar... ¿aceptaríamos una situación así? Por

supuesto que no, sería inaceptable! Entonces, ¿Porque este mismo ejemplo pero trasladado al mundo del software es comúnmente aceptado? Ha llegado el momento de abrir los ojos.

Hay compañías de este ámbito que tienen una cuota de mercado del 90%, eso mismo en cualquier otro sector sería intolerable. Pero no es solo eso, el problema principal está en la industria del software en su conjunto que nos ofrece software, o mejor dicho, nos ofrecen licencias de uso abusivas sobre su software. ¿Por que digo abusivas? Pongamos un ejemplo bien sencillo, compramos un Microsoft Windows, o lo que estaría mejor dicho, compramos una licencia para utilizar un sistema operativo Microsoft Windows, llegamos a casa y resulta que tenemos 2 ordenadores, uno en la habitación y otro en el comedor, los cuales solo los uso yo, pues bien, la licencia que acabo de comprar para utilizar MS Windows solo me permite instalarme dicho sistema operativo en uno de los 2 ordenadores, si lo quiero en los 2 debo comprar otra más. Es más, imaginemos que tenemos 1 ordenador con 2 procesadores, ¡resulta que también tendríamos que comprar 2 licencias! Pero no nos quedemos con la idea de lo mala que es Microsoft, no, no estamos haciendo una crítica directa contra Microsoft, pretendemos criticar el software en general que se está vendiendo hoy en día, este tipo de licencias no solo las utiliza Microsoft, sino que la inmensa mayoría de empresas por desgracia también lo hace.

Historia

Allá por el 1971, cuando la informática todavía no había sufrido su gran boom, las personas que hacían uso de ella en ámbitos universitarios y también empresariales creaban y compartían el software al igual que nuestras madres crean y comparten recetas de cocina, era algo normal y habitual. En aquella época, a dicho software ni tan siquiera se le llamaba software libre, era simplemente software.

Llegaron los 80 y la situación empezó a cambiar, las computadoras más modernas de aquella época comenzaban a utilizar sus propios sistemas operativos propietarios, los cuales para poder usarlos debían firmar acuerdos en los que garantizabas a las empresas que no modificarías su software para nada. Es decir, si existía algún error, te impedían que lo arreglases tu mismo y debías pedirle a la compañía que lo hiciese por ti. Esto lo que provocaba era la destrucción absoluta de comunidades cooperativas, en las que si tu vecino tenía un problema con cierto software y tu podías arreglarlo, lo hacías sin ningún inconveniente y podías compartir dichas soluciones perfectamente. Este tipo de software se estaba convirtiendo en algo anti- social.

Fue en esta época en la que un señor llamado Richard Stallman, que trabajaba en el laboratorio de IA del MIT (Massachusetts Institute of Technology), se dio cuenta que las cosas estaban cambiando. En su trabajo habían recibido una impresora donada, la cual era utilizada por todos los trabajadores, el problema que presentaba ese dispositivo era que se trababa el papel frecuentemente y cuando esto sucedía no se

enviaba ninguna señal a los ordenadores que hacían uso de ella, es decir, cuando la gente iba a buscar sus trabajos impresos se encontraban con que la impresora llevaba bloqueada horas sin que nadie se hubiese dado cuenta. Entonces Stallman decidió arreglar el mismo el problema, busco los controladores del dispositivo pero solo encontró los binarios, entonces pidió a la compañía de la impresora que le facilitasen el código fuente para solucionar un problema que tenían, y esta se negó.

Fue en ese momento cuando se vio en una encrucijada, debía elegir entre aceptar el nuevo software propietario firmando los nuevos acuerdos de no revelación y acabar desarrollando más software propietario con licencias restrictivas que deberían ser más adelante aceptadas por colegas suyos con los que anteriormente había compartido su código libremente, o no aceptar ese nuevo mundo propietario y no contribuir a hacer un mundo más dividido y con menos libertades. Evidentemente el primer camino, el aceptar el software propietario, tenía un aliciente económico importante, así que la encrucijada se podía resumir entre ganar dinero sin importarme nada más o intentar evitar que el mundo se convierta en un lugar un poco más injusto.

