

Software libre:  
técnicamente viable,  
económicamente sostenible  
y socialmente justo

Software libre:  
técnicamente viable,  
económicamente sostenible  
y socialmente justo

Jordi Mas i Hernàndez



Esta obra es libre y está sometida a las condiciones de una licencia *Creative Commons*. Puede redistribuir, copiar, y reutilizar y mezclar este libro siempre que se haga con fines no comerciales y dando crédito a su autor Jordi Mas i Hernández . Para una copia completa de la licencia visitar la web: <http://creativecommons.org/licenses/by-nc/2.0>.

Zero Factory S.L.  
Av. Icaria, 205, 2<sup>a</sup> 1<sup>a</sup>  
Tel. 93 224 01 50. Fax 93 225 19 81  
08005 Barcelona  
info@infonomia.com  
<http://www.infonomia.com>

© Jordi Mas i Hernández  
Primera edición: 2005  
Depósito legal:  
ISBN:

Diseño de la cubierta: Nomadesign  
Impresión: Cargraphics  
Impreso en España – Printed in Spain

## PRÓLOGO

El software libre tuvo sus inicios en pequeños grupos de entusiastas y activistas que querían cambiar la industria del software. Después de muchos años esta comunidad ha crecido rápidamente e incorporado no solamente voluntarios en todo el mundo, sino atraído la atención y la colaboración de centenares de empresas claves.

El movimiento del software libre se ha desarrollado orgánicamente y cualquier instalación de éste está conformada de distintos componentes provistos por distintos participantes.

Inicialmente el software libre fue popular en los servidores y con el paso del tiempo y el trabajo de miles de voluntarios las piezas faltantes se completaron. Linux que antes estaba limitado a ser un sistema que no era visible a los usuarios finales --- Google y Amazon son ejemplos de sistemas contruidos sobre Linux--- ahora es un sistema que es usado por miles de usuarios en todo el mundo en sus

computadoras personales, teléfonos o organizadores personales.

Aunque prácticamente usar el software libre es muy similar a usar cualquier otro software, el software libre es un legado que le pertenece a toda la humanidad y como movimiento es interesante entender como un grupo de programadores, voluntarios y activistas ha creado esta plataforma.

El software libre es propiedad de todos: cada persona en el mundo tiene derecho a usar el software, modificarlo y copiarlo de la misma manera que los autores de este mismo. Es un legado de la humanidad que no tiene propietario, de la misma manera que las leyes básicas de la física o las matemáticas. No existe un monopolio y no es necesario pagar peaje por su uso.

Un factor interesante en los últimos años ha sido la adopción del software libre en los gobiernos. Algunos lo han adoptado para ahorrar dinero, otros lo han hecho por cuestiones de seguridad, otros para ayudar a la creación de industrias locales y otros porque el software libre les pertenece.

Este libro es una estupenda introducción para quienes quieren aprender más sobre el software libre y buscan entenderlo: cómo nació este movimiento, quién desarrolla este software, cómo se desarrolla, la importancia de las licencias, los modelos de negocios y los principales grupos de desarrollo.

Son temas importantes que ayudarán al lector a rápidamente entender cómo funcionan las distintas piezas. Jordi, un querido amigo, es una autoridad en el tema y ha participado en varios proyectos libres como AbiWord, Softcatalà y el proyecto Mono.

Miguel de Icaza, Enero de 2005

## **PREFACIO**

Este libro es fruto de mi experiencia profesional y de mi contribución al voluntariado tecnológico en los últimos años que ha estado focalizada en el desarrollo y difusión del software libre. La posibilidad de trabajar en Ximian – ahora parte de Novell - y poder desarrollar software con algunas de las personas más brillantes de la comunidad de software libre ha sido una experiencia muy enriquecedora. Así como los numerosos viajes a Estados Unidos y conversaciones informales con mis compañeros de trabajo en Boston, Utah, Barcelona, y por chat.

La oportunidad que surgió con la Universitat Oberta de Catalunya en el año 2002 de crear un máster internacional en Software Libre – el primero a nivel mundial – fue sin duda una oportunidad única. La experiencia de poder coordinar, revisar, y escribir parte, de más de catorce temarios sobre software libre en registro académico y trabajar con más de 25 autores y 20 consultores ha sido determinante en la

consolidación de mi visión del software libre y sus repercusiones.

Muchas de las ideas y conceptos desarrollados en este libro son fruto de las innumerables conversaciones informales con Óscar del Pozo, Jesús Corrius, Xavi Caballé, Juan Tomas García, Javier Canderia, Lluís Sánchez, Jordi Murgó, Roberto Santos, Marcelo D'Elia Branco, Miquel Piulats, y en general, toda esa gente que apasionada por la tecnología nos hemos ido encontrando en los últimos años. También a la gente que trabajamos desde voluntariado alrededor de la comunidad Softcatalà.

Agradezco sinceramente a Carmen Jané su esfuerzo en la revisión de las numerosas versiones de este libro, y conseguir finalmente que la versión final tuviera la soltura y legibilidad que nunca podría haber alcanzado por mí mismo. A Malcom Bain, Xavi Caballé, David Megías, Jesús Corrius, Òscar del Pozo, Daniel Luque Font, Javier Candeira, Quim Gil y Carles Mateo por su revisión del original y por sus aportaciones, valiosísimas en todos los casos y sin las cuales este libro no sería lo que es hoy.



Finalmente, un agradecimiento especial a todos los *hackers* que día a día escriben y mejoran software, porque sin ellos el software libre no sería una realidad.

Jordi Mas i Hernández [jordi@ximian.com](mailto:jordi@ximian.com)

Barcelona, Enero de 2005

# ÍNDICE

Prólogo

Prefacio

## 1. Visión histórica del software libre

1.1 La importancia del software

1.2 Ética *hacker*

1.3 La cultura Unix

1.4 GNU

1.5 Libertad y coste

1.6 Linux

1.7 *Open Source*

1.8 El papel del software libre en el crecimiento de Internet

1.9 Formatos abiertos

1.10 Situación de monopolio

## 2. Ventajas y mitos del software libre

2.1 Ventajas del software libre

2.2 Coste

2.3 Innovación tecnológica

2.4 Requisitos de hardware y durabilidad de las soluciones

- 2.5 Escrutinio público
- 2.6 Independencia del proveedor
- 2.7 Industria local
- 2.8 Datos personales, privacidad y seguridad
- 2.9 Adaptación del software
- 2.10 Lengua
- 2.11 FUD (miedo, incertidumbre y duda)

### 3. Marco jurídico

- 3.1 Introducción
- 3.2 Los derechos de autor
- 3.3 Licencias de software
- 3.4 Las patentes de software
- 3.5 Marcas
- 3.6 Secreto comercial
- 3.7 Garantías e indemnizaciones
- 3.8 Protección contra violaciones de derechos
- 3.9 Gestión digital de contenidos
- 3.10 *Creative Commons*: contenidos digitales libres

### 4. La producción del software libre

- 4.1 Motivos que impulsan al desarrollo de software libre
- 4.2 Perfil de los desarrolladores de software libre
- 4.3 Herramientas colaborativas

- 4.4 Organización de proyectos libres
- 4.5 Toma de decisiones
- 4.6 Ejemplo: la comunidad Abiword

## 5. Proyectos libres

- 5.1 Linux
- 5.2 Apache
- 5.3 KDE
- 5.4 GNOME
- 5.5 Mozilla
- 5.6 OpenOffice.org
- 5.7 Mono

## 6. Oportunidades de negocio en el software libre

- 6.1 Introducción
- 6.2 La industria del software
- 6.3 Modelos de negocio
- 6.4 Experiencias fallidas
- 6.5 Experiencias de éxito

## 7. Software libre en el sector público

- 7.1 La importancia del software libre en la administración
- 7.2 Iniciativas legales
- 7.3 Caso Universidad de Lleida
- 7.4 Caso de Extremadura
- 7.5 Caso de Brasil

## 8. A modo de conclusión

### Apéndices

Webs relacionadas con el software libre

Declaración de Barcelona para el avance del software libre

Medida de gobierno para el impulso del software libre en el Ayuntamiento de Barcelona

### Bibliografía

# 1. VISIÓN HISTÓRICA DEL SOFTWARE LIBRE

## 1.1 La importancia del software

La transición que estamos viviendo actualmente hacia una sociedad del conocimiento ha modificado profundamente las relaciones entre las personas, empresas, y gobiernos: las empresas usan la Red para relacionarse con clientes y proveedores, utilizan también herramientas de gestión del conocimiento para ser más eficientes, los gobiernos mejoran su presencia en Internet y los servicios a los ciudadanos a través de la Red, los usuarios usan las herramientas para mejorar sus relaciones personales, etc. Vamos de forma imparable hacia una sociedad altamente interconectada donde el eje fundamental es la información.

El software es cada vez más el gran intermediario entre la información y la inteligencia humana. De la misma manera que nos preocupa la libertad para poder acceder a la información y, si existe censura, nos debe preocupar con igual intensidad quién controla este

intermediario y qué garantías tenemos de su transparencia y fiabilidad.

En principio, el software es un programa o un conjunto de programas informáticos que tienen una tarea determinada. Es el procesador de textos que usamos, el controlador de grabación de nuestros espacios televisivos favoritos, o las aplicaciones que permiten operar un teléfono móvil.

El software está compuesto por un conjunto de instrucciones que un ordenador ejecuta para poder realizar una función específica. Normalmente los programadores escriben en un lenguaje que los humanos podemos entender y que posteriormente es traducido a secuencias de unos y ceros que es el único lenguaje que las máquinas entienden. El conjunto de órdenes en el lenguaje que los humanos trabajamos se llama código fuente.

Sin acceso al código sólo podemos usar el programa, no podemos ver cómo está hecho o introducir mejoras. Un símil muy utilizado es el de la receta de cocina, en que el código fuente sería las instrucciones que permite confeccionar un plato. Sin la receta sólo se puede degustar el plato, pero no sabemos si, al añadirle algo, vamos en contra de alguno de sus

ingredientes, ya que se desconoce su composición y la proporción. En este sentido, el código fuente juega un papel fundamental en cómo entendemos las libertades del software.

Para entender la importancia del software se podrían poner varios ejemplos. A finales de los 90 pudimos ver en todo el mundo la preocupación por parte de empresas y gobiernos por las repercusiones que podía tener el llamado efecto 2000. El ya famoso error informático era causado por el hecho de que muchos programas almacenaban la parte de la fecha correspondiente al año usando únicamente dos dígitos, de tal forma, que después del año 99 (el 1999) podíamos pasar al año 00 (¿año 2000 o año 1900?) causando todo tipo de errores en el cálculo de períodos de tiempo. Los ordenadores de empresas eléctricas, centrales nucleares, sistemas de control de aviación, bancos y, en general, todo el software de uso cotidiano, tuvieron que ser revisados. Finalmente, algunas aplicaciones fueron corregidas, otras ya funcionaban correctamente y no hubo que lamentar ninguna catástrofe, pero hubo miles de predicciones apocalípticas sobre las consecuencias que podía llegar a tener este error. Y así podría haber sido si no se hubiera reparado a tiempo.



Cuando los ingenieros de software nos hallamos ante un programa que no da acceso al código fuente –es decir, que no es libre – nos encontramos que no lo podemos entender, y por tanto que no lo podríamos arreglar aunque hubiéramos descubierto un error y conociéramos su solución. Es decir, aunque como profesionales tengamos el remedio, nos vemos incapacitados para aplicarlo.

El software tiene un papel muy destacado en la sociedad y es importante garantizar métodos transparentes en sus diferentes fases de producción y explotación. El software libre, al dar acceso al código, es el único que puede garantizar esta transparencia.

## **1.2 Ética *hacker***

Durante los años 60, los sistemas informáticos medianos se popularizaron en algunas universidades. Eran todavía muy caros y estaban básicamente destinados a un uso científico. Uno de los centros de investigación más importantes de aquella época era el *Massachusetts Institute of Technology*, popularmente conocido por sus siglas MIT<sup>1</sup>, donde se realizaban destacados proyectos de investigación.

En aquellos tiempos, los sistemas informáticos eran en modo de texto y las interfaces gráficas todavía no existían. De hecho, no existían ni siquiera los estudios de informática como disciplina. Alrededor de los centros de cálculo se fueron formando grupos de expertos que eran capaces de llevar estas máquinas hasta límites insospechados y que eran, por encima de todo, entusiastas de la tecnología. Estas personas se denominaron *hackers*, en el sentido que eran capaces de encontrar soluciones ingeniosas a problemas complejos. Desgraciadamente, años más tarde, esta palabra se empezó a utilizar para nombrar a los intrusos informáticos que robaban datos o programadores que escribían virus, a quienes realmente se debe llamar *crackers*.

Durante los años 60 y parte de los 70 era habitual distribuir las aplicaciones junto con su código fuente. De esta manera todo el mundo podía participar en el proceso de mejora del software, el intercambio de trucos y compartir un espíritu muy extendido de ayuda entre usuarios que queda reflejado en los mensajes que circulaban en aquella época en redes como UUNET o Fidonet.

Los *hackers* tienen su propio código de conducta. Para ellos es fundamental garantizar el acceso a la

información y los recursos informáticos a todo el mundo. Richard Stallman<sup>2</sup>, por ejemplo, se sintió especialmente ofendido<sup>3</sup> cuando en su laboratorio de investigación en el MIT comenzaron a instaurar contraseñas para acceder a los sistemas informáticos. Stallman propuso a los usuarios del sistema que utilizaran la tecla retorno como contraseña que era el equivalente a no utilizarla. Finalmente las contraseñas fueran instauradas y compartir la información entre usuarios ya no fue posible como lo había sido hasta entonces en aquel laboratorio.

Para los *hackers* los retos intelectuales son importantes como también lo es ser capaces de compartir sus resultados con el resto de la comunidad.

Este modelo, donde prima la compartición de información y el trabajo cooperativo, es bastante similar al que tradicionalmente se ha usado en el mundo académico y científico. En estos ámbitos, los resultados de las investigaciones se publican y se divulgan en publicaciones científicas, y sirven de base para nuevas investigaciones. Éste es principalmente el modelo sobre el que la humanidad ha innovado y avanzado.

Los *hackers*, aún hoy en día, son los principales

impulsores y desarrolladores de software libre y su ética se ha convertido en el código ético implícito sobre el cual se ha forjado el movimiento.

### **1.3 La cultura Unix**

El sistema operativo es el programa que interacciona entre el hardware, el usuario y las aplicaciones. Sin sistema operativo, un ordenador es sólo un amasijo de cables y hierro. Unix ha sido el sistema operativo que más influencia ha tenido en la historia de la microinformática y constituye los fundamentos técnicos y, en parte filosóficos, sobre los cuales se desarrolló el software libre a principios de la década de los 80. Actualmente existen en el mercado desde organizadores personales hasta supercomputadores que usan sistemas basados en Unix. Linux es un familiar cercano de los sistemas Unix.

A finales de los años 60 la empresa AT&T se enfrentaba a los crecientes problemas de interoperabilidad que representaba tener diferentes ordenadores de diferentes fabricantes con distintos sistemas operativos. Para solucionarlo crearon el proyecto Multics (*Multiplexed Information and Computing Service*) que fracasó por su extremada complejidad.

En 1969 Ken Thompson y Dennis Ritchie, dos ingenieros que habían trabajado en Multics en los laboratorios Bell, crearon la primera versión del sistema operativo Unix. Entre las características que lo hacían un verdadero avance en su tiempo destacan la posibilidad de utilizarlo en diferentes plataformas de hardware y su capacidad multiusuario y multiproceso. La simplicidad y elegancia en su diseño hacen que, más de treinta años después, aún sea un sistema de amplísimo uso, aunque ha sufrido muchas mejoras desde entonces.

Durante los siguientes años, Unix fue evolucionando pero su uso siguió limitándose exclusivamente a sistemas ubicados en AT&T. En 1974 Thompson y Ritchie publicaron en el simposium de la *Association for Computing Machinery* - ACM - de sistemas operativos un artículo sobre el sistema Unix que habían diseñado. Este fue un punto de inflexión, ya que empezaron a recibir peticiones de universidades y centros de cálculo para poder obtener una copia en cinta de aquel sistema llamado Unix. Gracias a esto, se fue extendiendo progresivamente.

AT&T no facilitaba soporte de Unix. Era un software que debía ser mantenido por los propios usuarios. Así

se inició la tradición de ayuda entre los propios usuarios que intentaban solucionarse mutuamente problemas fomentando el intercambio de información a través de diferentes foros. AT&T también facilitaba una copia del código fuente de Unix, al principio por unos pocos dólares, lo que permitió que el sistema pudiera ser adaptado en las universidades y centros de computación.

Una de las universidades que fue pionera en la adopción y desarrollo de Unix fue la Universidad de Berkeley<sup>4</sup>. En 1978 liberó la primera versión del Unix BSD (*Berkeley Software Distribution*) que incluía varias herramientas diseñadas en la propia universidad y que mejoraba considerablemente la versión original de Unix. Uno de los *hackers* involucrados en el desarrollo de las primeras versiones de BSD fue Bill Joy, que años más tarde crearía Sun Microsystems, donde usarían Unix como sistema operativo para toda la gama de su hardware.

Con el tiempo se formalizaron dos grandes ramas de Unix, una producida en los laboratorios Bell de AT&T y la otra producida en Berkeley bajo el nombre de BSD que estaba basada en código fuente que AT&T que les proporcionó años atrás. Ambas versiones

requerían un pago para obtener una copia del código fuente, aunque la BSD era mucho más asequible.

En 1979 AT&T liberó la versión 7 de su versión de Unix y empezó a poner grandes trabas en la distribución del código fuente. Desde Berkeley siguieron innovando, desarrollando mejoras como la implementación de *sockets* y protocolo TCP/IP, sobre el cual se basa la infraestructura de Internet.

A principios de los años 80 proliferaron los fabricantes de sistemas como IBM o Hewlett-Packard que facilitaban con su hardware versiones personalizadas del sistema operativo Unix basadas en la versión de AT&T. Algunos otros, como Sun Microsystems o Cray Inc., se basaron en la versión de Berkeley. Esto creó un periodo de unos años donde estas versiones de Unix eran incompatibles entre sí pero que luego se solucionaría al definirse y acordarse la implementación de estándares.

AT&T fue forjando una actitud de fabricante de software propietario y llegó incluso a querellarse contra los autores de BSD Unix. Estas actitudes crearon muchos recelos hacía AT&T entre la comunidad científica y universitaria.

## 1.4 GNU

Con la explosión de la microinformática, el descenso de precio de los sistemas informáticos y su popularización entre las empresas, aparecieron las primeras compañías de software. Muchas de ellas empezaron contratando *hackers* que estaban alrededor de los centros de cálculo de las universidades, de forma que éstas se fueron despoblando de aquellos pioneros. Pero, además, muchas de estas empresas creyeron que si denegaban el acceso a los usuarios y a otros desarrolladores al código fuente de las aplicaciones que mejoraban o desarrollaban, podrían realmente conseguir una ventaja competitiva. Éste fue un punto de inflexión importante, ya que se rompió con la tradición de compartir el código que había imperado hasta entonces y consecuentemente con la propia ética *hacker*.

Poco a poco se fue extendiendo un modelo de código cerrado en el cual el software se vendía sin el código fuente y, cada vez más, las libertades de los usuarios se fueron acortando. Esta fue la época en que aparecieron técnicas como, por ejemplo, las bombas de tiempo que limitaban el periodo durante el cual un usuario podía utilizar un producto. Los programas *shareware* popularizarían más tarde estas bombas de



tiempo como sistema para obligar a los usuarios a adquirir una licencia.

Una de las personas que había vivido de cerca toda aquella evolución era Richard Stallman, quien fue pionero en defender las libertades que se habían perdido y acuñó el término 'software libre'. El 27 de septiembre de 1983 Richard Stallman muy preocupado por esta pérdida de libertades anunciaba en el foro Usenet *net-unix.wizards*, que empezaba a trabajar sobre una implementación libre de un sistema inspirado en Unix que denominaría GNU y que estaría libre de código de AT&T, es decir, una implementación desde cero sin posibles problemas legales con AT&T. En el mensaje a Usenet, Stallman explicaba detalladamente su experiencia como desarrollador de sistemas y pedía la ayuda de todo el mundo que quisiera ofrecer parte de su tiempo, dinero, o hardware.

El 1984, Stallman creó la *Free Software Foundation* con el objetivo de crear el sistema Unix libre GNU y la potenciación del software libre. La definición de software libre propuesta<sup>6</sup> por la *Free Software Foundation*, se basa en cuatro libertades básicas que cualquier programa considerado libre debe proporcionar:

- 0: libertad para utilizar el programa para cualquier propósito.
- 1: libertad para poder estudiar cómo funciona el programa. Implica acceso al código fuente del mismo.
- 2: libertad para redistribuir el programa.
- 3: libertad para hacer modificaciones y distribuir las mejoras. Implica también acceso al código fuente del mismo.

El software libre se basa en la cooperación y la transparencia y garantiza una serie de libertades a los usuarios. Estos aspectos, junto al hecho de que su desarrollo ha sido paralelo al de Internet, han causado que sea abanderado para un gran número de usuarios que tienen una concepción libertaria del uso de las nuevas tecnologías. Los programas que no son libres se les llaman propietarios o privativos. Por ejemplo, todas las versiones de Microsoft Windows o Adobe Acrobat son ejemplos de software propietario.

## 1.5 Libertad y coste

Es habitual que los usuarios confundan el software libre con el software gratuito. Es importante distinguir entre las libertades que nos proporciona un software y el coste del mismo. Un programa, por el simple hecho de ser gratuito, no es ni mucho menos libre. Por ejemplo, Internet Explorer de Microsoft es un programa gratuito pero no es libre, ya que no da a sus usuarios la posibilidad de estudiarlo (incluyendo el acceso a su código fuente), ni de mejorarlo, ni de hacer públicas estas mejoras con el código fuente correspondiente, de manera que todo el mundo se pueda beneficiar. Internet Explorer es un programa propietario – en cuanto a las libertades – y gratuito – en cuanto al coste –.

Existe una distinción fundamental entre los programas que garantizan los derechos de distribución y modificación, el software libre, y los que no los garantizan que consideramos propietarios.

Respecto al coste, cualquier software libre se puede vender, siempre y cuando se respeten las libertades originales que lo definen. Por ejemplo, la empresa francesa Mandrake o la norteamericana Novell venden

distribuciones de GNU/Linux, y se trata de software libre porque conserva las libertades que lo definen.

## **1.6 Linux**

Durante los años 80 Stallman continuó trabajando en el desarrollo de las herramientas necesarias para crear un sistema operativo completamente libre. Publicó una versión del editor GNU Emacs y trabajó en herramientas que son fundamentales para el movimiento del software libre, como, por ejemplo, el compilador GCC o el depurador GDB.

Ya en sus inicios Stallman identificó la necesidad de crear las protecciones jurídicas necesarias para el software libre. En 1989 publicó la versión 1.0 de la licencia GPL<sup>7</sup> - *General Public License* - un proyecto que elaboraba desde el 1985 y que consistía en un contrato entre el autor del software y el usuario que garantizaba la cesión de los derechos que definían al software libre. La licencia GPL era una herramienta legal muy importante dado que Stallman había padecido mucho viendo cómo algunos programadores cogían código que era software libre, hacían modificaciones y no aportaban estas modificaciones a la comunidad.

En 1991 un estudiante finlandés de 21 años llamado Linus Torvalds publicaba en la Red su propio núcleo de sistema operativo inspirado en Unix que más tarde se denominaría Linux. Rápidamente, muchos *hackers* se sumaron a un esfuerzo colectivo para ayudarle a mejorar el núcleo del sistema operativo y creció en importancia. Éste era precisamente el corazón que le faltaba al sistema GNU que Stallman estaba diseñando.

Así, los *hackers* de Linux fueron adaptando las herramientas de GNU al núcleo de Linux y crearon las primeras distribuciones GNU/Linux. Estas distribuciones incluían Linux y todas las herramientas necesarias para que un usuario lo pudiera utilizar. Richard Stallman ha insistido mucho<sup>8</sup> en el hecho que el sistema GNU/Linux existe gracias a las aportaciones de la *Free Software Foundation* y propone usar la denominación GNU/Linux para referirnos a sistemas completos que incorporan un núcleo Linux. La mayoría de usuarios y distribuciones comerciales siguen hablando de Linux para referirse al sistema y al núcleo.

Linux se ha convertido en el buque insignia del movimiento del software libre, ya que es un sistema

operativo completamente libre que incluye todo tipo de herramientas para servidores y usuarios finales.

En 1993 Jordan Hubbard y otros desarrolladores, basándose en el trabajo previo de la versión de Unix de BSD, crearon el proyecto FreeBSD<sup>9</sup> con el objetivo construir una comunidad alrededor de una versión libre del ya histórico Unix BSD. Sin embargo, para evitar problemas legales tuvieron que reescribir una parte importante del sistema y no fue hasta 1995 que pudieron empezar a ofrecer un sistema libre de problemas legales. El sistema operativo libre FreeBSD nunca ha gozado de la popularidad de Linux pero es de amplio uso en Internet y es usado en sitios web como Yahoo!, Hotmail o Apache. También es la base para las nuevas versiones del sistema operativo Mac dentro del proyecto Darwin<sup>10</sup>.

En 1996 se inicia el proyecto KDE<sup>11</sup> con el objetivo de desarrollar un entorno de escritorio moderno y sencillo de usar para Linux. Su decisión más polémica fue basarlo en la librería QT escrita por la empresa TrollTech<sup>12</sup>. Esta librería era entonces gratuita para el uso no comercial pero no era software libre. Esta decisión enojó a la comunidad de software libre ya que para muchos *hackers* el objetivo era hacer un sistema operativo completamente libre, y una parte tan

importante del mismo no podía basarse en software propietario. En 1997 el mexicano Miguel de Icaza lanzó el proyecto GNOME<sup>13</sup> con el objetivo de desarrollar también un escritorio moderno y sencillo de usar pero basado en la librería GTK que era completamente libre. En septiembre del 2000, TrollTech anunció que licenciaba la librería QT bajo la licencia GPL, lo cual la convertía en libre, pero sólo para usos libres. Tanto KDE como GNOME hoy en día son escritorios usados ampliamente en Linux y han alcanzado un grado importante de madurez y solidez.

El año 1998 fue año especialmente importante para el software libre. Netscape anunció la publicación bajo licencia libre de su navegador web Netscape Communicator 4, hasta ese momento un producto comercial, así como la publicación de su código fuente bajo el nombre de Mozilla. Este anuncio sorprendió a muchos analistas, ya que no tenía un antecedente en la industria del software. Fue un momento de inflexión para el software libre, porque una empresa importante en aquel momento como Netscape hacía una apuesta firme por el movimiento y ponía a disposición de la comunidad un navegador de Internet con las últimas tecnologías. El siguiente año Sun Microsystems hacía lo propio con Star Office,

creando el proyecto OpenOffice.org. En los siguientes meses los mayores fabricantes de bases de datos como IBM y Oracle anunciaban que soportarían Linux como plataforma. Lo que supuso un aval ante su uso en el mundo empresarial.

## 1.7 Open Source

Durante el año 1998, Eric S. Raymond, Bruce Perens y otros *hackers* involucrados en el desarrollo de software libre lanzaron la *Open Software Initiative* y propusieron el uso de término *open source* (código abierto) en contraposición al término *free software* (software libre) como término más atractivo al entorno empresarial. El término *free software* en el mundo anglófono creaba una situación incómoda debido a la doble acepción que en inglés tiene el término *free* (que puede significar gratuito o libre). La gran mayoría de empresas en Estados Unidos usan principalmente el término código abierto para evitar dar la percepción que el software libre es un recurso totalmente gratuito y para poner énfasis en valor diferencial que representa el hecho de que el código fuente está disponible.

Bruce Perens, de la *Open Source Initiative* y antiguo coordinador de la distribución de Linux Debian<sup>14</sup>, creó



una lista de condiciones que debe cumplir un programa para ser poder ser considerado *Open Source*. Estas condiciones son muy similares y, de hecho están basadas, en las directrices de software libre de Debian<sup>15</sup>. Estas condiciones también son aplicables a cualquier programa que sea software libre y pueden ayudarnos a matizar sus implicaciones.

1. Libre distribución. No se puede impedir la venta o distribución del programa o parte de él. Así mismo, tampoco se puede exigir el pago de un canon o tasa a cambio de su distribución por parte de terceros.

2. Código fuente. El programa debe incluir su código fuente y no se puede restringir su redistribución.

3. Trabajos derivados. No debe impedirse realizar modificaciones o trabajos derivados del programa y debe permitirse que éstos sean distribuidos bajo mismos términos del software original.

4. Integridad del código de fuente original. Puede exigirse que una versión modificada del programa tenga un nombre y número de versión diferente que el programa original para poder proteger al autor original de la responsabilidad de estas versiones.

5. No discriminación contra personas o grupos. Las condiciones de uso del programa no pueden discriminar contra una persona o un grupo de personas.

6. No discriminación contra usos. No se puede negar a ninguna persona hacer uso del programa para ningún fin como, por ejemplo, comercial o militar.

7. Distribución de la licencia. Los derechos del programa deben aplicarse a todos quienes se redistribuyen el programa sin ninguna condición adicional.

8. La licencia no debe ser específica de un producto. Los derechos garantizados al usuario del programa no deben depender de que el programa forme parte de una distribución o paquete particular de software.

9. La licencia no debe restringir otro software. La licencia no debe poner restricciones en otros programas que se distribuyen junto con el software licenciado.

10. La licencia debe ser tecnológicamente neutra. No

puede existir ninguna disposición de la licencia que obligue al uso de una tecnología concreta.

## **1.8 El papel del software libre en el crecimiento de Internet**

El software libre ha tenido un papel fundamental en el crecimiento y extensión de la Red. Si usted ha utilizado Internet, ya es usuario de software libre porque la mayor parte de la infraestructura de Internet se basa en protocolos abiertos. Aproximadamente el 67% de servidores web emplean Apache<sup>16</sup>, otro gran número usan SendMail para gestionar el envío de correo electrónico y prácticamente la totalidad de los servidores de nombres (DNS), esenciales en el funcionamiento de la Red, utilizan el programa BIND o derivados de su código fuente.

Tim Berners-Lee, creador de la web, describe<sup>17</sup> cómo en el año 1993 decidió presionar al CERN, donde trabajaba, para que cediera bajo la licencia GPL, creada por la *Free Software Foundation*, todo el software que había creado para realizar la infraestructura de la web hasta entonces. Finalmente, en abril de ese mismo año el CERN accedió a permitir

a cualquiera el uso de su software y protocolos libremente.

Es indiscutible la importancia que ha tenido el software libre en la extensión y el desarrollo de Internet desde sus inicios. Sin la existencia del software libre Internet hoy en día probablemente no existiría. Ha sido igualmente importante el hecho de que los protocolos que definen la arquitectura de Internet sean abiertos y que no hayan sido controlados por una o varias empresas<sup>18</sup>.

## **1.9 Formatos abiertos**

A menudo nos encontramos con creadores de sitios que sólo prueban sus páginas con Microsoft Internet Explorer, muchas veces por desconocimiento de la existencia y del grado de implantación de otros navegadores o, en ocasiones, simplemente porque no consideran la compatibilidad como un tema importante.

Internet ha creado sus propios mecanismos para evitar este tipo de situaciones. En 1994 se creó el *World Wide Consortium* (W3)<sup>19</sup>, que agrupa a los principales fabricantes de software de Internet, con la misión principal de definir y promover la creación de

estándares para la Web. En realidad, cuando hablamos de estándares web nos referimos casi siempre a las definiciones y recomendaciones de este consorcio, que ha conseguido que prácticamente todos los navegadores modernos funcionen en un grado aceptable con los estándares más recientes.

Esto que hemos visto que sucede en la Web, ocurre también en un ámbito muy importante: en el intercambio de datos entre aplicaciones. Normalmente se definen formatos de intercambio que permiten operar entre ellas. Pese a que lo deseable sería que estos formatos, contenedores de nuestra información, fueran conocidos por todos y que no tuvieran restricciones de uso, en ocasiones pueden tener limitaciones de uso o simplemente no estar documentados.

Los formatos libres garantizan la libertad de los usuarios para intercambiar información con todo el mundo independientemente de la aplicación que utilicen, ya que permiten a cualquier programador desarrollar software que trabaje con estos formatos. Si, por ejemplo, utilizamos Microsoft PowerPoint para enviar una presentación, sólo tienen garantizado el acceso con todas las particularidades del documento los usuarios de este programa.

La restricción en el acceso de la información a un determinado navegador o formato representa una discriminación contra los usuarios de los otros navegadores o aplicaciones y es un hecho que afecta y preocupa especialmente al mundo del software libre. En el año 2003 se aprobó un real decreto<sup>20</sup> en el Boletín Oficial del Estado que regula los registros y las notificaciones telemáticas donde se establece el cumplimiento de los estándares web.

### **1.10 Situación de monopolio**

Actualmente, en la informática existe una situación de monopolio en dos de las áreas más importantes para los usuarios: sistemas operativos y aplicaciones ofimáticas. Cada ordenador del mundo necesita un sistema operativo para poder funcionar. Existen incluso personas que piensan que el sistema operativo es algo tan básico y fundamental en la Sociedad de la Información que debería ser declarado un bien público.

Microsoft, con las diversas versiones del sistema operativo Windows, goza de una base instalada de más del un 90% del total de los sistemas operativos

del mundo. Esta situación le sitúa en una posición de control del mercado muy por delante de sus competidores. Al ser productor del sistema operativo Windows y de aplicaciones que se ejecutan sobre él – como Microsoft Office -La multinacional conoce mejor que nadie cómo sacar provecho del sistema. Conoce detalles de su diseño a los que nadie más tiene acceso al ser un programa propietario. Además, sabe con mucha más antelación qué nuevas funcionalidades tendrán futuras versiones del sistema y cómo explotarlas. Durante los juicios antimonopolio que Microsoft tuvo en Estados Unidos se propuso que la compañía se dividiera en dos empresas<sup>21</sup>: una que diseñará los sistemas operativos y otra las aplicaciones.

Microsoft no sólo ha utilizado esta ventaja evidente, sino que además en multitud de ocasiones ha tenido prácticas éticamente cuestionables. Sirva a modo de ejemplo el caso de DR-DOS. A principios de los 90 Microsoft vendía MS-DOS como principal sistema operativo y empezaba también a vender Windows 3.1 que se ejecutaba encima de MS-DOS. En aquel entonces, la empresa Digital Research comercializaba DR-DOS, un sistema alternativo y compatible MS-DOS mucho más barato y técnicamente superior. La alternativa de Digital Research fue ganando

aceptación y empezó a ser una amenaza para las ventas de MS-DOS. Microsoft decidió añadir deliberadamente unas instrucciones en el programa Windows 3.1 para que sólo se ejecutará correctamente<sup>22</sup> si se ejecutaba sobre MS-DOS y diera un mensaje de error si se ejecutaba en DR-DOS. Digital Research demandó a Microsoft. Es simplemente uno de los muchos ejemplos donde Microsoft ha usado su poder sobre la plataforma Windows para dañar a la competencia.

Otra tendencia de Microsoft es utilizar el sistema operativo como plataforma para integrar aplicaciones e imponer su tecnología. Así, Microsoft decidió integrar Internet Explorer en Windows dejando a Netscape fuera del mercado o integrar Windows Media Player dejando Real Audio fuera del sistema. Ambas empresas llevaron a juicio a Microsoft.

Después del sistema operativo, la segunda aplicación más usada por los usuarios es el paquete ofimático, es decir, un procesador de textos, una hoja de cálculo y las herramientas asociadas. Aquí de nuevo Microsoft con su producto Microsoft Office goza de una cuota de mercado similar a la de Windows, es decir, también de monopolio.



Microsoft tiene diversos juicios en Estados Unidos y Europa por prácticas monopolísticas. Gobiernos, empresas, y usuarios de todo el mundo dependen de forma exclusiva de este software propietario.

Muchas personas ven en el software libre una solución para romper esta situación de monopolio y que el mercado y los usuarios puedan recuperar su libertad.

No es de extrañar que Microsoft y la comunidad del software libre tengan a menudo roces y discrepancias, ya que la visión de la ética en el mundo del software son contrapuestas. Ver con recelo a Microsoft es una tendencia en general de la industria informática.

-----

<sup>1</sup><http://www.mit.edu>

<sup>2</sup><http://www.stallman.org>

<sup>3</sup><http://www.faifzilla.org/ch04.html>

<sup>4</sup><http://www.berkeley.edu/>

<sup>5</sup><http://www.fsf.org>

<sup>6</sup><http://www.fsf.org/philosophy/free-sw.html>

<sup>7</sup><http://www.gnu.org/copyleft/gpl.html>

<sup>8</sup><http://www.gnu.org/gnu/linux-and-gnu.html>

<sup>9</sup><http://www.freebsd.org>

<sup>10</sup><http://developer.apple.com/darwin/>

<sup>11</sup><http://www.kde.org>

<sup>12</sup><http://www.trolltech.com/>

<sup>13</sup><http://www.gnome.org>

<sup>14</sup><http://www.debian.org>

<sup>15</sup>[http://www.debian.org/social\\_contract.html](http://www.debian.org/social_contract.html)

<sup>16</sup>[http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)

<sup>17</sup>Ver libro Weaving the web, The past, Present and Future of the  
World Wide Web

<sup>18</sup><http://www.ietf.org/>

<sup>19</sup><http://www.w3.org>

<sup>20</sup>[http://www.cde.ua.es/boe/frame.htm?boe20030613\\_22890.gif](http://www.cde.ua.es/boe/frame.htm?boe20030613_22890.gif)

<sup>21</sup><http://www.dwightsilverman.com/ms-busted.htm>

<sup>22</sup><http://www.kickassgear.com/Articles/Microsoft.htm>

## **2. VENTAJAS Y MITOS DEL SOFTWARE LIBRE**

El software libre conlleva toda una serie de ventajas sobre el software propietario por los derechos que otorga a sus usuarios. Algunas de estas ventajas pueden ser más apreciadas por los usuarios particulares, otras por las empresas, y otras por las administraciones públicas. Desgraciadamente el software libre ha sido objeto de desinformaciones y mitos, algunos provocados deliberadamente, que han intentado distorsionar su credibilidad. Vamos a empezar a enumerar las principales ventajas y luego analizaremos algunos de los falsos mitos.

### **2.1 Coste**

Para la mayoría de usuarios individuales el software libre es una opción atractiva por las libertades que garantiza sin necesidad de verse lastrados por el precio. Sin embargo, en el caso de empresas y la Administración Pública, el coste del software es un factor importante y a veces determinante en la elección de nuevos sistemas informáticos.

Cuando se analiza el precio de una solución tecnológica se suele hablar del TCO (*Total Cost of Ownership*), es decir, del coste total de la propiedad que tiene una determinada solución de software. Este concepto fue inventado por el Gartner Group<sup>23</sup> en 1987 como herramienta de análisis exhaustiva de los costes de una solución de mercado y desde entonces se ha convertido en un estándar. En este análisis se reflejan el coste del programa, la ayuda, y el mantenimiento tecnológico de la solución.

Partimos de la base que el software libre no tiene prácticamente coste de licencia y por lo tanto que esta parte del presupuesto se puede invertir para mejores fines como mejorar la adaptación de los programas y la formación en esta tecnología.