Este buen hombre se decantó por la segunda opción, e inició un proyecto para intentar formar una comunidad de personas, en las que compartir el código volviese a ser algo natural, una comunidad donde se volviesen a recuperar las libertades del software, porque no nos engañemos, el software nació libre y Stallman lo que hizo fue intentar recuperar esa libertad. Este proyecto fue llamado GNU (GNU's Not Unix) y consistía en construir un sistema operativo compatible con los UNIX pero completamente libre.

Definición

Software libre en inglés es Free Software, cosa que lleva a errores. Free en Inglés significa tanto gratis como libre y puede dar lugar a confusión. Por eso últimamente se oye bastante a los ingleses hablar de Libre Software, utilizan el adjetivo en castellano, así no se puede confundir con gratuito. Y lo cierto es que he observado que hay una idea entre las personas que ya conocen el Software libre, creen que todo software libre es gratuito, y esto NO es cierto. El software libre no tiene porque ser gratuito, se puede pagar perfectamente por él.

Para saber si un programa es libre este debe cumplir las 4 libertades siguientes:

Libertad 1: Usted tiene libertad para ejecutar el programa, con cualquier propósito y sin restricciones. No te pueden obligar a ejecutarlo solo en un numero determinado de máquinas o en unas condiciones específicas.

Libertad 2: Usted tiene la libertad para modificar el programa para adaptarlo a sus necesidades o para estudiar cómo funciona. Como cualquier informático sabe, para que esta libertad sea efectiva, se debe tener acceso al código fuente, porque

modificar un programa sin disponer del código fuente es imposible.

Libertad 3: Usted tiene la libertad para redistribuir copias, tanto gratis como cobrando por ellas.

Libertad 4: Usted tiene la libertad para distribuir versiones modificadas del programa, de tal manera que todo el mundo pueda beneficiarse con sus mejoras.

Si un programa cumple estas 4 libertades, entonces se trata de Software Libre.

Llegados a este punto sería importante reflexionar sobre nuestras preferencias:

- Software propietario con licencias de uso restrictivas, que no te imponen condiciones para su uso y te impiden adaptarlo a tus necesidades o....
- Software libre, que te permite modificarlo sin restricciones y que cuando lo compras, realmente estas comprando el programa y no una licencia de uso.

Parece bastante intuitiva la respuesta, el Software Libre nos reporta unos beneficios mayores al propietario. Por tanto es importante que todos exijamos que el software que compremos/usemos sea libre, no aceptemos las condiciones establecidas por las licencias propietarias como algo normal. Por supuesto no basta con esto, sino que todos los que desarrollamos software deberíamos pensar en no cometer los mismos errores que algunas empresas y hacer de nuestro producto un beneficio para nuestros clientes potenciales.

Anécdota universitaria

La mayoría de programadores universitarios suelen tener la tendencia a pensar que si desarrollamos un programa, debemos ocultar el código fuente, no liberarlo, no vaya a ser que se aprovechen de nuestro trabajo. Es necesario cambiar esa mentalidad y pensar en la necesidad de compartir el conocimiento, y en extensión el software, y si lo que se quiere es ganar dinero desarrollando, pues centralizar el negocio en vender servicios y no únicamente el software en si. Lo cierto es que no me extraña que la mayoría tenga esa idea de no compartir el código fuente, es un poco lo que hemos visto siempre, las empresas hacen eso, estamos acostumbrados a ello, incluso en la universidad se ve esa tendencia simplemente en la forma de hablar. Por ejemplo, recuerdo una clase que tuve hace tiempo donde el profesor tenía la mala costumbre de decir cosas como... "claro, una página web con javascript, cuando llega al cliente, todo el código del javascript es visible y por lo tanto fácilmente pirateable". Este es un enfoque incorrecto, si el código javascript es visible tenemos la ventaja de que podremos aprender de él, ¿porque empeñarnos en que compartir dicho conocimiento es malo? Hay que cambiar nuestra mentalidad.

Así que dentro del ámbito universitario no solo los alumnos, sino los profesores

deberían fomentar el uso y desarrollo de software libre. No solo utilizando herramientas libres para las practicas, sino también educando a sus alumnos sobre el tema, por ejemplo, que en las asignaturas de Ingeniería del Software se explique como tienen lugar desarrollos de aplicaciones libres, o otro ejemplo, que en asignaturas relacionadas con los sistemas operativos se explique correctamente cual es la filosofía del software libre y como se construyó uno de los productos estrella: GNU/Linux. Incluso tener una asignatura de libre elección sobre el Software libre, hay otras universidades que la tienen, como por ejemplo la Universidad Politécnica de Madrid o la Universidad Rey Juan Carlos I.

La calidad del Software Libre: mito

Existe un gran mito al respecto del Software Libre, seguramente cualquiera que haya oído hablar de esta filosofía quizás también haya escuchado también gente que afirma que el Software Libre es de mayor calidad. Esto no es en absoluto cierto, imaginemos que acabamos de aprender a programar y decidimos hacer nuestro primer programa, y hacemos que este sea libre. Seria de locos creer que nuestro primer programa va a ser de gran calidad, en cambio si es Software Libre. Por tanto, las cosas se pueden hacer mal tanto con Software Libre como con Software propietario.

Lo que si es cierto es que ese programa de mala calidad que hemos hecho, al ser libre, cualquier otra persona podría cogerlo, mejorarlo y volver a liberar el código, otro individuo podría coger esa mejora y le añadirla una funcionalidad que le interesa y la cual no tenia, y así sucesivamente. Es decir, poco a poco, al ser libre y tener la posibilidad de mejorarlo y adaptarlo a nuestra necesidades, ese programa que en principio era algo de baja calidad se puede ir reciclando y mejorando consiguiendo llegar a una calidad aceptable. Este mismo ejemplo con software propietario seria imposible, ya que no podemos acceder a su código fuente y no nos permiten hacer modificaciones ni para mejorarlo ni para adaptarlo a nuestras necesidades. Por tanto, si un programa propietario es de baja calidad y la empresa decide no continuar invirtiendo dinero en él, así se quedará.

Licencias

Existen muchos tipos de licencia en el mundo del software, pero podríamos clasificarlas en 3 categorías a groso modo:

- Licencias Propietarias
- Licencias Libres
- Licencias Abiertas

• Licencias Propietarias

Las **Licencias Propietarias** son las que todos conocemos y que desgraciadamente estamos más acostumbrados a acatar y aceptar. Aunque, en realidad, y sobre todo aquí en España, es donde menos caso se le hace. Aunque en general no se les preste mucha atención las licencias son serias, plantean restricciones con cargos legales grandes si se incumplen, y hay organismos que se dedican únicamente evitar que no sean respetadas.

Las licencias propietarias, en general, vienen a significar que no se puede acceder al código fuente de la aplicación, que esta no puede ser modificada ni se puede compartir o copiar y que sólo se puede instalar en 1 ordenador (o en un ordenador con 1 sola CPU).

• Licencias Libres

Cuando hablamos de **licencia libre** nos viene una palabra a la cabeza: GPL. La licencia GPL tiene sus años ya, la versión 1.0 data de Febrero del 89. Es la formalmente conocida como GNU GPL (GNU General Public License). Es la base de las demás licencias libres. Normalmente, cuando se va a crear una licencia a un software, si pretende ser libre, primero se comprueba la GPL y a partir de ella, si no se esta de acuerdo en alguna punto, se adapta hasta hacer la licencia a medida. Es por esto que existen muchas licencias libres: licencia apache, bzip2, php, postgresql, mozilla, sendmail... Normalmente se les suele bautizar la licencia con el nombre de la aplicación.

Veamos más detalladamente lo que nos permite la GNU GPL y derivadas:

–Licencia GPL (Versión 1 y 2)

La diferencia entre versiones (GPL, GPL-2) reside en pequeñas cuestiones que se van añadiendo conforme a los avances i los nuevos problemas, por ejemplo la problemática con las patentes. Para más información sobre patentes y sus consecuencias [http:// patents.caliu.info](http://patents.caliu.info).

Como las diferentes versiones no son más que pequeñas modificaciones, generalmente no nos tiene que importar la versión de la GPL que le ponemos a nuestro programa porque el corazón de la licencia nunca se va tocar de una versión a otra. Por norma general los programadores o no especifica versión o indica la última versión disponible cuando pone a disposición pública su software.