Según un estudio de la consultora Robert Frances Group publicado en el año 2002<sup>24</sup> el coste total de propiedad del sistema operativo libre Linux era menos de la mitad que el de Windows. En el estudio se analiza el coste de diferentes servidores durante un período de tres años y se constata que gran parte del ahorro proviene de no tener que pagar licencia y de sus menores costes de administración. En el mismo sentido se expresa un estudio realizado por la consultora ConsultingTimes<sup>25</sup> en este caso sobre el

coste de propiedad de sistemas de correo: también concluye que las soluciones basadas en software libre son mucho más económicas en todos los casos planteados.

Sin embargo, a parte de los menores costes, también deben considerarse otros aspectos positivos del software libre que no quedan reflejados en los análisis TCO como la independencia del proveedor y la posibilidad de una adaptación completa.

Por último destacar que existen bastantes análisis de TCO que se decantan claramente hacia el fabricante que esponsorizó la ejecución del análisis. Es imprescindible ser cauto con este tipo de análisis y buscar segundas fuentes que nos puedan ayudar a contrastar la información facilitada.

## **2.2 Innovación tecnológica**

El modelo del software libre, donde prima el hecho de compartir la información y el trabajo cooperativo, es bastante similar al que tradicionalmente se ha usado en el mundo académico y científico. En estos ámbitos, los resultados de las investigaciones se publican y se divulgan en publicaciones científicas, y sirven de base para nuevas investigaciones. Éste es principalmente el

modelo sobre el que la humanidad ha innovado y avanzado.

En el mundo del software propietario, las licencias de software, las patentes de software y otras herramientas legales y técnicas se utilizan para impedir que terceros participen en ese conocimiento y para que éste continúe siendo patrimonio exclusivo de la empresa que lo creó. La innovación pertenece a una empresa, mientras que en el mundo del software libre, de forma muy similar al dominio público, el conocimiento pertenece a la humanidad.

En el software libre los usuarios tienen un destacado papel dado que influyen decisivamente en la dirección hacia donde evolucionan los programas: votando los errores que quieren que sean corregidos, proponiendo nueva funcionalidad al programa, o contribuyendo ellos mismos en el desarrollo del mismo.

A finales del año 2004 se publicó<sup>26</sup> una lista de las innovaciones más importantes en software del año 2004. Se consideró como innovación número uno el navegador libre FireFox y de los diez programas mencionados también se encontraba OpenOffice.org.

## **2.3 Requisitos de hardware y durabilidad de las soluciones**

Aunque resulta imposible generalizar, sí que existen casos bien documentados donde las soluciones de software libre tienen unos requisitos de hardware menor, y por lo tanto son más baratas de implementar. Por ejemplo, los sistemas Linux que actúan de servidores pueden ser utilizados sin la interfaz gráfica con la consecuente reducción de requisitos de hardware necesarios.

También es importante destacar que en el software propietario el autor puede decidir en un momento dado no continuar el proyecto para una cierta plataforma, para un hardware que considera antiguo, o discontinuar el soporte para una versión de su software. En las aplicaciones de software libre, estas decisiones no pueden ser tomadas por una empresa o individuo sino por toda una comunidad, con diferentes intereses. Esto se traduce en un mejor soporte en general para las versiones antiguas de software y de plataformas de hardware o software más minoritarias.

## 2.4 Escrutinio público

El modelo de desarrollo de software libre sigue un método por el que el software se escribe de forma cooperativa por programadores, en gran parte voluntarios, que trabajan coordinadamente en Internet. Lógicamente, el código fuente del programa está a la vista de todo el mundo, y son frecuentes los casos en que se reportan errores que alguien ha descubierto leyendo o trabajando con ese código.

El proceso de revisión pública al que está sometido el desarrollo del software libre imprime un gran dinamismo al proceso de corrección de errores. Los usuarios del programa de todo el mundo, gracias a que disponen del código fuente de dicho programa, pueden detectar sus posibles errores, corregirlos y contribuir a su desarrollo con sus mejoras. Son comunes los casos en que un error de seguridad en Linux se hace público y con él la solución al mismo. Con el software propietario la solución de los errores no llega hasta que el fabricante del programa puede asignar los recursos necesarios para solventar el problema y publicar la solución.



## 2.6 Independencia del proveedor

Uno de los grandes problemas en la industria del software es la dependencia que se crea entre el fabricante y el cliente. Este hecho se acentúa con especial gravedad cuando el fabricante no entrega el código fuente, ya que inevitablemente el cliente queda atado a él para nuevas versiones y, en general, para cualquier mejora que necesite.

El software libre garantiza una independencia con respecto al proveedor gracias a la disponibilidad del código fuente. Cualquier empresa o profesional, con los conocimientos adecuados, puede seguir ofreciendo desarrollo o servicios para nuestra aplicación. En el mundo del software propietario, sólo el desarrollador de la aplicación puede ofrecer todos los servicios.

A menudo los proveedores de software propietario se ven obligados a dejar de fabricar un producto por un cambio drástico de las condiciones del mercado, o simplemente porque consideran que ya no podrán rentabilizar la inversión. Disponiendo del código fuente, cualquier programador puede continuar su desarrollo y sus actualizaciones hasta que el cliente decida que es el momento adecuado para migrar a un nuevo sistema informático.

## 2.7 Industria local

Según Sedisi (Asociación Española de Empresas de Tecnologías de la Información), en el año 2001 la industria del software en España<sup>27</sup> movió 1.139,84 millones de euros, de los que 315 millones se destinaron a las ventas de sistemas operativos, que en su totalidad están desarrollados en Estados Unidos. Del resto, las herramientas de desarrollo y software de bases de datos representan 126,68 y 156,03 millones, respectivamente, la práctica totalidad también desarrollados en Estados Unidos.

Existe poca industria propia del software estándar en España, y casi todas las aplicaciones de consumo masivo se desarrollan en otros países. Un gran porcentaje de la industria se basa en distribuir y dar apoyo y formación a productos realizados fuera de nuestras fronteras, lo que ubica el desarrollo de los productos, que es realmente la parte de la industria que requiere ingenieros más cualificados y que genera valor y conocimiento, en otros países.

En el ámbito de la Administración Pública, una parte importante de la inversión en software se realiza en licencias de sistemas operativos, servidores y paquetes

de ofimática, que son producidos totalmente en el extranjero y que sólo tienen repercusión económica en España en los márgenes de distribución.

En el software libre no hay coste de licencia debido al derecho a copia y, al disponer del código fuente de la aplicación, es posible desarrollar internamente las mejoras o las modificaciones necesarias, en vez de encargarlas a empresas de otros países que trabajan con sistemas propietarios. De este modo, se contribuye a la formación de profesionales en nuevas tecnologías y al desarrollo local.

Por otro lado, todas las mejoras que se realicen no tienen restricciones y se pueden compartir con cualquier otra administración, empresa, institución u organismo que las necesite. En el software propietario, estas mejoras o no se pueden llevar a cabo o quedan en manos de la empresa creadora, que normalmente se reserva los derechos de uso y propiedad intelectual y establece en qué condiciones las comercializará.

## **2.8 Datos personales, privacidad y seguridad**

En España existe la Ley orgánica de regulación del tratamiento automatizado de los datos de carácter personal (LORTAD), que regula la protección de

datos de las personas individuales<sup>28</sup> Sin embargo, hay aspectos que no están regulados por la ley y que no por ello dejan de ser menos preocupantes.

El software libre, al disponer del código fuente, mejora diversos aspectos relacionados con la perennidad de los datos y su seguridad. Para empezar, los sistemas de almacenamiento y recuperación de la información del software son públicos y cualquier programador puede ver y entender cómo se almacenan los datos en un determinado formato o sistema, lo que garantiza la durabilidad de la información y su posterior migración. Nos sirve de ejemplo de ello el caso de una base de datos de un censo electoral. El software propietario trabaja habitualmente con formatos propios, cuyos mecanismos de almacenamiento no siempre se han hecho públicos, por lo que quizá no sería posible, en caso de que se precisara migrar el sistema, recuperar el contenido de este censo.

El software libre, por su carácter abierto, dificulta la introducción de código malicioso, espía o de control remoto, debido a que el código lo revisan muchos usuarios y desarrolladores que pueden detectar posibles puertas traseras. En el software propietario nunca podremos saber si los programadores originales

introdujeron a título personal, o por encargo de la empresa, puertas traseras que ponen en peligro la seguridad del sistema o la privacidad de los datos.

Algunos fabricantes de software propietario han colaborado con agencias gubernamentales<sup>29</sup> para incluir accesos secretos al software para así poder visualizar datos confidenciales; de este modo, se comprometen aspectos de la seguridad nacional cuando estos sistemas se utilizan para almacenar datos críticos de gobiernos. En el mundo del software libre, cualquier programador puede realizar una auditoría para comprobar que no se ha introducido ningún código malicioso, y, a su vez, cualquier entidad puede añadir libremente encriptación adicional a la aplicación que utilice para proteger sus datos.

La empresa Mitre ha elaborado un estudio<sup>30</sup> por encargo del Departamento de Defensa de los Estados Unidos donde se analiza el uso de software libre y de código abierto en sistemas que se encuentran en producción en este departamento. Las conclusiones son claramente favorables a seguir incrementando el uso del software libre, y se destaca la posibilidad que ofrece de solucionar errores de seguridad de forma inmediata sin depender de un proveedor externo. Según este informe, hay más de 115 aplicaciones de

software libre en uso en el Departamento de Defensa, con más de 250 ejemplos de su empleo en diferentes entornos.

## **2.9 Adaptación del software**

El software propietario habitualmente se vende en forma de paquete estándar, que muchas veces no se adapta a las necesidades específicas de empresas y administraciones. Una gran parte de la industria del software se basa en desarrollar proyectos donde se requiere software personalizado. El software libre permite personalizar, gracias al hecho de que disponemos del código fuente, los programas tanto como sea necesario hasta que cubran exactamente nuestra necesidad. La personalización es un área muy importante en que el software libre puede responder mucho mejor que el software de propiedad a unos costes mucho más razonables.

Aproximadamente el 75%<sup>31</sup> por ciento del software que se escribe en el mundo es software de uso interno para empresas, que requiere un alto grado de personalización y donde el software puede proporcionar desarrollos más económicos.

## 2.10 Lengua

Las lenguas minoritarias como el catalán, el euskera, el gallego o el occitano han tenido pocas posibilidades de desarrollarse en el mundo del software propietario. Ello es debido a que, por su licencia, las traducciones sólo las puede realizar o autorizar el fabricante del software. Además, en el pasado, tanto en el caso del catalán como en el del euskera las traducciones de sistemas operativos y paquetes de ofimática de Microsoft se han hecho a precios desorbitados<sup>32</sup> y que en poco tiempo han quedado obsoletos.

En el mundo del software libre estas lenguas, y muchas otras, gozan de una buena salud. Los principales proyectos libres, como OpenOffice.org, KDE, Mozilla o GNOME<sup>3</sup>, tienen numerosas traducciones, así como muchas otras aplicaciones, gracias a que no precisan autorización de ningún propietario y cualquier persona o institución puede realizarlas. En cambio, en el software propietario sólo la empresa productora posee los derechos para realizar la traducción. Además, si el programa que traducimos no dispone de corrector ortográfico en nuestra lengua podemos desarrollar nuestro propio corrector o adaptar alguno de los existentes en el mundo del software libre<sup>34</sup>.

Por último, cabe destacar que cada vez que se crea un nuevo recurso lingüístico en el ámbito del software libre (una traducción, un diccionario, un glosario, etc.), éste, al quedar a la disposición de todo el mundo, puede ser reutilizado en futuras aplicaciones. En el mundo del software propietario, cada traducción y recurso lingüístico creado está ligado al fabricante y a sus restricciones de uso.

### **2.11 FUD (miedo, incertidumbre y duda)**

FUD es un acrónimo inglés que corresponde a *Fear, Uncertainty and Doubt* (miedo, incertidumbre y duda) . El FUD es una estrategia que consiste en el confundir del público dando información incompleta, una visión distorsionada de la realidad, y dejando el resto a la imaginación del receptor del mensaje con el objetivo de crear miedo, incertidumbre y duda en el momento de tener que evaluar soluciones de la competencia. Los objetivos principales del FUD son los usuarios, analistas de mercado, y responsables de informática.

Microsoft y otros fabricantes de la industria han utilizado estas técnicas ampliamente. Por ejemplo, la empresa dirigida por Bill Gates intentó hacer creer a la



industria que se debían usar sus productos Windows NT en el cliente y servidor juntos creando incertidumbre, miedo, y duda sobre el hecho de si NT sería capaz de funcionar con otros sistemas que no fueran los suyos.

El software libre ha sido objetivo de ataques de este tipo desde distintos frentes teniendo como efecto el retraso en algunos casos en adopción de soluciones basadas en software libre.

En octubre de 1998 se filtraron, con pocos días de diferencia, varios documentos internos de Microsoft que describían y analizaban la amenaza que representaba el software libre para la compañía y recomendaba algunas estrategias en la línea FUD. Microsoft a los pocos días confirmó la autenticidad<sup>35</sup> e los documentos y reconoció que habían estado escritos por algunos de sus empleados pero dijo que no representaban la posición oficial. Estos documentos se conocen con el nombre de documentos de Halloween ya que coincidió con esta festividad el día que el primer documento se filtró.

Los documentos de Halloween reconocen ya en el año 1998, que el software libre había alcanzado niveles de credibilidad importantes, que tiene una calidad similar

o superior al software propietario, y que representa una amenaza para las ventas de Microsoft. Los documentos además constatan que según la opinión de Microsoft las técnicas FUD no han tenido gran efecto sobre el software libre y que la mejor estrategia es extender los estándares actuales de forma que sólo Microsoft pueda proporcionar el software que los soporte correctamente<sup>36</sup>. Popularmente se conoce esta técnica como contaminación de estándares.

Para combatir el FUD y sus efectos negativos la mejor estrategia es dar la máxima información sobre el software libre. A continuación vamos a intentar enumerar algunos de los mitos y errores más comunes sobre el software libre y dar la visión que se tiene desde el software libre de estos hechos.

### **Sólo existe software libre para Linux**

Linux es uno de los barcos insignia del movimiento del software libre pero la realidad es que hay muchos más programas libres, como por ejemplo OpenOffice.org, Mozilla, Abiword, GIMP o muchos otros, que se encuentran disponibles para multitud de sistemas, incluyendo Microsoft Windows y Mac OS. El concepto de software libre no está atado a ningún sistema, sino que, de hecho, también es aplicable a la

documentación y a toda clase de creaciones artísticas digitales.

### **No hay soporte para las empresas**

Hoy en día, empresas como por ejemplo IBM, Sun, Compaq, Dell, RedHat, Suse, Mandrake, Novell y muchas otras apoyan y dan servicios profesionales de sistemas basados en software libre a usuarios, empresas e instituciones. Por otra parte, el apoyo técnico de empresas de software propietario se acostumbra a adquirir como un servicio adicional.

### **Nadie escribe software de forma gratuita**

Las motivaciones de los programadores de software libre son muy diversas. Hay cientos de proyectos impulsados solamente por voluntarios, otros proyectos vinculados a universidades, otros a empresas, y muchos que son combinaciones de los anteriores. Cada día se escriben líneas de código nuevo y existen más proyectos libres como así lo demuestran estadísticas como las de SourceForge<sup>37</sup>.

Por otro lado, son comunes los casos donde un cliente contrata un programador para desarrollar una aplicación que luego liberará en licencia libre o que

paga por ampliar uno ya existente. Que el resultado sea libre no está contrapuesto a recibir renumeración por el trabajo.

### **El software libre destruye la industria**

En todas las industrias hay factores disruptivos que ocasionan que algunos fabricantes salgan del mercado y aparezcan nuevos. El software libre no destruye la industria sino que representa un modelo de producción diferente por el cual cada vez más a menudo empresas, usuarios y administraciones tienen interés. Según LibroBlanco.com<sup>38</sup> una iniciativa que hace seguimiento del software libre en el ámbito del sector público, a julio de 2004 había 289 empresas en España contabilizadas que trabajan y dan servicios basados en software libre.

### **El software libre destruye la innovación**

La comunidad del software libre crea constantemente nuevos programas, mejora los existentes, y crea nueva documentación y material de soporte que son patrimonio de la humanidad y que centros de investigación, universidades, usuarios, y empresas usan en todo el mundo.

Con el software propietario, solamente la empresa propietaria del código lo puede mejorar, dejando la innovación en manos exclusivas de esa empresa y sus intereses comerciales. El software propietario crea monopolios, y en una situación así la innovación es mucho menor que la situación de libre mercado que crea el software libre.

### **La disponibilidad del código fuente crea versiones incompatibles**

El mundo del software libre ha aprendido mucho de los Unix propietarios donde esto fue realmente un problema durante los años 80 y 90. En el ejemplo más claro de que esto no está sucediendo en el mundo del software libre es cómo las diferentes distribuciones de Linux son compatibles entre ellas.

### **Las empresas competidoras copiarán tu código**

El software libre está protegido por los derechos de autor. Si una empresa copia parte del código de una aplicación debe hacerlo respetando la licencia en que el código copiado fue liberado. Si esta licencia es del tipo *copyleft*, como por ejemplo la GPL, deberá hacer su código también GPL y por lo tanto deberá convertirse en un proveedor de soluciones libres. No

existe ninguna empresa que desee violar la propiedad intelectual de otra empresa.

## **El software libre es más inseguro que el propietario porque el código esta disponible en la Red**

Uno de los informes que han analizado con más profundidad el tema de la seguridad del software libre en comparación al propietario es el que Ross Anderson presentó<sup>39</sup> en el año 2002 en un congreso en Tolouse, Francia. El informe concluye que el hecho de que el código este disponible a largo plazo no supone una diferencia por si mismo y que este hecho no representa un hecho diferencial.

---

<sup>23</sup><http://www.gartner.com/>

<sup>24</sup><http://www-1.ibm.com/linux/RFG-LinuxTCO-vFINAL-Jul2002.pdf>

<sup>25</sup><http://consultingtimes.com/>

<sup>26</sup>[http://www.vbrad.com/pf.asp?p=source/src\\_top\\_10\\_features\\_2004.  
hm](http://www.vbrad.com/pf.asp?p=source/src_top_10_features_2004.htm)

<sup>27</sup>[http://www.sedisi.es/06\\_index.htm](http://www.sedisi.es/06_index.htm)

<sup>28</sup><http://www.uniondeconsumidores.org/legislacion/lotard.htm>

<sup>29</sup><http://www.cnn.com/TECH/computing/9909/03/windows.nsa.02/>

<sup>30</sup><http://www.egovos.org/pdf/dodfoss.pdf>

<sup>31</sup><http://www.opensource.org/advocacy/jobs.html>

<sup>32</sup>La traducción de Windows 98 al catalán costó (ver <http://www.softcatala.org/windows/>) 80,5 millones de las antiguas pesetas. Por la traducción de Windows 2000, Microsoft pidió originalmente más de 500 millones de pesetas a los Gobiernos vasco y catalán por las respectivas traducciones.

<sup>33</sup><http://110n-status.gnome.org/>

<sup>34</sup><http://fmg-www.cs.ucla.edu/geoff/ispell-dictionaries.html>

<sup>35</sup><http://web.archive.org/web/20010417195837/www.microsoft.com/nstserver/nts/news/mwarv/linuxresp.asp>

<sup>36</sup><http://zdnet.com.com/2100-11-512681.html?legacy=zdn>

<sup>37</sup><http://sourceforge.net/>

<sup>38</sup><http://www.libroblanco.com/>

<sup>39</sup><http://www.cl.cam.ac.uk/ftp/users/rja14/toulouse.pdf>

## 3.MARCO JURÍDICO

### 3.1 Introducción

Hoy en día, cuando la capacidad de generar conocimiento y expresiones intelectuales se ha convertido en el eje central de nuestro entorno económico y social, conocer el marco legal que regula nuestras expresiones intelectuales es casi tan necesario como tener la capacidad para realizarlas. Una parte importante de la innovación que ha introducido el software libre consiste en redefinir cómo gestionamos las herramientas legales disponibles para colectivizar el conocimiento y garantizar más derechos a sus usuarios. Algunas de estas ideas han sido adoptadas más tarde por creadores de contenidos digitales, como música, vídeo o libros.