–*LGPL (Versión 2 y 2.1)*

El principal problema que surgió con la GPL es que ésta no permite que un programa libre se junte a nivel de código con un programa propietario. Esta cláusula existe para que cualquiera que desarrolle y venda software propietario no se pueda aprovechar del código de un programa libre sin aportar nada a cambio. Si alguna empresa tiene un programa propietario y quisieran añadirle alguna funcionalidad de un programa con licencia GPL, la única forma de reutilizar el código de ese programa libre es liberando también el propietario, es decir, cambiándole la licencia. Pero hay un aspecto para el cual algunas personas sí están interesados en juntar software propietario con software libre: las librerías. Es más que probable que algún programador/empresa tenga que hacer alguna librería para otro programa propietario, y quiera que esta librería sea libre. La solución a este problema es la licencia LGPL o Lesser General Public License.

–*FDL*

Existe también una licencia con las mismas condiciones de la GPL pero adaptada al tema de la documentación, al texto en general. Se trata de la GNU FDL (Free Document License) que le puede interesar a programadores que hagan un manual de usuario de una aplicación o incluso a profesores cuando publican sus apuntes, transparencias o cualquier texto educativo.

–*BSD*

La licencia BSD es bastante polémica. Es una licencia libre pero con la característica especial de que cualquier persona puede coger el programa, realizar modificaciones y ofrecerlo de forma cerrada cambiando la licencia. No sólo el dueño del programa puede realizar dicha acción, sino que cualquiera puede coger el producto y hacerlo propietario. Por supuesto, el programa original que ya fue liberado con la licencia BSD sigue siendo libre. La utilidad principal de esta licencia es el interés que genera por parte de algunas empresas que no esta segura a liberar el código directamente a la GPL y desean reservarse la posibilidad de volver a cerrarlo. A pesar de esta característica es que es una licencia bastante habitual.

• **Open Source (Código Abierto) vs Free Software (Software Libre)**

Existe mucha confusión entre el termino Open Source y el Software Libre. No es lo mismo aunque los objetivos que persiguen sí. El movimiento Open Source coge las bases del Software libre pero se basa en el lado más práctico dejando, poniendo en segundo plano las razones éticas que tanto defiende el movimiento del Software Libre.

Los defensores del Software Libre argumentan que el acceso al código y las libertades sobre este es un derecho natural, mientras que el Open Source dice que abrir el código para el libre acceso es beneficioso en la practica para crear mejores aplicaciones. Por lo tanto, el Open Source o Código Abierto se centra más en el hecho de publicar el código fuente, de abrirlo, sin dar excesiva importancia en las restricciones que se le impone a este código fuente. Es por esto que no es lo mismo que un programa se libere a que se abra su código.

Las licencias Open Source son las que tienen cabida dentro del grupo que he llamado licencias abiertas. Un ejemplo de éstas es la licencia QPL, la de las librerías gráficas QT que utiliza KDE.

- A no ser que las utilices para desarrollo de un programa comercial, la licencia a aplicar es la GPL (libre). En caso contrario, necesitas una licencia de uso que hay que pagar a Trolltech (empresa creadora de QT).
- La QPL entonces no es propietaria porque pone a disposición del público el código fuente y es posible modificarlo, distribuirlo, etc. Pero no es libre porque es necesario pagar si se utiliza en proyectos comerciales, por tanto esta en el grupo de las abiertas.

Distribuciones GNU/Linux

El sistema operativo libre más usado es GNU/Linux (también comúnmente conocido como Linux). Respecto al nombre de este sistema Richard Stallman está algo molesto por el hecho de que se llama a GNU/Linux, simplemente Linux sin nombrar GNU y por tanto pierde importancia.

Un sistema base Linux se compone de las herramientas GNU más el kernel Linux, de que deba ser llamado GNU/Linux para hacer justicia. Actualmente es imposible arrancar/desarrollar Linux sin ninguna herramienta GNU.

Hay muchas distribuciones Linux, veamos un pequeño repaso de las distribuciones más populares.

• Distribuciones comerciales

–*Red Hat*

Red Hat es la distribución linux más usada del mundo. Prácticamente para toda aplicación de linux existe un precompilado en formato rpm, que es el sistema de paquetes originario de RedHat (que también utiliza Mandrake y UnitedLinux). Por lo tanto RedHat es un distribución más orientada a empresas que al usuario de escritorio.