En el mundo del software, entender el marco jurídico que regula la propiedad intelectual y industrial es fundamental para conocer bajo qué condiciones podemos ceder nuestros programas informáticos o utilizar los de terceros. Si somos usuarios, es importante comprender qué derechos y obligaciones



recibimos al adquirir una determinada aplicación o paquete, y si somos creadores, entender cómo podemos ceder nuestros programas a nuestros usuarios y qué derechos y obligaciones les estamos ofreciendo.

El software es fruto del esfuerzo, valor intelectual e inversión del autor y representa su creatividad. Asimismo, en la gran mayoría de los casos, tiene un uso industrial en el sentido amplio: en las diferentes actividades comerciales, industriales o recreativas del hombre. Por lo tanto, se ha estimado que el software merece una protección legal como cualquier otra creación humana – con la diferencia que es una creación intelectual. Se recurre a la protección ofrecida por el derecho de la propiedad intelectual e industrial que otorga a los autores ciertos derechos monopolísticos de explotación de su obra y control sobre varias actividades relacionadas: la copia, la modificación, la distribución, la comercialización, y hasta su uso en procesos y la producción industriales.

Tradicionalmente en el mundo del software propietario se ha utilizado este marco legal para garantizar que el fabricante del software – que tiene derechos exclusivos sobre el programa informático - pueda bloquear el acceso al código fuente de sus programas, impedir a la competencia desarrollar

productos similares, y limitar los derechos que los usuarios tienen sobre el programa, como copia o modificación. Así mismo con una patente, puede impedir a la competencia desarrollar productos similares. Es, por ejemplo, imposible que nosotros traduzcamos (porque se considera modificación) un programa informático propietario que hemos comprado para nuestro propio uso, y ya totalmente impensable, que podamos distribuir esta traducción entre otros usuarios, incluso aunque hayan adquirido también el programa.

A la inversa, en el mundo del software libre, se utiliza el marco legal de los derechos de autor para colectivizar los derechos que el autor tiene sobre el software, incluyendo los derechos a copia, uso y modificación.

### **3.2 Los derechos de autor**

Los derechos de autor son la herramienta jurídica básica utilizada para la protección de los programas informáticos y creaciones intelectuales. Regulan los derechos reconocidos de forma exclusiva al creador de una obra. Son principalmente los derechos “morales” de reconocimiento del autor y la divulgación de su obra, y los derechos “patrimoniales” de reproducción,

distribución, comunicación, y transformación de la misma. Se obtienen de forma automática, son gratuitos (no requieren ninguna tasa ni registro) y tienen reconocimiento internacional.

Fueron creados sobre todo para proteger los textos literarios, obras artísticas y similares y no se adaptan muy bien al software que tiene un componente “escrito” (por lo cual se protege el software de esta manera) y un componente técnico (los algoritmos, la arquitectura y el código ejecutable). Es importante resaltar que los derechos de autor protegen la expresión literal de una idea (en nuestro caso, el código) pero no las ideas o algoritmos que hay detrás.

Los derechos de autor concedidos de forma exclusiva al autor suelen vencer a los 70 años del fallecimiento del mismo, y en este momento pasan al dominio público donde la obra ya no está atada por derecho exclusivo alguno. Entonces, cualquiera la puede reproducir, distribuir o hacer trabajos derivados sin ningún tipo de restricción. El dominio público ha tenido un papel muy destacado en el avance de la cultura pero no ofrece todas las protecciones jurídicas necesarias que el software libre requiere ya que no permite garantizar que las modificaciones que se hagan a un programa sean también libres.

Cada país ha desarrollado su propia legislación de derechos de autor con sus propios matices, pero desde hace unos años se han ido homogeneizando gracias a varios convenios internacionales. A nivel global, se firmó el convenio de Berna desde ya 1886, ampliado en el año 1971 por la convención de París. Más recientemente, el ADPIC (en el marco de la OMC) en 1994 y los acuerdos de la OMPI de 1996, se aprobaron para adecuar – discutiblemente - este marco legal al mundo globalizado y a la Sociedad de la Información. Gracias a estos acuerdos, cualquier creador de una obra obtiene de forma automática unos derechos de autor sobre la misma, derechos que son válidos y defendibles en casi todos los países del globo. Esto no impide que haya diferencias – sobre todo entre el modelo anglosajón del *copyright* basado en títulos de propiedad y el modelo continental de los derechos de autor que defiende más la persona del autor y sus derechos personales.

### **3.3 Licencias de software**

Debido a que la ley no estipula una fórmula legal específica para la cesión de los derechos de autor a

terceros, el autor de un programa informático y titular de los derechos utiliza habitualmente un contrato. Éste determina cómo el autor cede – normalmente de manera no-exclusiva - parte de sus derechos al usuario (copia, modificación, distribución, etc.), determinando en qué condiciones el usuario puede utilizar el programa informático y detallando el ámbito de los derechos y obligaciones asociados. A este contrato se le denomina licencia de software.

Cuando la licencia no es concretamente negociada entre proveedor y usuario de un programa informático, para asegurar que el usuario tenga conocimiento y acepte las condiciones de la misma, los proveedores le obligan a aceptar la licencia cuando se instala el programa - licencia *click-wrap* - o, a veces incluso en algunos programas propietarios, por el simple hecho de abrir el sobre o paquete que lo contiene - licencia *shrink-wrap* - , procesos ambos cuya validez legal se discute aún hoy en día.

En el mundo del software propietario prácticamente cada fabricante de software ha creado su propia licencia adecuada al software en cuestión y el modelo de negocio del fabricante. En el mundo del software libre, existe también una gran diversidad de licencias. Según las estadísticas de SourceForge<sup>40</sup>, uno de los

mayores repositorios de proyectos libres en Internet, en julio del 2004 un 70% de los proyectos acogidos usaban licencia GPL (*General Public License*), un 11% licencia LGPL (*Lesser General Public License*) y un 7.1% que usaban licencias tipo BSD. Como vemos, es abrumador el uso y extensión de las licencias GPL y LGPL creadas por la *Free Software Foundation*<sup>41</sup>

En términos generales dentro del mundo del software libre, aunque haya muchas licencias distinguimos tres modelos principales de licenciamiento de software. Éstos se diferencian en cómo los propietarios de los derechos ceden parte de los mismos a los usuarios y bajo qué condiciones.

Licencias con *copyleft*. Son aquellas<sup>42</sup> que ceden los derechos de copia, distribución y modificación del programa bajo las condiciones que definen al software libre pero que además exigen que cualquier versión modificada herede el mismo tipo de obligaciones y derechos que tenía el programa original. Estas licencias a menudo se llaman víricas por el efecto de contagio que tienen sobre trabajos derivados. El objetivo es garantizar que cualquier usuario conserve en el futuro las libertades originales que definen al software libre, y este futuro incluye los trabajos derivados del software original.

Por ejemplo, el núcleo del sistema operativo Linux se encuentra liberado bajo esta licencia, garantizando que cualquier empresa o usuario que haga modificaciones y las distribuya esté obligado a publicar el código fuente y así contribuir con sus mejoras al crecimiento de Linux.

La licencia *copyleft* por antonomasia es la General Public License<sup>43</sup> (conocida por su acrónimo GPL) que ha sido formulada por la Free Software Foundation, y que goza de amplia popularidad. Es una licencia que cubre con gran detalle los diferentes aspectos del software. Existe una recopilación larga pero exhaustiva de preguntas frecuentes con sus respuestas sobre la GPL<sup>44</sup>.

Licencias de código abierto o permisivas. Aquellas que ceden el uso del programa bajo las condiciones que definen el software libre pero no obligan necesariamente a hacer públicas las mejoras que realicemos sobre el código. Con las licencias más permisivas este tipo alguien puede usar nuestro programa informático libre, hacer ampliaciones y crear un producto propietario sin compartir con el resto de la comunidad las mejoras introducidas. Así por ejemplo, Netscape creó su producto comercial

propietario *Netscape Directory Server* basándose en el código fuente del servidor del software libre de la Universidad de Michigan sin necesidad de tener que publicar las mejoras<sup>45</sup>, con lo cual sólo los clientes de Netscape podían beneficiarse de estas mejoras a pesar que la empresa había usado un programa libre.

En la comunidad de software libre encontraremos programadores que argumentan que las licencias permisivas dan más libertad a los creadores de software ya que no imponen ninguna restricción porque permiten incluso crear software propietario de un proyecto libre, y otros que argumentan que este tipo de licencias restan libertad a los usuarios ya que no garantizan que futuras versiones sean libres.

La licencia BSD<sup>46</sup> es una de las más comunes de este tipo. Lo más destacable es que permite el uso y explotación de la aplicación, en forma de programa o código, sin ningún tipo de limitación. También protege al autor de los usos que terceros puedan hacer de su nombre para publicitar el producto y obliga a menudo a dar crédito a los autores originales.

Licenciamiento dual. Se basa en que el autor cede su creación bajo dos licencias diferentes según el uso que se vaya a hacer de su software y las libertades y



obligaciones que se deseen adquirir. Bajo este modelo, una empresa puede ofrecer una versión libre de su programa y otra con condiciones más óptimas para las necesidades de sus clientes que quieran trabajar fuera del modelo del software libre. Este modelo es utilizado por bastantes empresas ya que permite ofrecer los productos de forma libre y no libre, según las necesidades de cada cliente.

Una de las empresas pioneras en el uso del sistema de licencia dual es Sleepycat Software<sup>47</sup>. El producto Berkeley DB está disponible bajo una licencia libre, que lógicamente garantiza acceso al código fuente, siempre y cuando el usuario acceda a hacer público el código fuente de las aplicaciones que desarrolle usando este producto. Como cualquier licencia libre, Sleepycat Software permite también a los que se acojan a la opción de licencia libre vender productos basados en su propio software<sup>48</sup>. Esta licencia permite que el software de Berkeley DB sea hoy en día usando en miles de aplicaciones libres, ya que de hecho es un software libre si nosotros desarrollamos también software libre.

Por el contrario, si lo que deseamos es desarrollar una aplicación usando Berkeley DB para una distribución propietaria de la cual no queremos licenciar su código

como software libre debemos adquirir una licencia tradicional del producto Berkeley DB, que además tiene un coste de varios miles de dólares.

### **3.4 Las patentes de software**

A diferencia de los derechos de autor que protegen la implementación de una idea, las patentes de software permiten proteger ideas y algoritmos. Las patentes son la concesión de un monopolio sobre una invención durante un periodo limitado de 20 años, tras el cual todo el mundo puede hacer uso de la misma. Originalmente fueron creadas para que las personas pudieran recuperar las grandes inversiones hechas en investigación, a cambio de que adelantaran al público los detalles de su creación, produciendo así un beneficio social.

Básicamente, el monopolio cubre la fabricación, distribución, comercialización y utilización del proceso o producto patentado – lo que impide, para el software, la creación de programas similares pero no iguales. A diferencia de los derechos de autor que se conceden de forma automática y gratuita, las patentes se han de solicitar expresamente en la Oficina de Patentes relevante (país por país, o a nivel global) por lo que tienen un coste económico muy elevado, de

hecho casi prohibitivo para empresas pequeñas e individuos.

Estados Unidos fue el primer país del mundo en permitir la patentabilidad del software en el año 1981. Hasta entonces, se consideraba que el software no se podía patentar. De hecho, la ley actual en Europa (de 1973) impide la patentabilidad de programas informáticos “puros” (es decir, sin ser asociadas a un proceso material de aplicación industrial). Esto no ha impedido que muchas patentes sobre simple software se hayan otorgado por las oficinas europeas de patentes – considerando la interacción con el ordenador una suficiente aplicación técnica. Actualmente, en la Unión Europea aún se esta debatiendo un cambio de ley, permitiendo y reconociendo en la práctica las patentes de software en sí.

El concepto de patentes aplicado a los programas de ordenador es un sin sentido. Por un lado, en la industria del software pueden no existir inversiones grandes en tiempo ni dinero en investigación sino la simple creación de productos y, por lo tanto, no hay razón retributiva para conceder una patente. Por el otro, la innovación informática es acumulativa y

basada en experiencias e información previas. Faltaría por tanto el requisito de “inventividad”.

Hoy en día ya existen los derechos de autor, las licencias de software, las marcas, y el secreto industrial, que representan mecanismos más que válidos por proteger a los creadores. Permitir la patentabilidad de programas informáticos es tan absurdo como permitir patentar las fórmulas básicas de la matemática ya que es abrir la puerta a patentar algoritmos universales y básicos que pueden ser parte de cualquier programa.

Con las patentes de software se hace realidad el establecimiento de un impuesto encubierto para trabajar con las nuevas tecnologías, y que solamente las empresas más grandes podrán pagar o intercambiar patentes en el juego de licencias de patentes cruzadas. Así por ejemplo, tenemos el caso de British Telecom, que afirmaba tener una patente sobre el concepto de enlace, tan básico en la propia web o los sistemas de documentación, o el caso de Amazon, que patentó el concepto de la compra en un único click "1-click". Estos son sólo dos casos representativos, pero hay centenares de patentes concedidas en los Estados Unidos sobre algoritmos e ideas básicas en la informática.

Las patentes permiten establecer a las empresas con más recursos económicos y legales barreras de acceso a las nuevas tecnologías minando la innovación de otras empresas, aumentan los costes de desarrollo de software innecesariamente y crean una incertidumbre a las empresas que desarrollan, ya que en cualquier momento pueden infringir una patente sin saberlo. Además, las patentes que existen no revelan el código fuente del programa, violando un aspecto esencial de la patentabilidad y la idea que la sociedad pueda beneficiarse de la invención.

El movimiento de software libre y una parte muy importante de la industria del software están en contra de la concesión de patentes de software por los motivos expuestos. En este sentido, desde el colectivo Proinnova<sup>49</sup> se han organizado acciones de protesta contra las patentes de software y se trabaja en concienciar a usuarios, empresas y administraciones del problema que suponen.

### **3.5 Marcas**

Las marcas son una forma de protección legal de la imagen y reputación de un negocio, que conceden al titular el uso exclusivo de un nombre o logotipo

(gráfico) de forma ilimitada en relación con una categoría determinada de productos. Las marcas suelen justificarse para que un cliente pueda identificar de forma clara que un producto o servicio provenga de un determinado fabricante. Tienen un coste económico razonable, se renuevan cada 5 o 10 años, y se han de solicitar en la oficina de marcas correspondiente a la zona de comercialización del producto.

Las marcas pueden usarse en el mundo del software libre para proteger un servicio o producto y hasta un grupo de desarrollo. Por ejemplo, nosotros podemos crear una aplicación, distribuirla con licencia libre, incluso de tipo *copyleft*, y registrar como marca el nombre del producto. La licencia libre de uso no incluye el uso de la marca. De esta manera concedemos todas las libertades del software libre, pero podemos impedir que terceros modifiquen y distribuyan el programa – quizás de menor calidad – con la misma marca. Protegemos así la inversión que hayamos realizado para crear y dar a conocer nuestro producto.

La marca Linux®, por ejemplo, es propiedad de Linus Torvalds, lo que le permite tener un control sobre el uso del nombre del núcleo del sistema operativo

GNU/Linux que él mismo inició a principios de los años 90.

### **3.6 Secreto comercial**

El secreto comercial es otra forma de protección legal que se utiliza habitualmente en la industria para proteger el software. El secreto comercial impide la divulgación de información (económica, técnica, etc.) que sustenta la ventaja competitiva de las empresas. El secreto se puede forzar de diferentes formas pero una habitual es con los contratos de confidencialidad (con clientes, consultores y empleados). Google, por ejemplo, usa el secreto comercial para no revelar cómo funcionan ninguno de sus algoritmos de búsqueda, clasificación y puntuación de las páginas.

Pero el secreto va en contra del espíritu de compartición de información que impera en el software libre y es incompatible con la apertura del código. Los acuerdos de confidencialidad enojaron especialmente a Richard Stallman ya en los años 70<sup>50</sup>

Esta forma de protección tiene que tenerse en consideración en las empresas cuyos empleados participan en el desarrollo de software libre, y en la provisión del código fuente del software: en muchos

casos, se puede considerar que el código fuente engloba los secretos de negocio – en ese caso no se puede proteger la información revelada indirectamente en un distribución libre.

### **3.7 Garantías e indemnizaciones**

Un aspecto que preocupa a usuarios y empresas es la garantía que ofrece un determinado programa informático, es decir, quién se hace responsable de si el programa no funciona correctamente y quién solucionará los posibles problemas.

Cualquier usuario sabe, por experiencia propia, que los programas no son perfectos. Los programas tienen errores. Cuanto mejor sea el proceso de ingeniería del software que se utiliza y mejores los programadores involucrados en su desarrollo, menos errores tendrá. El software tiene un media de 0,150 errores por cada 1000 líneas de código<sup>51</sup>. Si tenemos en cuenta que un producto como Open Office 1.0 tiene aproximadamente siete millones de líneas de código, la aritmética es sencilla. Microsoft por ejemplo admitió que cuando lanzó Microsoft NT 4.0 al mercado tenía más de 60.000 errores registrados<sup>52</sup> la mayoría pendientes de corregir. La calidad del



software es una preocupación y se dedican muchos esfuerzos a mejorarla, pero el software casi nunca es perfecto.

Para sorpresa de muchos, prácticamente la totalidad de las licencias de software propietario tienen cláusulas específicas en las que se declina cualquier responsabilidad y garantía sobre el software, se limita el coste de cualquier posible indemnización y se rechaza cualquier posible responsabilidad causada por un error del programa. Es habitual encontrar usuarios que creen que el software propietario tiene garantía por el simple hecho de que alguien vende el producto: la realidad es muy diferente, y una simple lectura a las licencias de software aclara este punto. La mayoría de licencias de software propietario se comprometen simplemente a reemplazar el soporte físico del programa (habitualmente un CD-ROM o DVD) y los manuales del mismo si estos están dañados.

Por su parte todas las licencias de software libre tienen cláusulas que se declinan cualquier tipo de garantía sobre el software y limitan cualquier indemnización. Podemos afirmar que prácticamente ningún tipo de software, sea libre o propietario, incluye garantía alguna.

En Europa, las leyes de protección de consumidores imponen obligaciones de calidad y funcionamiento adecuado o “satisfactorio” de los productos comercializados a consumidores (“adecuado” no significa libre de errores), las cuales en general el proveedor no puede limitar. Se discute, sin embargo, si estas protecciones se aplican al software libre, entre otras razones por su carácter casi de donativo. Para paliar estas “lagunas” y la desprotección del usuario, empresas comerciales de distribución de software libre como RedHat ofrecen garantías de funcionamiento, corrección de errores, soporte y mantenimiento a sus clientes.

### **3.8 Protección contra violaciones de derechos**

La incertidumbre que crea el problema de las patentes en Estados Unidos donde rutinariamente se conceden patentes por invenciones triviales, y los posibles juicios por violación de derechos de autor son temas que preocupan en especial a empresas y gobiernos. En algunos casos pueden frenar la adopción de soluciones basadas en software libre.

A medida que el software libre gane en adopción y más empresas hagan negocio implementando sistemas libres aumentará el riesgo de que aparezcan

fabricantes de software propietario que vayan a juicio para reclamar posibles violaciones de derechos de autor o patentes no sólo a los distribuidores del software sino también a sus usuarios.

Algunos distribuidores de soluciones Linux, como Novell o Hewlett Packard, proveen protección legal a los usuarios de sus soluciones de software. Incluso Red Hat llega más lejos, con el compromiso de reemplazar cualquier software incluido en su distribución<sup>53</sup> que sea objeto de problemas legales. También existen empresas como Open Source Risk<sup>54</sup> que ofrecen un seguro en caso de cualquier problema legal que pudiera derivarse del uso de software libre.

Por su parte, la comunidad de software libre ha reaccionado y ha empezado a organizar recolectas de fondos<sup>55</sup> para poder defenderse legalmente. Desgraciadamente, existen empresas que tienen dinero suficiente para prolongar los juicios durante años y crear una erosión mediática, y económica, al software libre.

### 3.9 Gestión digital de contenidos

Internet desde sus inicios hasta mediados de los noventa representó la panacea del mundo libertario: una red de ámbito mundial sin regular donde imperaba un código ético implícito de buen comportamiento<sup>56</sup> que en general todos respetaban. El ciberespacio era básicamente un medio autorregulado muy influenciado por la comunidad científica y universitaria. Durante bastante tiempo representó un espacio global con nulo control por parte de las autoridades. Muchos pensaron que, por el hecho de tratarse de un espacio que abarcaba más allá de fronteras de países, quedaría fuera de toda regulación. Con la popularización de Internet y la llegada masiva de usuarios no nacidos ya en aquella cultura inicial con código ético implícito se produjeron los primeros cambios.

La aparición de Napster fue un fenómeno que revolucionó la Red y enojó a empresas muy poderosas de fuera del ciberespacio. Napster introdujo la idea, y el software, de que la Red podía ser utilizada para el intercambio de contenido digital, inicialmente música. Una vez la música se encuentra en formato digital (en un archivo), el coste de transferencia es prácticamente nulo. La reacción inmediata de las discográficas y de

algunos grupos musicales fue perseguir legalmente el fenómeno de la forma más contundente posible. Han sido numerosos los casos de empresas y usuarios demandados por intercambio ilegal de archivos.

Sin embargo, el fenómeno, en vez de disminuir, creció y se expandió a nuevos tipos de contenidos digitales, como libros, películas, o software. Aplicaciones como Kazaa<sup>57</sup> Emule<sup>58</sup> o Direct Connect<sup>59</sup> han puesto al alcance del usuario herramientas de intercambio de archivos, en lo que se conoce como redes entre iguales o P2P (*peer-to-peer*).

El fenómeno Napster<sup>60</sup> fue la constatación final de que algo había cambiado. Los formatos de alta compresión de vídeo y audio, las grabadoras de CD-ROM y más tarde de DVD de bajo coste, el abaratamiento de los soportes vírgenes multimedia, la popularización de la conexiones de banda ancha a Internet, son fenómenos que, combinados, han puesto en manos de los usuarios la facilidad de crear, copiar, y transmitir contenidos digitales prácticamente sin coste. Napster, además, aportó un sistema de distribución de contenidos musicales donde los propios usuarios podían publicar e intercambiar contenidos. Aunque una gran parte de estos contenidos eran copiados ilegalmente, la amenaza más profunda era la aparición de un nuevo

modelo de distribución donde las distribuidoras no eran necesarias.

Para frenar este fenómeno las discográficas, la industria cinematográfica, y los fabricantes de hardware y software están trabajando en la creación de sistemas de gestión de contenidos digitales, que son un conjunto de tecnologías que permiten al titular de los derechos de autor tener un control sobre cómo sus obras se distribuyen a través de medios digitales, tanto Internet como soportes como el CD-ROM o el DVD.

Mientras que los derechos de autor proporcionan protección legal a los autores, los sistemas de gestión de contenidos digitales proporcionan protecciones basadas en soluciones tecnológicas. Existen soluciones simples como *Content-Scrambling System* que impide que DVD comprados en una zona geográfica del mundo puedan ser utilizados en otra libremente, hasta sistemas conocidos como la informática de confianza<sup>61</sup>. Una propuesta centrada en incrementar la seguridad de los ordenadores personales basada en añadir protecciones criptográficas al contenido digital e identificar al cliente de una forma precisa.

En el momento de escribir estas líneas todos los sistemas de gestión de contenidos digitales han fracasado en detener la copia ilícita de contenidos y han ocasionado inconvenientes a usuarios legítimos. A nivel mundial, se han promulgado nuevas leyes para proteger estos medios tecnológicos de protección de las obras y la información usada para la gestión de derechos (*Rights Management Information*). Es ilegal eludir las protecciones tecnológicas eficaces y remover las RMI – y también ilegal promocionar, comercializar o distribuir sistemas tecnológicos cuyo fin es dicha eliminación.

Dentro del mundo del software libre estos sistemas se ven como una amenaza a la libertad personal de los usuarios, que crean barreras e inconvenientes en la distribución lícita de software libre<sup>62</sup> y establecen una situación de potencial monopolio donde las tecnologías para distribuir contenidos están en manos de unos pocos. Asimismo, la utilización no solamente de varios programas de uso doble (por ejemplo, para la copia o grabación de CDs) sino también de ciertos programas libres podría considerarse una infracción potenciales de estas nuevas leyes.

### **3.10 *Creative Commons*: contenidos digitales libres**

Algunas de las ideas del movimiento del software libre, como usar los derechos de autor para garantizar permiso a terceros para modificar y distribuir obras, han inspirado movimientos similares entre autores de libros, artículos periodísticos, música, o vídeo.

Una de las iniciativas más populares es *Creative Commons*<sup>63</sup> una organización sin ánimo de lucro que ha sido creada porque sus impulsores consideran que la legislación actual de derechos de autor no se ajusta a las necesidades de nuestros días y es demasiado restrictiva.

El avance de la tecnología en los últimos años permite que con equipos de bajo coste cualquier persona pueda crear, combinar, y distribuir con un coste menospreciable (en Internet por ejemplo) contenidos digitales. Son tremendamente populares los weblogs, que aportan gran contenido a la Red, libros y en general artistas que distribuyen sus trabajos a través de Internet por la difusión que les ofrece.

Sin embargo, las facilidades de la tecnología contrastan con la rigidez del sistema jurídico.



Para muchos creadores, los derechos de autor no son un mecanismo adecuado para explotar sus creaciones ya que no proveen un sistema que permita de forma automática ciertos usos de sus obras, y exigen siempre el permiso del autor. De esta forma, si estamos realizando una creación combinando creaciones de terceros, como por ejemplo un artículo, debemos pedir permiso al autor original. Esto establece un entorno donde por defecto todos los derechos de una obra se encuentran reservados y crea una situación donde la creatividad y la innovación en base al compartir obras e ideas vuelven a ser realmente difíciles o limitadas.

Adicionalmente, el periodo que debe transcurrir para que una obra pase al dominio público, generalmente 70 años desde la muerte del autor (pero varía entre diferentes países), parece excesivo en la nueva sociedad de la información marcada por ritmos cada vez más acelerados. Esto hace que hay muchísimas obras que están bloqueadas por los derechos de autor y debemos esperar años hasta que puedan convertirse en un bien común y pasar al dominio público.

El objetivo de *Creative Commons* es crear un ecosistema de contenido digital abierto donde los autores puedan ceder parte de los derechos sobre sus obras y retener sólo los derechos que realmente tienen

interés, permitiendo así aumentar el número de creaciones libremente disponibles.

*Creative Commons* ofrece a través de su web una página<sup>64</sup> que permite a los creadores de las obras escoger con dos preguntas muy sencillas qué van a permitir que terceros hagan con sus trabajos y qué derechos desean conservar. La primera pregunta hace referencia a si queremos permitir usos comerciales de nuestros trabajos, es decir, si vamos a permitir que la gente pueda siempre modificar, distribuir, o reproducir la obra o solamente cuando no haya una finalidad comercial. La segunda pregunta hace referencia a si queremos que la gente pueda o no modificar nuestra obra, y si lo hace, si queremos obligarles a que las versiones modificadas deban ser compartidas (tipo *copyleft*). Con estas dos preguntas, la página web nos crea una licencia, igual que sucede en el software libre, que nos permite ceder los derechos de nuestra creación bajo las condiciones que hemos especificado. Una vez realizada nuestra selección obtenemos tres representaciones de nuestra solución: una en lenguaje fácilmente comprensible, otra en lenguaje jurídico (la licencia o contrato) y otra en lenguaje informático (en forma de código web).

Además, en la web de *Creative Commons*<sup>65</sup> hay un buscador que contiene miles de creaciones digitales que han sido liberadas hasta ahora y que incluyen desde canciones de Gilberto Gil hasta el fondo digital de la cadena pública BBC.

-----

<sup>40</sup><http://sourceforge.net/>

<sup>41</sup><http://www.fsf.org>

<sup>42</sup><http://www.gnu.org/copyleft/copyleft.html>

<sup>43</sup><http://www.gnu.org/copyleft/gpl.html>

<sup>44</sup><http://www.gnu.org/licenses/gpl-faq.html>

<sup>45</sup><http://wp.netscape.com/newsref/pr/newsrelease126.html>

<sup>46</sup><http://www.opensource.org/licenses/bsd-license.php>

<sup>47</sup><http://www.sleepycat.com>

<sup>48</sup><http://www.sleepycat.com/download/licensinginfo.shtml>

<sup>49</sup><http://proinnova.hispalinux.es>

<sup>50</sup><http://www.faiozilla.org/ch01.html>

<sup>51</sup> *Preliminary Data from Global Software Process Survey*, Michael A. Cusumano

<sup>52</sup><http://www.accountingsoftwareadvisor.com/southware/techsouthware.htm>

<sup>53</sup> [http://zdnet.com.com/2100-1104\\_2-5143326.html?tag=nl](http://zdnet.com.com/2100-1104_2-5143326.html?tag=nl)

<sup>54</sup><http://www.osriskmanagement.com/>

<sup>55</sup>[http://news.com.com/Intel+chips+in+on+anti-SCO+defense+fund/2100-7344\\_3-5138820.html?tag=nl](http://news.com.com/Intel+chips+in+on+anti-SCO+defense+fund/2100-7344_3-5138820.html?tag=nl)

<sup>56</sup> NetEtiquette

<sup>57</sup><http://www.kazaa.com/>

<sup>58</sup><http://www.emule-project.net/>

<sup>59</sup><http://www.neo-modus.com/>

<sup>60</sup><http://www.napster.com/>

<sup>61</sup>En inglés *Trusted computing*

<sup>62</sup><http://www.gnu.org/philosophy/can-you-trust.html>

<sup>63</sup><http://creativecommons.org/>

<sup>64</sup><http://creativecommons.org/license/>

<sup>65</sup><http://creativecommons.org/getcontent/>

## **4. LA PRODUCCIÓN DEL SOFTWARE LIBRE**

El modelo de producción del software libre ha demostrado tener un éxito considerable y ser altamente productivo en el desarrollo de programas tan complejos como Linux o Apache. Más allá del software, proyectos como Wikipedia<sup>66</sup> o Open Directory<sup>67</sup> muestran que este modelo es exportable a otros tipos de creaciones intelectuales.

### **4.1 Motivos que impulsan al desarrollo de software libre**

A primera vista puede parecer sorprendente que software que ha costado miles de horas de desarrollar se distribuya de forma libre y sin coste alguno a través de la Red y otros medios tanto físicos como digitales. Los incentivos y motivaciones que pueden llevar a una persona a contribuir en el desarrollo del software libre son variados, y en general, al contrario de lo que se puede pensar, los de carácter económico no son los principales.

## Motivaciones sociales

Diversión. Linus Torvalds explica<sup>68</sup> que, como muchos otros *hackers*, empezó a trabajar en el desarrollo de Linux simplemente por la diversión que implicaba hacer algo interesante. Un eje central en la cultura *hacker* ha sido siempre hacer cosas ingeniosas que representan un reto, además de ser una excelente manera de mejorar nuestro conocimiento. Enfrentarse a un reto y encontrar una buena solución produce un estado de satisfacción similar al que experimenta un científico cuando realiza un descubrimiento.

Reputación. La cultura del regalo siempre ha tenido un papel importante en las sociedades donde ha habido una cierta abundancia de bienes y ha permitido tejer enlaces sociales más fuertes. En el software libre, tal como describe Eric S. Raymond<sup>69</sup>, la economía del regalo en este entorno tiene como objetivo el ganar reputación entre los otros miembros de la comunidad.

Convicción ideológica. Cuando Richard Stallman creó la *Free Software Foundation* en 1984 e inició el proyecto GNU su mayor motivación era la creencia de que el software libre era necesario para preservar las

libertades de los usuarios y era bueno para la sociedad. Aunque no siempre es la motivación principal que impulsa a una persona a colaborar en un proyecto libre, sí que es sin duda uno de los factores que influyen enormemente en ello.

## **Motivaciones tecnológicas**

Resolver un problema propio. Si un desarrollador de software tiene una necesidad que no cubre ningún programa es probable que escriba el programa que necesita o, si encuentra un problema en una aplicación de terceros que quiera resolver, que colabore en el proyecto solucionando el problema específico.

Aprender. El mundo del software libre es un excelente campo de aprendizaje para profesionales de las tecnologías de la información. Ofrece la posibilidad de estudiar el código fuente de miles de programas informáticos, participar en su desarrollo y corrección de errores, o simplemente mejorar nuestros conocimientos generales siguiendo la evolución de las diferentes comunidades.

## **Motivaciones económicas**

Parte de un proyecto empresarial o universitario. Son habituales los casos donde una empresa – como pueda ser Red Hat o Novell – tiene desarrolladores colaborando en un proyecto libre o una universidad que ha conseguido apoyo financiero para desarrollar o mejorar un proyecto libre. Estos desarrolladores contribuyendo al software libre están realizando su trabajo por el cual son remunerados. En el caso de las empresas es común que estos desarrolladores sean contratados de las propias comunidades, es decir, voluntarios que ya estaban contribuyendo al proyecto y que ahora se podrán dedicar a tiempo completo como parte de su trabajo.

Bajo coste de oportunidad. El software libre tiene unas barreras de entradas muy bajas y ofrece un conjunto de recursos muy amplios que hacen que, efectivamente, el coste de oportunidad sea realmente bajo. Cualquier persona con un PC y una conexión a Internet puede iniciar o contribuir a un proyecto de software libre.

Reputación. El software libre aplica el modelo académico tradicional de reconocimiento del trabajo



por parte de otros participantes en la comunidad. De esta manera se obtiene reconocimiento del capital intelectual del participante. Esto se puede traducir en oportunidades laborales. Muchos de los desarrolladores más destacados han tenido ofertas laborales de RedHat, Novell o Apple, entre otros, para trabajar en proyectos relacionados con su actividad en el software libre. Así por ejemplo, uno de los líderes del sistema operativo libre FreeBSD, Jordan Hubbar, fue contratado por Apple<sup>70</sup> para trabajar en el sistema operativo para Mac basado en FreeBSD.

Todos los estudios empíricos realizados a desarrolladores de software libre demuestran que habitualmente es una combinación de estos motivos los que les impulsan a colaborar en comunidades de software libre.

## **4.2 Perfil de los desarrolladores de software libre**

En año 2002 el *Boston Consulting Group* realizó un estudio<sup>71</sup> a 526 desarrolladores de software libre para conocer las motivaciones por las cuales contribuyen a estos proyectos. El estudio afirma que las principales son la estimulación intelectual, mejora de los conocimientos y pensar que el software libre es conveniente. El 98% de los entrevistados son hombres

y el 70.4% se encuentran en la franja de entre 22 a 37 años de edad. En cuanto a su ubicación geográfica, el 48% viven en Estados Unidos y el 42.2% en Europa.

Durante el año 2003, Karim R. Lakhani y Robert G. Wolf realizaron un estudio basado en una encuesta vía web a 684 desarrolladores de software libre de 287 proyectos diferentes. Según las conclusiones de su investigación, el perfil del típico programador es un hombre (97.5%) con una media de edad de 30 años. El estudio concluye que mejorar las habilidades como programador y la estimulación intelectual son los motivos más habituales para escribir software libre destacando que aproximadamente el 40% de los encuestados contribuían al software libre como parte de su trabajo.

### **4.3 Herramientas colaborativas**

El mundo del software libre ha desarrollado y adaptado herramientas colaborativas que permiten a grupos de personas trabajar de forma conjunta en el desarrollo de proyectos a través de Internet.

La web ha tenido un papel central en el crecimiento de Internet y, consecuentemente, en la difusión del software libre. Todos los proyectos de software libre

tienen una web que sirve de punto de inicio y encuentro de los recursos que usuarios y desarrolladores necesitan. Algunos proyectos tienen sitios web propios y otros utilizan repositorios públicos como SourceForge.

Cada proyecto suelen tener una o varias listas de distribución donde se discute la dirección que toma el proyecto, se plantean posibles soluciones a problemas complejos, y se toman decisiones sobre el proceso de desarrollo del software. También es usual disponer de canales de chat que son mucho más inmediatos y tienen mayor interactividad que el correo electrónico y suelen utilizarse para resolver problemas muy concretos de forma rápida.

El código fuente del proyecto se ubica en un servidor centralizado que gestiona un software de control de versiones. Estos sistemas de control de versiones funcionan como la columna vertebral del proyecto. Son sistemas que señalan las diferentes versiones del código fuente para identificarlas posteriormente, facilitan el trabajo en paralelo de grupos de usuarios, indican la evolución de los diferentes módulos del proyecto, y disponen de un control detallado de los cambios que se han realizado; funciones que son indispensables durante la vida del proyecto. Estos

sistemas no sólo tienen aplicación en el desarrollo del software, sino que además son ampliamente utilizados en la creación de documentación, sitios web y en general cualquier proyecto colaborativo que requiera trabajar con equipos de personas de forma concurrente. CVS<sup>72</sup> (*Concurrent Versions System*) es el programa más utilizado en el mundo del software libre para el control de versiones de software y lo utilizan proyectos como Mozilla, OpenOffice.org, KDE o GNOME, por mencionar sólo algunos.

Cualquier usuario puede acceder al servidor donde se encuentra el código fuente de un proyecto, siguiendo la idea de que todo el mundo puede ver el código, pero sólo unos pocos disponen de privilegios para añadir nuevo código o modificar el existente.

Un aspecto central en cualquier proyecto de software es la gestión y el seguimiento de los errores. Cuando Netscape en 1998 liberó el código de Mozilla se encontró con la necesidad de tener una aplicación de gestión de errores vía web que permitiera la interacción entre usuarios y desarrolladores. Decidieron adaptar la aplicación que usaban internamente en Netscape a las necesidades de un proyecto abierto y así nació Bugzilla<sup>73</sup>. Inicialmente fue el sistema de gestión y seguimiento de errores del

proyecto Mozilla, pero con el tiempo ha sido adoptado por muchos proyectos libres, incluyendo KDE, GNOME, o OpenOffice.org entre otros. Bugzilla permite a los usuarios enviar errores facilitando la clasificación del error, su asignación a un desarrollador para que lo resuelva, y todo el seguimiento de las incidencias relacionadas.

Los proyectos libres tienden a crear documentación con manuales de uso y instalación o recopilatorios de preguntas y respuestas más frecuentes. Los sistemas Wiki<sup>74</sup> o sus derivados se han convertido en herramientas muy populares para facilitar la edición colaborativa de contenido abierto. Estas herramientas permiten que los usuarios creen y corrijan cualquier página en un sitio del web incentivando la participación de usuarios. Este es el mismo concepto y sistema que se utiliza para producir la enciclopedia Wikipedia.

Todas estas herramientas, junto con el talento de los miembros de los diferentes proyectos, es lo que hace posible la producción de software libre.

## 4.4 Organización de proyectos libres

En 1997 Eric S. Raymond publicó un ensayo bajo el título *la Catedral y el bazar*<sup>75</sup> que describía dos estilos de desarrollo en las comunidades de software libre. Aunque se trata de un ensayo y no está acompañando de ninguna evidencia empírica, y que ha recibido algunas críticas por su falta de precisión, es cierto también que se ha convertido en un referente para cualquiera que pretenda entender cómo funciona la organización de proyectos libres.

Raymond defiende que existen dos modelos de organización de proyectos libres: la catedral y el bazar. En el modelo catedral existe una jerarquía muy definida de quien es el responsable de cada área del proyecto y es habitualmente ejecutado por un grupo reducido de personas. Este modelo tiene paralelismos en la forma que se construían catedrales en la Edad Media donde la construcción era un proceso muy planificado. En el modelo bazar el desarrollo se realiza entre muchos desarrolladores sin una jerarquía clara, de la misma forma que están organizados los bazares en oriente.