–Mandrake

La distribución reina del escritorio para usuarios medios es Mandrake. Esta basada en RedHat y esta orientada al usuario con pocos conocimientos de informática. Mandrake suele ser muy recomendada a nuevos usuarios de GNU/Linux. Sus asistentes son muy conocidos, estos son visualmente atractivos y ofrecen ayuda para las tareas más comunes. Posee un instalador inicial muy sencillo que facilita la instalación del sistema.

–UnitedLinux

Para acabar con las distribuciones comerciales falta hablar de UnitedLinux, una iniciativa que tuvieron 4 empresas: Connectiva S.A, SCO Group (antigua Caldera), SuSELinux y TurboLinux. Estas compañías han firmado un acuerdo abierto (pueden entrar más compañías) para ir implementando y mejorando UnitedLinux, que será la distribución base a partir de la cual cada compañía la utilizará y añadirá cada una utilidades y mejoras propias que la diferenciarán del resto, pero sin tocar la base para no romper el estándar.

Lo que se consigue con este estándar es que de cara a las empresas de software, en vez de tener que certificar que su producto funciona bien en 4 distribuciones distintas, sólo sera necesario verificar que funciona correctamente en UnitedLinux ya que la base es común.

También cabe destacar el hecho de que haya 4 empresas conocidas detrás de UnitedLinux le da cierta fuerza en el mercado frente a RedHat, que no tenían antes por separado.

Un ejemplo claro es Oracle, que antes sólo daba soporte técnico a las plataformas RedHat y ahora también lo hace a UnitedLinux.

• Distribuciones no comerciales

–Debian

Como distribución no comercial, la más conocida y preferida es Debian. Esta es una distribución creada y mantenida desde el principio únicamente por los propios usuarios. Tiene su formato propio para empaquetar sus aplicaciones (.deb) y el conocido instalador “apt”. Con el apt se puede instalar en cuestión de segundos

cualquier paquete, resolviendo automáticamente las dependencias de éste. Los paquetes que se instalan son paquetes precompilados creados por los propios usuarios y testeados para que no contengan ningún error. Los controles de debian son muy estrictos, tanto que la versión estable de Debian (actualmente llamada Woody (3.0- r1)). A causa de su tremenda estabilidad y del tiempo que debe transcurrir para probar todas las aplicaciones, la rama estable ofrece software algo antiguo. Se puede considerar la distribución Linux más estable, ideal para sistemas de producción ininterrumpida.

–LinEX

Y LinEx es la distribución oficial de la junta de Extremadura (provincia de España) y esta basada en Debian. Extremadura ha sido la primera comunidad en España en apostar fuerte desde la administración pública por el Software Libre y LinEx es el estandarte. Los resultados no se han hecho esperar y debido al éxito que están teniendo es posible que otras comunidades adopten la misma política. La junta de Andalucía (provincia de España) ha sido la primera y hace poco (antes del 07/05/2003) firmó un acuerdo con la de Extremadura para difundir en su administración el uso de software libre. En Cataluña (provincia de España) se presentó un plan antes que en extremadura pero se rechazó.

–Slackware

Slackware fue de las primeras distribuciones. Tiene un sistema de paquetes basado en “.tgz” (no confundir con la abreviatura de tar.gz que a veces se le pone tgz, sobre todo en los paquetes para sistemas BSD).

Es más tradicional que las demás, suele ser más recomendable para usuarios algo expertos o que ya hayan profundizado con GNU/Linux.

–Gentoo

Acabamos la sección de distribuciones con la preferida de los autores del estudio: Gentoo Linux. Es una distribución muy joven de abril del año 2002 aproximadamente. Y cuenta ya con unos 4.000 paquetes/programas diferentes, que en la distribución se llaman ebuilds. En Gentoo está el comando “emerge” que es un cruce entre al apt-get de Debian y los ports de los BSDs. La función de emerge cuando se le pasa por parámetro el nombre de una aplicación es resolver las dependencias, bajarse el código fuente, compilarlas junto con la aplicación e instalarlo todo automáticamente (e.g. emerge mozilla). Se puede ajustar incluso los FLAGS del compilador para conseguir más optimización según tu CPU. El corazón del sistema de paquetes se llama portage y desde él se pueden ver todas las aplicaciones disponibles para instalar mediante emerge.

Alternativas Software Libre

La problemática de muchos a la hora de pasar al software libre o más concretamente a un sistema operativo GNU/Linux es el miedo a no tener esa aplicación preferida que tienes en tu sistema habitual. Pues bien, ahora comentaré algunas buenas alternativas a programas que normalmente utiliza un usuario en casa.

Ms Office -> OpenOffice

OpenOffice es una suite ofimática basada en la StarOffice de Sun, y que actualmente tiene millones de usuarios en todo el mundo. Tiene prácticamente todas las funciones que tiene MS Office además de las suyas propias, y es compatible con los formatos .doc, .pwt (powerpoint), xls (excel), etc. También puede interactuar con bases de datos en conjunto con mysql.

MS Internet Explorer -> Mozilla, Konqueror, Galeon... entre otros.

MSN Messenger -> AMSN (clon con soporte para envío de archivos), MSN4Lin, Gaim, Gabber, PSI, ... entre otros.

Msn NetMeeting -> GnomeMeeting

MS Outlook -> Evolution, Mozilla Mail

WinAmp -> XmmS

Winzip -> gzip, zip, bzip2

Windows Media Player -> Mplayer, Xine

Photoshop -> Gimp

FreeHand -> SodiPodi, Sketch

Software libre en las Universidades (notas finales)

Creemos que el software libre hoy en día es una alternativa real al software propietario, y que si hay soluciones libres igual o mejores que las propietarias no hay razón para utilizar las propietarias. Desde la universidad, cuna del conocimiento que dicen, el compartir es la principal meta. Es por eso que desde aquí dentro es desde donde tenemos que empujar más el uso y desarrollo de aplicaciones libres ya

sea para nuestros departamentos como para nuestros alumnos. Siempre que exista una alternativa libre de calidad, ésta debe de ser apoyada. ¿Pero qué pasa cuando esa alternativa no existe? Una muy fácil forma de conseguirla es haciéndola uno mismo. Bien desde algún proyecto interno de la universidad o desde proyectos de final de carrera de alumnos de informática. Se tiene que impulsar no sólo el uso, sino también la creación.

Presentación del proyecto GPL URV (07/05/2003)

Narra: Sergio Blanco Cuaresma

Para ser sinceros, desde que entre en la Universidad Rovira i Virgili de Tarragona tuve ganas de formar alguna especie de grupo de alumnos que compartiesen mis inquietudes sobre el software y el cual se dedicara a fomentar el uso de software libre. Pero no ha sido hasta ahora, en tercero de Ingeniería Técnica Informática de Gestión que he podido contar con el apoyo de Jonathan Hernández, una de las pocas personas que conozco que se ha adentrado profundamente en el mundo del software libre, así que entre los 2 nos hemos animado a formar un grupo de usuarios de software libre, o mejor dicho Grup d'usuaris de Programari Lliure de la URV (GPL URV) el cual tiene como objetivo fundamental fomentar el uso y desarrollo de software libre en nuestra universidad.

Ya existen muchas otra universidades que tienen grupos de usuarios similares, muchas de ellas son más bien grupos de usuarios de linux, nosotros lo hemos querido enfocar más hacia el software libre en general, porque consideramos que linux no es mas que otro ejemplo de software libre. Quizás el futuro no sea linux, quizás sea otro sistema operativo, pero nosotros lucharemos porque ese sistema sea libre.

Nuestra metas a más corto plazo son:

- La elaboración de una página web sobre el tema. El Departament d'Enginyeria Informàtica i Matemàtiques (DEIM) de la Universidad Rovira i Virgili nos apoya dejándonos una máquina para llevar a cabo esto. Dominio: **<http://www.gplurv.org>**
- Facilitar software libre a aquellos alumnos o profesores que nos lo pidan, de forma que solo tengan que pagar por el medio en el que se le entregue, ya sea el CD o el diskette.
- Y finalmente estamos a dispuestos a aceptar ayuda de todo aquel que quiera colaborar y/o ser miembro del grupo. Podéis enviarnos un mail a <gpl at etse dot urv dot es> diciéndonos que queréis colaborar y/o ser miembros del grupo y os informaremos de los datos que necesitamos que nos enviéis.