Aunque ambos modelos no son analogías perfectas para todos los proyectos de software libre, sí que

definen dos estructuras comunes, aunque lógicamente muchos proyectos son combinación de ambas en mayor o menor grado.

#### **4.5 Toma de decisiones**

Algunas veces podemos tener la impresión de que existe cierta desorganización en los proyectos de software libre por la forma tan abierta en que se gestionan. La realidad es muy diferente, ya que existen tareas muy definidas con sus respectivos responsables. Aunque se respira un ambiente de cooperación, la mayoría de proyectos combinan contribuciones de empresas, voluntarios y instituciones que además representan diferentes sensibilidades.

En los proyectos pequeños existe un único responsable de proyecto. Se puede llegar a esta responsabilidad siendo la persona que fundó el proyecto o porque el fundador pasó el testigo de la responsabilidad con el beneplácito del resto de integrantes de proyecto. En proyectos de gran envergadura, como por ejemplo Mozilla.org, se distribuye la toma de decisión entre un conjunto de personas<sup>76</sup>. A cada una de ellas se le asigna un módulo sobre el cual tiene potestad. Los responsables del

proyecto toman decisiones como si se debe o no aceptar o no una contribución externa concreta a un proyecto – por ejemplo la corrección de un error -, cuándo se liberan las diferentes versiones del software, o decidir qué errores son imprescindibles que sean corregidos antes de liberar una nueva versión.

Algunos proyectos han formalizado un poco más el proceso de toma de decisiones. El proyecto Apache cuenta con el Apache Group, que está formado por desarrolladores que han colaborado durante un periodo largo de tiempo y que son escogidos por votación<sup>77</sup> para formar parte del núcleo de personas que toman las decisiones. El proyecto FreeBSD tiene también un consejo formado por desarrolladores que han participado activamente en el proyecto y que son los responsables de las decisiones más importantes. Como vemos, se aplica un sistema meritocrático donde los responsables son escogidos por el trabajo que han aportado al proyecto.

#### **4.6 Ejemplo: la comunidad Abiword**

Abiword es un proyecto que tiene como objetivo producir un procesador de textos libre para Unix, Linux, Mac y Windows. Durante los dos primeros



años fue desarrollado por SourceGear en colaboración con la comunidad hasta que en el año 2000 pasó a ser un proyecto mantenido exclusivamente por una comunidad de voluntarios liderados por Dom Lachowicz<sup>78</sup>.

Lachowicz fue elegido responsable de proyecto por acuerdo de todos los integrantes. Actualmente existe además un responsable de la versión de Linux, otro de Mac, y otro de Windows (que resulta ser el autor de estas líneas) que nos encargamos de asegurar que nuestras versiones van incorporando el código necesario para cada plataforma. A día de hoy Abiword cuenta con 12 desarrolladores considerados principales y desde sus inicios han ayudado 250 voluntarios que han colaborado mejorando la aplicación así como traduciéndola a más de 50 lenguas.

Los usuarios y desarrolladores de Abiword se comunican diariamente a través de las diferentes listas del proyecto<sup>79</sup> así como del canal Abiword del servidor de chat gimp.org. Semanalmente, se recopila lo más importante que ha sucedido en el desarrollo de la aplicación y se publica en un boletín de noticias<sup>80</sup>. Estas noticias de carácter semanal son muy útiles para poder continuar estando al día en el desarrollo del

proyecto en épocas que no podemos dedicarle el tiempo que requiere leer los mensajes de la lista e ir apareciendo por el canal chat.

Anualmente, una parte del grupo de desarrolladores principales y entusiastas de Abiword se reúne en la GUADEC<sup>81</sup> que es la conferencia anual de programadores de GNOME y que sirve de marco de encuentro. En esta reunión presencial se suelen acordar las funcionalidades que se incluirán en las próximas versiones del proyecto en base a las ideas que han ido comentando los usuarios y desarrolladores. La dirección del que tomará el proyecto en los próximos meses se recoge en la hoja de ruta<sup>82</sup> que incluye también quién será el responsable de implementar cada una de las nuevas funcionalidades.

Abiword, como otros proyectos, tiene siempre dos versiones de la aplicación: una estable y otra de desarrollo. La versión estable es la que se considera para usuarios y a medida que aparecen errores nuevos se van corrigiendo pero no se añade ninguna funcionalidad nueva. La versión de desarrollo, por su parte, incluye también las correcciones que se van efectuando y además toda la nueva funcionalidad que incluirá la próxima versión. Con este sistema, siempre

tenemos una versión estable para ofrecer a nuestros usuarios y una versión de desarrollo que refleja la dirección que tomará el proyecto para usuarios avanzados y desarrolladores.

En cuanto a documentación para los usuarios de Abiword, el programa incluye una completa ayuda en línea que también puede ser accedida vía web y un recopilatorio de preguntas y respuestas más frecuentes basado en un sistema colaborativo tipo Wiki<sup>83</sup>. Pero los usuarios también tienen un papel muy importante en la dirección que toma el proyecto ya que pueden votar qué errores quieren que sean corregidos primero<sup>84</sup>, pueden proponer nuevas funcionalidades<sup>85</sup>, y pueden reportar cualquier problema que tengan para que los desarrolladores lo tengan documentado y lo puedan corregir en las próximas versiones del programa.

-----

<sup>66</sup><http://www.wikipedia.org/>

<sup>67</sup><http://www.dmoz.org/>

<sup>68</sup>Ver prólogo de *La ética del hacker* de Pekka Himanen

<sup>69</sup><http://www.catb.org/~esr/writings/homesteading/homesteading/>

- <sup>70</sup> <http://www.workingmac.com/article/32.wm>
- <sup>71</sup> <http://www.bcg.com/opensource/BCGHACKERSURVEY.pdf>
- <sup>72</sup> <https://www.cvshome.org/>
- <sup>73</sup> <http://www.bugzilla.org/>
- <sup>74</sup> <http://www.wiki.org/>
- <sup>75</sup> <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>
- <sup>76</sup> <http://www.mozilla.org/hacking/module-ownership.html>
- <sup>77</sup> <http://www.apache.org/foundation/how-it-works.html>
- <sup>78</sup> <http://www.advogato.org/person/cinamod/>
- <sup>80</sup> <http://www.abisource.com/maillinglists/>
- <sup>81</sup> <http://www.abisource.com/information/news/>
- <sup>82</sup> <http://www.guadec.org>
- <sup>83</sup> <http://www.abisource.com/developers/2.2-roadmap.phtml>
- <sup>84</sup> <http://www.abisource.com/twiki/bin/view/Abiword/AbiWordFAQ>
- <sup>85</sup> <http://www.abisource.com/support/vote/>
- <sup>86</sup> <http://www.abisource.com/contribute/suggest/>

## **5. PROYECTOS LIBRES**

A continuación comentaremos algunos de los proyectos de software libre más conocidos. Es importante saber cómo se iniciaron, su estado actual, y la dirección que están siguiendo para entender un poco más cómo funciona y hacía dónde va el software libre. Cualquier selección de este tipo siempre deja en el cajón proyectos que quizás han realizado contribuciones mayores en términos absolutos y en importancia, pero que son menos conocidos. Hemos intentando presentar aquí los más representativos.

### **5.1 Linux**

El sistema operativo Linux se ha convertido en el buque insignia del movimiento del software libre de tal forma que algunos usuarios no iniciados confunden el concepto de Linux con el de software libre. Es un sistema tan popular que podemos ver cómo hoy en día en el mundo hay más grupos de usuarios de Linux que no de software libre, aunque lógicamente todos los usuarios de Linux utilizan otros programas libres.

Linux nació de la mano de Linus Torvalds. En 1991, cuando tenía 21 años y era estudiante de la universidad de Helsinki en Finlandia, anunció en un foro de Internet la primera versión del núcleo de su sistema operativo. Torvalds había estado trabajando durante seis meses en crear un sistema operativo para sistemas Intel 386. A los pocos días de ser publicado numerosos *hackers* empezaron a contribuir con mejoras y arreglos. El núcleo que Torvalds había escrito era precisamente el corazón que le faltaba al sistema GNU que Richard Stallman estaba diseñando desde principios de los 80.

Así se crearon las primeras distribuciones Linux, que incluían el núcleo desarrollado por Torvalds más un conjunto de herramientas GNU. Hoy en día, existen distribuciones enfocadas a usos específicos como hacer de cortafuegos en una red o de servidores en Internet, o para ámbitos específicos como a educación (LinEx por ejemplo), o de uso general.

Entre las distribuciones comerciales de Linux más conocidas se encuentran Red Hat<sup>87</sup>, Mandrake Linux<sup>88</sup> y Suse<sup>89</sup> (ahora parte de Novell). Todas ellas están pensadas para el usuario final y son de amplio uso. Dentro de las distribuciones Linux merece una especial mención Debian<sup>90</sup>. Esta distribución fue

creada en agosto de 1993 por Ian Murdock con el objetivo de proporcionar una distribución totalmente libre. Debian está basada en gran parte en voluntarios y no tiene ninguna empresa detrás de su organización. Actualmente, tiene más de 10.000 paquetes de software listos para instalar.

Un área donde Linux es de creciente popularidad son los LiveCD. Se trata de distribuciones que se ejecutan directamente desde un CD-ROM y no requieren ningún tipo de instalación en el disco duro ni indicar ningún tipo de parámetro de configuración. Simplemente introduciendo el CD-ROM e iniciando el PC se pone a funcionar. Este tipo de distribuciones gestionan y reconocen automáticamente muchos tipos de tarjetas gráficas, de sonido, dispositivos SCSI y otros periféricos. Se están usando extensamente para dar a conocer Linux a los usuarios neófitos y se pueden conseguir a través de revistas de informática. La distribución LiveCD más popular es la alemana Knoppix<sup>91</sup>, que está basada en Debian y de la cual se han hecho muchas versiones personalizadas.

Linux no sólo ha innovado como software sino también en su modelo de producción. Torvalds impulsó desde el inicio un ritmo de liberación de versiones constante bajo el lema “libera pronto, libera

a menudo”. Bajo este sistema Torvalds iba publicando versiones de Linux cada pocas semanas que imprimían un gran dinamismo al desarrollo del producto. Torvalds, que aún es el responsable del proyecto, ha sido capaz de gestionar las contribuciones al proyecto de una forma sensata, lo que le ha valido el apodo de dictador benevolente.

Linux, hoy en día, está disponible para todo tipo de plataformas de hardware – desde *mainframes* a asistentes personales - y es usado en millones de instalaciones en el mundo. Google, Amazon, la Agencia Europea Espacial o la NASA figuran entre los muchos usuarios que basan su infraestructura en Linux y los millones de usuarios individuales que lo usan diariamente.

## 5.2 Apache

En 1994 se creó la empresa Netscape, al principio bajo el nombre de Mosaic Communications. Detrás quedaba el navegador Mosaic y otras herramientas que estudiantes y administradores de la Universidad de Illinois<sup>92</sup> habían escrito vertiginosamente en pocos meses y que habían popularizado la web en Internet. Uno de estos programas que quedó prácticamente



huérfano fue el servidor web NCSA httpd escrito por Rob McCool.

El programa era libre y tenía muchos usuarios que lo continuaban utilizando pero que necesitaban introducirle mejoras. Así se fue creando una colección de parches para añadir funcionalidad extra al antiguo httpd. Hacia 1995 se creaba el proyecto Apache tomando como base estos parches y el trabajo del programa httpd de NCSA.

Apache es uno de los ejemplos de comunidades de desarrollo que han sido tremendamente productivas coordinando más de 800 contribuidores voluntarios, de empresas y universidades. El proyecto cuenta con el Apache Group que está formado por desarrolladores que han colaborado durante un periodo largo de tiempo y que son escogidos por votación<sup>93</sup> para formar parte del núcleo de personas que toman las decisiones.

Hoy en día, Apache es el servidor web más usado con una cuota del 67% sobre el total de servidores en Internet. Junto con los lenguajes de programación PHP y Perl y el sistema de bases de datos MySQL forman el conjunto de herramientas libres de creación web más populares en la Red.

### 5.3 KDE

En 1996, Matthias Ettrich, un estudiante alemán de la Universidad de Tuebingen<sup>94</sup>, inició el proyecto KDE<sup>95</sup> con el objetivo de crear un entorno de escritorio gráfico para entornos Unix. Dentro del proyecto KDE se han desarrollado múltiples aplicaciones como Kmail, un completo gestor de correo electrónico; Konqueror, un navegador web y gestores de archivos, o Koffice, un paquete ofimático de prestaciones básicas.

KDE tomó la polémica decisión de basar su desarrollo en la librería QT escrita por la empresa TrollTech<sup>96</sup>. Esta librería era entonces gratuita para el uso no comercial pero no era software libre. Tres años más tarde TrollTech anunciaba que licenciaba la librería QT bajo la licencia GPL, lo cual la convertía en libre, pero sólo para usos libres.

Anualmente los desarrolladores y usuarios de KDE se reúnen en el aKademy<sup>97</sup>. Durante el evento, que dura varios días, se dan charlas sobre desarrollo sobre KDE, se corrigen errores de forma conjunta, se deciden nuevas funcionalidades del producto, y se dan

talleres de uso de los diferentes programas y tecnologías relacionadas con KDE.

Aunque el desarrollo de KDE se realiza principalmente por voluntarios, diversas empresas como Suse (ahora Novell), Mandrake o TrollTech colaboran con recursos y desarrolladores a tiempo completo en el proyecto. Una de las empresas que también colabora en el proyecto y ha construido un producto a partir de KDE es Xandros<sup>98</sup>, que comercializa escritorios avanzados para entornos Unix.

## 5.4 GNOME

Los mexicanos Miguel de Icaza y Federico Mena lanzaron en 1997 el proyecto GNOME<sup>99</sup> con el objetivo de desarrollar un escritorio moderno y sencillo para entornos Unix. Un año antes se había iniciado el proyecto KDE<sup>100</sup> pero la decisión de usar una librería propietaria para implementarlo enojó a sectores de la comunidad de software libre, ya que para muchos *hackers* el objetivo era hacer un sistema operativo completamente libre, y una parte tan importante del mismo no podía basarse en software propietario. Con el mismo objetivo que KDE,

decidieron empezar su propio proyecto basándolo completamente en software libre.

Después de muchos meses de esfuerzos, en marzo de 1999 en el marco de Linux World Expo en San José, se anunciaba GNOME 1.0. Era una versión importante ya que culminaba más de dos años de esfuerzos y brindaba una plataforma para que pudieran desarrollarse aplicaciones que aprovecharan toda su funcionalidad. Desde entonces GNOME ha ido mejorando y creciendo a un ritmo imparable, y hoy en día destaca por su alto grado de usabilidad.

Desde el año 2000 la comunidad GNOME organiza la GUADEC<sup>101</sup>, un evento de carácter anual que sirve de foro de encuentro entre desarrolladores, usuarios, gobiernos, y empresas que están involucradas en el desarrollo y uso de GNOME.

GNOME tiene una fundación que coordina la liberación de nuevas versiones y determina qué proyectos forman parte del mismo. La fundación actúa como voz oficial ante los medios de comunicación y coordina la creación de materiales educativos y documentación para ayudar a los usuarios a aprender a usar el entorno. Además, representa al proyecto en

conferencias y ayuda a crear los estándares técnicos y especificaciones para el mismo.

A parte de la fundación GNOME, existe un gran número de empresas que contribuyen diariamente y desde hace años a la mejora del sistema. Una de las primeras fue Red Hat que, a través de los Red Hat Labs, trabaja en la mejora general de GNOME. Ximian, fundada en 1999 y adquirida por Novell en Agosto del 2003, ha contribuido también en la mejora general de GNOME y con productos como RedCarpet o Evolution. Recientemente, Sun Microsystems ha tenido un papel importante en GNOME, ayudando a completar traducciones y a mejorar la accesibilidad del sistema para personas discapacitadas.

## **5.5 Mozilla**

El 23 de enero de 1998 Netscape Communications anunciaba<sup>102</sup> la distribución gratuita de su navegador web Netscape Communicator 4, hasta ese momento un producto propietario, así como la publicación de su código fuente bajo el nombre de Mozilla. Este anuncio sorprendió a muchos analistas, ya que no tenía un antecedente en la industria del software. Fue un momento de inflexión para el software libre, porque una empresa importante en aquel momento hacía una

apuesta firme por el movimiento y ponía a disposición de la comunidad un navegador de Internet en forma de software libre con las últimas tecnologías.

Era un momento complicado. Su navegador, desde su aparición en 1994, había visto disminuir su cuota de mercado desde el punto más álgido –el 70% de usuarios de Internet lo utilizaban en 1996– hasta el 57% que mantenía cuando decidió liberar el código. Netscape reaccionaba así ante la competencia de Internet Explorer, el navegador de Microsoft que la multinacional regalaba y ya comenzaba a distribuir de serie como parte de Windows, y ante el peligro de quedar relegado a una posición testimonial. Netscape esperaba que, liberando su código fuente, la comunidad ayudaría a continuar construyendo y popularizando su navegador, al más puro estilo de los primeros sistemas Unix.

La publicación del código fuente no fue un proceso inmediato, ya que Netscape sólo podía liberar la parte del código que le pertenecía y tenía que prescindir de todo el software integrado en el navegador que era propiedad de terceros. Eran más de 75 componentes y Netscape tuvo que contactar con ellos uno por uno para intentar explicarles las ventajas de su decisión. No todos estuvieron dispuestos a seguir el camino de

abrir el código y la versión que se publicó no incluyó, ni mucho menos, toda la funcionalidad de Netscape Communicator versión 4.

Aparte de publicar el código fuente debía crearse un sitio en Internet para organizar el desarrollo de todo el proyecto. Nació Mozilla.org<sup>103</sup>. El 31 de marzo de 1998 se presentaba públicamente Mozilla, una organización independiente con el objetivo de hacer de repositorio central del código del proyecto y de proporcionar las herramientas colaborativas necesarias para coordinar a los diferentes integrantes del proyecto.

Mozilla es un nombre que en realidad comprende cuatro cosas: el nombre del sitio web que contiene el proyecto, el conjunto de herramientas y librerías creadas en él, el navegador producido y distribuido en base a estas herramientas y la mascota del proyecto. La versión distribuida bajo este nombre contiene un navegador, un programa de mensajería, un programa de creación de páginas web, un cliente chat y un conjunto de herramientas de privacidad y depuración.

La publicación del código fuente y la creación de Mozilla.org levantaron muchas expectativas alrededor del proyecto. Sin embargo, en los meses siguientes se

demostró que coordinar el desarrollo de un producto tan grande era una tarea compleja y el equipo no fue capaz de producir un nuevo navegador en el período de tiempo que muchos habrían querido. Mientras, Internet Explorer iba ganando cuota de mercado y se iba consolidando como el navegador más popular en Internet.

Mozilla no tenía un trabajo fácil por delante debido a los problemas con las licencias con código de terceros. En la primera versión del código fuente del navegador liberada por Netscape, el lenguaje Java, los correctores ortográficos y los mecanismos de seguridad propiedad de RSA –imposibles de liberar en forma de código fuente bajo la legislación norteamericana vigente en aquel momento–, entre otros, no fueron publicados. De este modo, Mozilla empezaba su camino con una versión libre del navegador carente de muchos de los componentes más básicos, necesarios para tener un navegador competitivo. Los programadores de Netscape también eran conscientes de que sería necesario rediseñar el producto, especialmente el motor HTML, para poderlo adecuar a los nuevos estándares y tecnologías.

Con el tiempo se fue vertebrando una comunidad alrededor de Mozilla, con sitios web de noticias



independientes como MozillaZine<sup>104</sup> o MozillaNews<sup>105</sup>, que proporcionan información relacionada con la evolución de los proyectos de Mozilla y de terceros. También nació MozDev<sup>106</sup>, un sitio web que proporciona páginas web y recursos de colaboración gratuitos a toda clase de proyectos relacionados con Mozilla –más de sesenta. De MozDev ha salido un buen puñado de ideas y componentes que hoy en día ya se encuentran incorporados en el navegador. Pero Mozilla ha hecho más aportaciones. Algunas de sus herramientas, como Bugzilla o Bonsai, han sido adoptadas por muchos otros proyectos, y han contribuido de manera decisiva en la mejora de las infraestructuras de desarrollo en el mundo del software libre.

En noviembre del año 2000, treinta y dos meses después de la liberación del código fuente de Mozilla, Netscape publicó la versión 6.0 de su navegador, basada en el trabajo de Mozilla.org. Esta nueva versión introducía el nuevo motor de procesamiento de HTML, Gecko –un gran avance sobre las antiguas versiones 4.7 de Netscape–, así como todas las tecnologías más actuales, incluyendo herramientas para la gestión de la privacidad.

Actualmente Mozilla.org produce un conjunto de herramientas de Internet para Unix, Linux, Mac y Windows. Entre sus productos destacan FireFox, un navegador muy ligero, y ThunderBird, un cliente de correo electrónico.

En julio del 2003 se creó la fundación Mozilla<sup>107</sup> con el objetivo promover el desarrollo, la distribución y la adopción de las tecnologías desarrolladas por el proyecto. La fundación se inició con 2 millones de dólares cedidos por America Online y 300.000 dólares que contribuyó a título personal Mitch Kapor, creador de la hoja de cálculo Lotus 1-2-3 y fundador de Lotus Development<sup>108</sup>. Desde su creación ha recibido más de 500 contribuciones económicas de empresas y individuos.

## **5.6 OpenOffice.org**

Aunque el mundo del software libre ya contaba en 1999 con aplicaciones ofimáticas como Abiword, Gnumeric, o Koffice, la verdad es que todas ellas no ofrecían la funcionalidad que un usuario medio podía esperar y sus filtros de importación de documentos creados con Microsoft Office estaban lejos de ser aceptables. Para el usuario final, después del sistema operativo, el paquete ofimático es el conjunto de

programas más utilizados. Con Linux ya maduro y proyectos de escritorio como KDE y GNOME bastante avanzados, una de las últimas piezas que faltaba para completar el puzzle de una solución libre completa era un paquete ofimático.

En 1999 Sun Microsystems adquirió la compañía alemana StarDivision. Su producto estrella era Star Office, un paquete ofimático bastante maduro ya en aquella época, orientado al público tanto personal como profesional. Siguiendo los pasos de Netscape, Sun decidió liberar el código de StarOffice y crear una comunidad de software libre alrededor. Así nació OpenOffice.org, dando nombre a la comunidad que mantiene la versión libre del proyecto y a la propia comunidad. En poco tiempo, OpenOffice.org se convertía en la solución ofimática más utilizada dentro del mundo del software libre.

OpenOffice.org se encuentra disponible en más de 30 idiomas e incluye un procesador de textos, una hoja de cálculo, un programa de presentaciones, y una programa para crear gráficos. Una de las características importantes de OpenOffice es que permite abrir casi perfectamente cualquier documento creado con Microsoft Office, lo cual representa una gran ventaja para poder interoperar con usuarios de

este paquete así como para realizar migraciones. Incluye además opciones como exportar directamente documentos a formato PDF o presentaciones a formato Flash que no se encuentran en el resto de paquetes ofimáticos.

Una de las ventajas de OpenOffice.org es que sus formatos nativos de datos están basados en XML. Esto representa una gran flexibilidad para poder crear nuestras propias herramientas para procesar documentos o recuperarlos en otras aplicaciones. A finales de septiembre del 2003 aparecieron las primeras indicaciones<sup>109</sup> de que la Comunidad Europea podría estandarizar los formatos de OpenOffice.org bajo la norma ISO para su posterior adopción.

Sun, que aún sigue siendo el principal mantenedor de OpenOffice.org, produce a partir de la versión libre, usando licencia dual, el producto comercial StarOffice<sup>110</sup>. La principal diferencia respecto a OpenOffice es que incluye una base de datos personal y algunos filtros adicionales de importación y exportación de documentos. Con un precio de 79.95\$ de venta al público en el año 2004, Sun incluye

además una guía de instalación muy completa y soporte técnico.

## 5.7 Mono

Durante el año 2000 Microsoft presentó la tecnología .Net como una plataforma de software de última generación para simplificar el desarrollo de servicios web y aplicaciones de escritorio. Microsoft inició poco después el proceso de estandarización de la parte principal de la plataforma .Net en la asociación ECMA<sup>111</sup> para su formalización y aceptación como estándar. Esta asociación ha estandarizado numerosas tecnologías, entre ellas, el lenguaje Javascript.

A principios del año 2001 se inició el proyecto Mono<sup>112</sup> impulsado por la empresa Ximian, actualmente parte de Novell<sup>113</sup>, como una implementación de código abierto de la plataforma .Net de Microsoft y del estándar ECMA. El objetivo original era proporcionar una plataforma de software libre que facilitara el desarrollo de aplicaciones de escritorio en entorno Linux reduciendo el tiempo y los costes de desarrollo. Con el tiempo también se ha posicionado como un entorno que permite ejecutar en Linux aplicaciones diseñadas para Microsoft .Net en

entorno Windows, facilitando la migración de aplicaciones a Linux y aumentando su base de desarrolladores y usuarios. Actualmente unos 20 ingenieros de Novell trabajan en el proyecto Mono en el que han colaborado ya más de 300 voluntarios.

Los paquetes que componen la distribución de la plataforma Mono comprenden un compilador C# , una máquina virtual (que permite ejecutar las aplicaciones) y un conjunto de librerías de clases que proporcionan miles de funciones listas para ser utilizadas con una completa documentación en línea.

Con Mono se pueden escribir aplicaciones en múltiples lenguajes de programación, incluyendo entre ellos Python, Object Pascal, Nermele y C#. Una vez escritas, las aplicaciones se traducen a CIL (*Common Intermediate Language*), que es un lenguaje intermedio que no tiene particularidades de ninguna arquitectura. Una vez compilado en CIL, la aplicación se traduce al lenguaje específico de la arquitectura final donde será ejecutado. Este sistema permite distribuir un único programa binario para todas las arquitecturas en vez de un programa específico para cada plataforma. Pero no menos importante es la libertad de escoger cualquier lenguaje de programación o combinación de ellos y poder ejecutar

la aplicación en cualquiera de las plataformas en las que Mono se encuentra disponible, entre las que se incluyen Intel, AMD64, SPARC, StrongArm y S390x. Mono proporciona las herramientas para crear aplicaciones para Linux (diversas distribuciones), Solaris, Windows, Mac/OS y *mainframes* de IBM.

Dentro de las numerosas contribuciones por parte de voluntarios, una especialmente significativa es *Monodevelop*<sup>114</sup>, una adaptación a Mono en plataforma Linux del entorno de desarrollo de código abierto *SharpDevelop*, que integra editor de textos, depurador, gestión de proyectos, y documentación en línea. *Monodevelop* está escrito usando Mono y GTK# y en sí mismo representa un ejemplo del tipo de aplicaciones que se pueden crear con Mono de forma rápida y sencilla.

Hoy en día la plataforma Mono se está utilizando para desarrollar aplicaciones Linux de última generación. Algunas de las más significativas son *DashBoard*<sup>115</sup>, un gestor de información personal e *iFolder*<sup>116</sup>, un sofisticado sistema de gestión de archivos. También está siendo utilizado para poder ejecutar en Linux servicios web diseñados en ASP.NET bajo plataforma Windows, como por ejemplo las numerosas aplicaciones ASP.NET que ha portado a Mono la

consultora Voelcker Informatik AG para su uso en el Ayuntamiento de Munich funcionando en 350 servidores.

-----

<sup>87</sup><http://www.redhat.com>

<sup>88</sup><http://www.mandrakelinux.com/>

<sup>89</sup><http://www.novell.com>

<sup>90</sup><http://www.debian.org>

<sup>91</sup><http://www.knoppix.com/>

<sup>92</sup><http://www.uic.edu/>

<sup>93</sup><http://www.apache.org/foundation/how-it-works.html>

<sup>94</sup><http://www.uni-tuebingen.de/>

<sup>95</sup><http://www.kde.org>

<sup>96</sup><http://www.trolltech.com/>

<sup>97</sup><http://conference2004.kde.org/>

<sup>98</sup><http://www.xandros.com/>

<sup>99</sup><http://www.gnome.org>

<sup>100</sup><http://www.kde.org>

<sup>101</sup><http://www.guadec.org/>

<sup>102</sup><http://wp.netscape.com/newsref/pr/newsrelease558.html>

<sup>103</sup><http://www.mozilla.org>

<sup>104</sup><http://www.mozillazine.org/>

<sup>105</sup><http://mozillanews.org/>

<sup>106</sup><http://www.mozdev.org/>



- <sup>107</sup><http://www.mozilla.org/press/mozilla-foundation.html>
- <sup>108</sup><http://www.lotus.com>
- <sup>109</sup><http://www.tbray.org/ongoing/When/200x/2004/09/24/SmartEC>
- <sup>110</sup><http://www.sun.com/software/star/staroffice/>
- <sup>111</sup><http://www.ecma-international.org/>
- <sup>112</sup><http://www.mono-project.com>
- <sup>113</sup><http://www.novell.com>
- <sup>114</sup><http://www.monodevelop.com>
- <sup>115</sup><http://www.nat.org/dashboard/>
- <sup>116</sup><http://www.novell.com/products/ifolder/index.html>

## **6. OPORTUNIDADES DE NEGOCIO EN SOFTWARE LIBRE**

### **6.1 Introducción**

La libertad de comerciar con el software libre, de venderlo y dar servicios sobre él, es una de sus características imprescindibles. Muchas empresas e individuos aprovechan esta libertad para hacer negocios con el software libre, tanto vendiéndolo como dando servicios de formación y consultoría.

En el año 2005 el 67% de servidores web<sup>117</sup> en Internet utilizan el servidor libre Apache. El software libre ha arrinconado a los servidores basados en código propietario. Apache se desarrolla cooperativamente por voluntarios y empresas. Diariamente empresas y profesionales prestan servicios y realizan desarrollos gracias a él.

El software libre empezó a crear gran expectación en Estados Unidos en el sector privado en el año 1999 cuando Red Hat y VA Linux se convirtieron en las dos primeras empresas basadas en software libre en incorporarse al índice bursátil Nasdaq. Aunque, las

cosas han cambiado mucho desde entonces, especialmente para VA Linux, Red Hat hoy en día tiene más capitalización bursátil que algunas de las empresas tradicionales de software propietario.

Soy contrario al argumento que el software libre proporciona nuevos modelos de negocio y estoy más en la línea de pensar que el software libre es un nuevo modelo de producción de software y de entender la propiedad intelectual que crea una situación de mercado distinta a la que el software propietario ha establecido. Los modelos de negocio son los mismos que se han usado tradicionalmente en la industria del software con algún pequeño matiz.

El software libre garantiza a todo el mundo el acceso al código fuente, la posibilidad de modificarlo y consecuentemente la participación en el proceso de creación. También garantiza la posibilidad de distribución a cualquier usuario o empresa del software. Todos tenemos los mismos derechos sobre el software libre, lo que crea un entorno de libre competencia, que es beneficioso para las propias empresas y usuarios. Cualquier empresa tiene las mismas oportunidades de competir que cualquier otra persona o empresa del resto del mundo.

El software propietario tiende a crear mercados monopolistas, o oligopolistas en el mejor de los casos, debido a los derechos exclusivos que el fabricante tiene sobre el software y que se traducen en un control exclusivo sobre la mejora y distribución del propio software. En este sentido, la evidencia empírica demuestra que la mayoría de mercados horizontales en la industria del software se encuentran dominados por una o pocas empresas. Desde sistemas operativos, paquetes ofimáticos, hasta aplicaciones de diseño gráfico o aplicaciones de servidores.

En la industria de las tecnologías de la información muy pocas veces se crea una solución informática desde cero sino que se utilizan aplicaciones e infraestructuras de software ya existentes. El software libre abarata los costes de desarrollo de nuestras soluciones ya que tenemos a nuestra disposición gran cantidad de librerías y aplicaciones completas, que podemos usar para construir nuestras propias soluciones libres. Un hecho importante, si tenemos en cuenta que el aproximadamente el 75%<sup>118</sup> por ciento del software que se escribe en el mundo es de uso interno para empresas, que requiere un alto grado de personalización y donde el software libre puede proporcionar desarrollos más económicos.

## 6.2 La industria del software

El software libre ha invertido la lógica en cómo los titulares de los derechos de autor del software ceden parte de los mismos a sus usuarios. Los fabricantes de software propietario basan una parte de su fuente de ingresos en vender el derecho de uso de su software (a través de una licencia). Aunque ninguna de las licencias de software libre imponen restricciones en la venta del software<sup>119</sup>, el hecho de que el código esté disponible y que cualquiera pueda redistribuirlo hace que, efectivamente, el coste de licencia tienda a ser inexistente.

Existe la creencia errónea de pensar que la venta de licencias es una fuente extraordinaria de ingresos para cualquier empresa. Esta creencia viene impulsada por el ejemplo de las empresas de venta de software estándar empaquetado como Microsoft, Symantec, o Adobe. Este tipo de empresas, una vez que han cubierto los gastos de desarrollo y marketing del producto, incurren en unos costes marginales muy bajos, donde gran parte del precio se convierte en beneficio. Los costes fijos que tienen, desarrollo de nuevas versiones de producto principalmente, son

costes que no escalan con la venta de más unidades de producto.

La creación de software es un proceso económicamente muy costoso pero, una vez amortizado, el coste de copia es muy bajo. Esto produce el efecto de que una vez instaurado un vendedor en posición dominante en un mercado, si lo cree conveniente, pueda fácilmente vender a un precio muy bajo haciendo difícil la instauración de nuevos competidores.

Sin embargo, son relativamente pocas las empresas que pueden seguir este modelo ya que los mercados horizontales son ciertamente limitados. Conseguir escribir un producto que sea un éxito es tan difícil como conseguir escribir una novela que se convierta en un *bestseller*. En épocas de recesión económica, los ingresos por venta de licencias tienden a bajar, ya que las empresas congelan la adquisición de nuevo software. Si la situación de recesión es prolongada, los servicios suelen ser la única forma de ingresos. Las empresas con un modelo de negocio basado en servicios tienen mayor capacidad para enfrentarse a las situaciones adversas de mercado que las empresas basadas en la venta de producto que son mucho más vulnerables.

La media de ingresos por la venta de licencias en empresas que desarrollan software comercial – tanto propietario como libre - es de algo menos de un 30%<sup>12</sup> del total de su negocio, y el restante 70% viene de la prestación de servicios.

### **6.3 Modelos de negocio**

Frank Hercker escribió en 1998, mientras era empleado de la empresa Netscape, un detallado ensayo<sup>121</sup> sobre las posibilidades de negocio en el software libre. Netscape estaba considerando en aquel entonces posibles formas de conseguir ingresos una vez liberado el código fuente de su navegador, que más tarde se convertiría en el proyecto Mozilla. El ensayo es de los más completos en este área. Sin embargo, pese a ser destacable, describe muchas ideas que con el tiempo se han demostrado poco viables o aplicables por muy pocas empresas.

Vamos a enumerar los principales modelos negocio que se han puesto en práctica en el software libre con éxito durante los últimos años. Estos modelos también se han aplicado anteriormente a empresas de software no libre por lo que no los podemos considerar exclusivos del software libre.

### Empresas que venden el software como producto.

Empresas que crean programas informáticos o que participan activamente en el desarrollo de los mismos y, que usualmente, después venden a través de un canal en forma de producto. Son empresas que tienen presupuestos importantes destinados a financiar la investigación y desarrollo de estos proyectos libres, al contrario de las empresas de servicio que usualmente hacen contribuciones puntuales.

En la industria del software tenemos empresas como Sun Microsystems que desarrolla el producto Star Office a partir del proyecto libre OpenOffice.org, Novell que comercializa el cliente de correo electrónico Evolution o el producto Zen Networks que incluye el software libre RedCarpet, o IBM que incluye el entorno de desarrollo Eclipse como parte de su solución WebSphere. Aunque estos productos han sido desarrollados principalmente por ingenieros de estas compañías han contado con un apoyo muy importante de las comunidades de software libre que se han creado alrededor.

Prácticamente todas estas empresas ofrecen servicios relacionados con sus productos como pueden ser personalización, formación, soporte técnico o



integración de sus productos con sistemas de terceros, es decir, que no viven exclusivamente de la comercialización del producto.

Algunas de estas empresas ofrecen licencias de sus productos para usos no libres. Son habitualmente compañías que han impulsado proyectos de software libre y son titulares de los derechos de autor de los mismos.

Esto permite que, por ejemplo, estas empresas utilicen un sistema de licencia dual y puedan ofrecer por un lado una licencia libre a cualquiera que este dispuesto a publicar el código de sus mejoras, y una licencia no libre, previo pago lógicamente, a los que no quieran compartir sus modificaciones.

Este modelo es atractivo para las empresas porque permite ofrecer los productos de forma libre y no libre según las necesidades de cada cliente y permite obtener los beneficios de usar el método de producción del software libre. Más adelante veremos el caso de la empresa MySQL AB que explota con notable éxito este modelo.

Empresas de servicios informáticos. Empresas que se dedican a la consultoría, desarrollo a medida de

soluciones, formación y soporte técnico. Este tipo de empresas representan el área principal donde se está concentrando el desarrollo de negocio en software libre en España.

Su valor diferencial respecto a las empresas tradicionales de servicios son los beneficios que transmiten a sus clientes por el hecho de trabajar con tecnologías libres – como acceso al código fuente de las soluciones – . Es cada vez más común encontrar administraciones públicas y empresas que exigen el uso de tecnologías libres en el desarrollo de sus soluciones.

En general, las empresas que mejor funcionan de este tipo son aquellas que se especializan en un área concreta de conocimiento, como pueda ser la creación de gestores de contenidos con software libre o de servicios web. Ser un especialista en un área y ser reconocido como experto en la misma es una buena estrategia. De hecho, ser desarrollador o contribuidor de software libre es una de las mejores tarjetas de presentación que podemos tener si nos dedicamos a dar servicios basados en software libre.

Es común que las empresas de servicios tengan que desarrollar soluciones a medida. El mundo del

software libre ofrece un gran abanico de programas informáticos, librerías, y ejemplos de coste muy bajo que puede acelerar enormemente la creación de soluciones competitivas.

Según LibroBlanco.com<sup>122</sup>, una iniciativa que hace seguimiento del software libre en el ámbito del sector público, a julio de 2004 había 289 empresas en España contabilizadas que trabajan y dan servicios basados en software libre. Las tres comunidades con una concentración más alta de empresas de software libre son Madrid con 68, Cataluña con 46, y Andalucía con 40.

Vendedores de hardware. Son empresas que se centran su negocio en la venta de hardware y para las cuales el software es un complemento necesario de su producto.

Algunos ejemplos son Cobalt (adquirida por Sun Microsystems) que vende servidores con software Linux altamente personalizado o empresas como Sharp que vende el organizador personal de Sharp Zaurus<sup>123</sup> que usa Linux, y otros programas libres como base de su software de usuario.

En este sentido, a finales del año 2003, representantes de los gobiernos, universidades, y empresas de China,

Japón, Corea firmaron un acuerdo para desarrollar conjuntamente un sistema operativo basado en Linux para el mercado asiático, que fuera una alternativa al sistema Windows de Microsoft. Entre los motivos principales, destaca la independencia y la liberación de pago de royalties de software que representa para el importante sector asiático de fabricantes de electrónica de consumo. Durante el verano del 2004 se presentó<sup>124</sup> la versión 1.0 de Asianux, y los diferentes miembros del consorcio asiático han iniciado su proceso de adopción.

Todo esto no es sorprendente si tenemos en cuenta que, ya en los años 70, Japón, mediante empresas como Hitachi, Fujitsu o NEC, compitió de forma agresiva en la creación de hardware y software (incluyendo sistemas operativos) en el mercado de los *mainframes* o en los años 90 en la industria del entretenimiento, donde se ha convertido en un líder indiscutible en el desarrollo de software de videojuegos.

## **6.4 Experiencias fallidas**

Suele haber un cierto tabú cuando se habla de las experiencias fallidas en el mundo del software libre. La verdad es que de cualquier experiencia siempre hay

algo que se aprende y una parte positiva y es interesante conocer estas experiencias, cómo se enfocaron e intentar entender por qué no tuvieron éxito. Es importante reconocer el riesgo que tomaron porque, aunque fallidas, fueron innovadoras al intentar poner en marcha experiencias de negocio que antes no se habían intentado.

## **SourceGear**

La empresa SourceGear en 1998 empezó a trabajar en el desarrollo del procesador de textos libre Abiword<sup>125</sup>. La idea de SourceGear era crear un paquete ofimático completo de software libre para Linux, Windows y Mac. Esto fue antes de que Sun Microsystems hubiera liberado OpenOffice.org por lo que no había un paquete ofimático libre que liderara el mercado. Empezaron escribiendo Abiword, el procesador de textos, que fue el único producto que llegaron a completar.

SourceGear confiaba en el crecimiento de Linux de forma abrumadora como plataforma en el escritorio y en construir una comunidad alrededor del proyecto que les ayudara a crear su producto de forma más rápida, disminuyendo los costes de desarrollo del mismo. Desgraciadamente Linux creció en el

escritorio a un ritmo mucho más lento que el esperado y, aunque consiguieron construir una comunidad alrededor de su producto, el desarrollo les llevó mucho más tiempo y esfuerzo del que esperaban, algo de hecho común en muchos de proyectos de software.

Eric Sink, fundador de SourceGear, explica<sup>126</sup> cómo su empresa fracasó al intentar posicionar Abiword como solución en diferentes nichos de mercado. El error principal de SourceGear fue intentar obtener beneficios de la forma que lo han hecho tradicionalmente los vendedores de software propietario. Después de más de dos años de desarrollo y ante la imposibilidad de generar beneficios que consiguieran pagar el coste de desarrollo del producto, SourceGear abandonó la idea de Abiword y se reorganizó como empresa de software propietario. Actualmente facilita soluciones de control de versión de código con un éxito notable.

SourceGear cedió el proyecto Abiword a la comunidad de software libre. El servidor del proyecto fue acogido en una universidad y hoy en día continúa su desarrollo por un grupo de voluntarios.

## Eazel

Eazel fue un *startup* fundado por algunos de los ingenieros que trabajaron en el desarrollo del sistema Mac, y por ex-empleados de Netscape, Be y Sun Microsystems. Entre los fundadores figuraba Andy Hertzfeld<sup>127</sup>, que implementó una parte muy importante de la interficie gráfica del primer Mac. Se marcaron como objetivo conseguir que Linux fuera tan sencillo de usar como lo eran los sistemas Mac. La empresa gastó aproximadamente 13 millones de dólares y llegó a tener 70 empleados en plantilla.

Eazel generó muchas expectativas ya que contaba con ingenieros de primera línea, con una gran cantidad de proyectos de éxito a su espalda, y con un nivel de financiación que era envidiable para el resto de industria del software libre del momento.

Sin embargo la aventura duró seis meses. El único producto que diseñaron fue Nautilus, un completo gestor de archivos que facilita la navegación de los recursos del sistema: ficheros, carpetas y dispositivos de hardware. Nautilus era la herramienta que permitía establecer el negocio, era software libre y se ofrecía de forma gratuita.

La visión de Eazel era convertir el gestor de archivos en algo similar a un navegador que permitiera ofrecer todo un abanico de servicios de pago a sus usuarios. El modelo de negocio de Eazel estaba centrado en la venta de servicios por suscripción vía web que tendrían un coste de unos 4 dólares mensuales y que incluirían servicios de copias de seguridad de los archivos de los usuarios y de resolución de problemas de software. Sin embargo nunca llegó a acumular suficientes usuarios para hacer la empresa mínimamente rentable.

Eazel cerró sus puertas en mayo de 2001<sup>128</sup>. Nautilus se convirtió poco después en el gestor de archivos del escritorio GNOME y hoy en día es mantenido por una comunidad de voluntarios.

Tanto del caso de Eazel como del caso de SourceGear es destacable el hecho que, aunque las empresas han abandonado el desarrollo del software, el código fuente del mismo ha sobrevivido a ambas aventuras empresariales y ha logrado continuar gracias a las comunidades de voluntarios que se ha construido a su alrededor y han impulsado su desarrollo.



## 6.5 Experiencias de éxito

Existen en el mundo miles de empresas y profesionales que se ganan la vida cada día desarrollando, instalando, formando, y soportando software libre. Todos ellos son casos de éxito pero son casi imposibles de documentar. Veremos cuatro casos concretos de empresas que están usando el software libre como parte de su estrategia de negocio o incluso están totalmente basados en él.

### **Red Hat**

Ha sido una de las empresas pioneras en el mundo del software libre. En 1993 Marc Ewing y Bob Young formaron Red Hat Software. Empezaron en casa de Bob Young duplicando la distribución que ellos mismos habían empaquetado y vendiéndola a un precio de 50 dólares. Pocos años más tarde entrarían a cotizar en el índice bursátil tecnológico del Nasdaq.

En Estados Unidos se le ha dado mucha importancia a Red Hat. Frecuentemente sus éxitos o fracasos se han interpretado como la viabilidad global económica del software libre. Vincular el éxito de una empresa o grupo de empresas al hecho de que el software libre sobreviva sería tan absurdo como hace unos años

afirmar que Internet dependía del éxito de las puntocom para asegurar su continuidad. La diversidad de intereses de quienes trabajan con y por el software libre y su fuerte tradición académica aseguran una viabilidad más allá del éxito de un proyecto concreto.

Red Hat ha intentando siempre devolver parte del beneficio que obtiene gracias al software libre a la propia comunidad. Cuando en 1999 salió a bolsa en Estados Unidos repartió acciones de la empresa (que lógicamente luego se pueden vender) entre unos 5.000 programadores de diferentes proyectos de software libre. Red Hat también tiene en su plantilla desarrolladores muy destacados de los principales proyectos libres, incluyendo el kernel de Linux, GNOME, o las herramientas GNU.

Red Hat ha basado su negocio en cuatro grandes áreas:

Formación. Proporciona servicios de formación para usuarios, administradores de sistemas y desarrolladores. Los cursos cuestan a partir de unos 100 dólares el más barato y entre las materias que se imparten se encuentran: seguridad, desarrollo de kernel y sistemas empotrados, gestión avanzada de redes o desarrollo web. Su certificación *Red Hat*

*Certified Engineer* (RHCE) se inició en 1999 y se ha convertido en un estándar reconocido en el mercado que poseen más de 10.000 profesionales.

Soporte. Los servicios de soporte es un área donde Red Hat ha invertido muchos recursos, creando manuales detallados de instalación y uso en diferentes idiomas, una base de datos con incidencias comunes y sus soluciones, listas de hardware compatible con su distribución, y un servicio técnico disponible vía telefónica en varios idiomas durante los 365 días del año. Además, Red Hat también ofrece soporte para desarrolladores que creen sus propias aplicaciones sobre esta plataforma.

Servicios de ingeniería. Ya desde sus inicios, los servicios de ingeniería fueron una parte importante de sus ingresos con acuerdos con empresas del tamaño de Intel para hacer desarrollos o adaptaciones de software libre. Entre los clientes de Red Hat figuran algunas de las 100 empresas más importantes de Estados Unidos.

Venta de software. Red Hat vende software a través del canal para el usuario final a bajo precio y versiones para empresas que incluyen productos de software propietario desarrollados por terceros.

Los productos que proporciona Red Hat y la organización de su oferta de servicios es bastante similar a de otros distribuidores de Linux, como la empresa alemana Suse (actualmente parte de Novell) o la francesa Mandrake Software.

## **IBM**

A finales de 1997 IBM<sup>129</sup>, estaba considerando cómo reorganizar su oferta de servicios web. Con menos de 1% del mercado de servidores web en 1998 decidió reemplazar su servidor web propio por el servidor web de código libre Apache. IBM entró a formar parte de la fundación Apache y desde entonces ha dedicado de forma continuada dinero y recursos al desarrollo y mejora del proyecto. Además, IBM como parte de su oferta, también ofrece a sus clientes servicios de soporte y personalización basados en Apache.

Pero la apuesta principal de IBM ha sido con Linux. La multinacional inició en año 2001 la adaptación del sistema Linux a todos sus sistemas, incluyendo los *mainframe* de gama alta. Hoy en día Linux es el único sistema operativo que se encuentra disponible para todos los sistemas que comercializa la compañía. Además, IBM ha adaptado sus aplicaciones más

populares a Linux, incluyendo la base de datos DB2 y los servidores de la plataforma Notes.

Aunque IBM es uno de los mayores poseedores de patentes del mundo, ha ganado una cierta buena reputación entre los más escépticos con el caso SCO, el cual ha sido, pese a su inconsistencia, una de las peores amenazas jurídicas para Linux de los últimos tiempos. SCO, a principios del 2003, empezó a demandar a empresas usuarias de Linux, entre ellas IBM, por una supuesta violación de propiedad intelectual. IBM ha demostrado que está al lado del software libre y de defender Linux delante de los tribunales.

La multinacional, al usar y participar en la creación de software libre, se ha posicionado con uno de los grandes proveedores de tecnologías libres para grandes corporaciones. Ofrece sus soluciones basadas en Linux como alternativas a los servidores de Sun Microsystems y de Microsoft sobre plataforma Intel, lo que crea un camino de migración para sus clientes hacía hardware de IBM de gama alta.

IBM también ha cedido algunos de sus desarrollos a la comunidad de software libre, como por ejemplo el conjunto de herramientas Eclipse, un entorno para

programadores, donde ha invertido 40 millones de dólares<sup>130</sup>.

## Ándago

Ándago<sup>131</sup> es una consultora con sede en Madrid especializada en proyectos web para empresas e instituciones públicas centrada en tecnologías libres y especialmente en Linux. En el año 2001, recibió una fuerte inversión del grupo financiero Talde. Tuvo un crecimiento muy fuerte durante el año 2001 cuando llegaron a ser medio centenar de empleados. El retraso en la implantación de tecnologías libres por parte de la administración pública mermó las expectativas de crecimiento de la compañía.

En la trayectoria de Ándago destacan las dos importantes implantaciones de Linux realizadas en la administración pública: el Virtual Map, basada en la conexión de todas las sedes del Ministerio de Administraciones Públicas, y la primera versión del proyecto LinEx<sup>132</sup>, la distribución de Linux adaptada para su divulgación a toda la ciudadanía extremeña.

Ándago ha desarrollado plataformas tecnologías Linux que se ha especializado en adaptar a sus clientes, entre los que se incluyen operadores del

sector de las telecomunicaciones y la administración pública.

## **MySQL AB**

MySQL AB<sup>133</sup> desarrolla una familia de soluciones de bases de datos de alto rendimiento. Su producto principal es el servidor MySQL que junto con Linux, Apache, PHP y Perl, se ha convertido en una de las herramientas más populares para crear sitios web en Internet y es la base de datos libre más popular.

La empresa fue fundada en el año 2001 por dos suizos y un finlandés, y ha sido rentable desde sus inicios. En octubre de 2001 obtuvo financiación de un fondo de capital riesgo que ha utilizado para crecer ordenadamente.

MySQL AB es el poseedor único de los derechos de autor de sus productos. Utiliza el sistema de licencia dual y ofrece su producto bajo una licencia GPL y otra propietaria. Los usuarios pueden descargar el software, usarlo libremente y modificarlo, integrar y distribuir estas mejoras. Sin embargo, los usuarios de la versión libre deben seguir las reglas de la licencia GPL que estipulan. Si redistribuyen una solución

basada en MySQL el código de fuente completo de la solución también debe estar abierto y disponible para la redistribución. Los clientes que utilicen MySQL como parte de una solución de software o hardware propietario y no quieran proveer el código fuente pueden comprar una licencia propietaria del producto a partir de 500 dólares.

Los beneficios de MySQL AB proceden principalmente de la venta de servicios, soporte técnico, y licencias comerciales de su producto. Estos ingresos se utilizan para continuar desarrollando el producto libre MySQL.

-----

<sup>117</sup> [http://news.netcraft.com/archives/web\\_server\\_survey.html](http://news.netcraft.com/archives/web_server_survey.html)

<sup>118</sup> <http://www.opensource.org/advocacy/jobs.html>

<sup>119</sup> <http://www.gnu.org/licenses/gpl-faq.html#DoesTheGPLAllowMoney>

<sup>120</sup> Ver libro *Secrets of Software Success: Management Insights from 100 Software Firms Around the World* escrito por empleados de la consultora McKinsey en Alemania.

<sup>121</sup> <http://www.hecker.org/writings/setting-up-shop.html>

<sup>122</sup> <http://www.libroblanco.com/>

<sup>123</sup> <http://www.myzaurus.com/>



<sup>124</sup>[http://news.com.com/Asian+Linux+gaining+momentum/21001011\\_3-5278304.html](http://news.com.com/Asian+Linux+gaining+momentum/21001011_3-5278304.html)

<sup>125</sup><http://www.abiword.com>

<sup>126</sup><http://software.ericssink.com/20011228.html>

<sup>127</sup>[http://en.wikipedia.org/wiki/Andy\\_Hertzfeld](http://en.wikipedia.org/wiki/Andy_Hertzfeld)

<sup>128</sup><http://mail.gnome.org/archives/gnome-hackers/2001May/msg00203.html>

<sup>129</sup><http://www.ibm.com>

<sup>130</sup>[http://news.zdnet.com/2100-3513\\_22-275388.html](http://news.zdnet.com/2100-3513_22-275388.html)

<sup>131</sup><http://www.andago.com>

<sup>132</sup><http://www.linex.com>

<sup>133</sup><http://www.mysql.com/>

## **7. SOFTWARE LIBRE EN EL SECTOR PÚBLICO**

### **7.1 La importancia del software libre en la administración**

La administración pública tiene una responsabilidad y unos intereses muchos más amplios que una empresa o un particular cuando adquiere tecnologías de la información. Debe tener una vocación de consumo responsable porque está operando con dinero público. El ahorro que representa el software libre es atractivo para las administraciones ya que les permite la mejor administración de sus recursos. El proyecto para educación LinEx<sup>134</sup> permitió a la Junta de Extremadura ahorrar 60 millones de euros en licencias de software que pudieron reinvertir en hardware y en conseguir un ordenador para cada dos alumnos. A medida que pasan los años y la sociedad de la información avanza, el gasto en tecnologías de la información será mayor y mayores serán aún los ahorros en coste de licencia que el software libre representa.

Con la popularización de Internet, las comunicaciones electrónicas entre ciudadanos y administraciones se han incrementando enormemente y hoy es posible realizar trámites administrativos vía Internet. La administración debe dar un servicio universal y, por lo tanto, respetar e impulsar los estándares abiertos cuando se dirige a sus ciudadanos. No hacerlo supone favorecer a un determinado fabricante y a sus clientes, discriminando al resto de usuarios. Sería similar a si la televisión pública sólo la pudiéramos ver usando televisores de un fabricante determinado.

Un caso que hoy en día es aún frecuente son los sitios web de administraciones que sólo se pueden visualizar correctamente con Internet Explorer de Microsoft. ¿Qué sucede con los usuarios de otros navegadores? o peor aún, ¿qué sucede con los usuarios de plataformas donde no existe ese navegador – como Linux o Unix - ? El software libre, que soporta una parte importantísima de la infraestructura de Internet y ha abanderado los estándares desde sus inicios, asegura la no exclusión de otros programas o sistemas.

El gobierno también tiene la responsabilidad de garantizar la privacidad de los datos de los usuarios. Sin acceso al código fuente es imposible saber qué hace una aplicación con estos datos y cómo son

tratados. El software libre ofrece una transparencia total y permite a los gobiernos hacer sus propias auditorías de seguridad sobre el software. La empresa Microsoft, consciente de esta preocupación por parte de gobiernos, lanzó la iniciativa *Government Security Program*<sup>135</sup>. Gracias a ella los gobiernos, previa firma de un acuerdo que tiene una duración de tres años, pueden tener acceso a parte del código fuente de dos de las aplicaciones de Microsoft: el sistema operativo Windows y el paquete ofimático Microsoft Office. Esta iniciativa presenta varias limitaciones respecto al software libre. Sólo nos ofrece algunos productos de la compañía, debemos firmar un acuerdo, no podemos mejorarlo y no podemos distribuir las mejoras. Se trata en el fondo más de una operación de marketing que de una acción que busque añadir transparencia a la explotación de las soluciones Microsoft.

Cuando se piensa en la sociedad de la información y en acercar las nuevas tecnologías a todo el mundo, no sólo hay que pensar en los costes de las infraestructuras de comunicación y hardware, sino también en los costes del software. El precio del software propietario para un ordenador personal para poder acceder a la sociedad de la información oscila entre 400 y 800 euros<sup>136</sup> por PC, una suma totalmente inalcanzable para las regiones más desfavorecidas del

mundo y que conduce a la copia ilegal o a la fractura digital.

Por ejemplo, un ciudadano de Vietnam, donde la renta per cápita se sitúa en los 2.100 dólares al año<sup>137</sup> debería dedicar su sueldo entero durante seis meses a poder pagar una licencia de Microsoft Office y Windows XP Professional. No es de extrañar que el índice de copias ilegales en el país sea del 97%<sup>138</sup>. Y como Vietnam, muchos países en vías de desarrollo. Microsoft para frenar la migración de los países en desarrollo a Linux, ha creado una versión del sistema Windows, llamada Starter Edition, por un precio aproximado de 36 dólares<sup>139</sup>. La versión tiene limitaciones y prestaciones recortadas: puede ejecutar un máximo de tres programas simultáneamente, no tiene opciones para redes caseras y no funciona en Alta resolución gráfica.

Otro dato interesante es que en Vietnam se hablan más de 93 lenguas y una gran variedad de dialectos<sup>140</sup>. El paquete ofimático propietario más extendido, Microsoft Office, sólo se encuentra traducido a una de esas lenguas y no hay planes para adaptarlo a ninguna otra. Su principal competidor, OpenOffice.org, puede ser traducido a cualquiera de esas lenguas gracias a

que se trata de software libre. Existe actualmente un proyecto del Gobierno vietnamita para traducir el paquete ofimático OpenOffice.org y usarlo en diferentes departamentos gubernamentales. El software libre es probablemente la única opción que van a tener muchos países en vías de desarrollo para sumarse a las nuevas tecnologías.

Diferentes administraciones del mundo necesitan el mismo tipo de aplicaciones informáticas que son desarrolladas una y otra vez en diferentes lugares. El software libre permite a los gobiernos locales el intercambio de software y experiencias y ofrece un marco donde pueden compartir los gastos de desarrollo de software.

La Junta de Extremadura ha firmado un acuerdo<sup>141</sup> con la Junta de Andalucía para que ésta utilice la tecnología que generó el proyecto extremeño Linex, y otro con el gobierno de Colombia, que está desarrollando el Proyecto LinExCol<sup>142</sup> (LinEx de Extremadura para Colombia), una distribución creada por Fundehumano con el apoyo de la Junta de Extremadura. Recientemente, el Ayuntamiento de Barcelona y el de Porto Alegre firmaron un acuerdo<sup>143</sup> para intercambiar experiencias relacionadas con el

software libre que están extendiendo a otras administraciones.

## 7.2 Iniciativas legales

China siempre ha sido reticente a subcontratar proyectos tecnológicos a las grandes multinacionales<sup>144</sup> y ha tenido una cierta predisposición a la creación de una industria del software propia. El Gobierno chino financia desde hace varios años la distribución RedFlag de Linux<sup>145</sup> y trabaja en un gran número de proyectos gubernamentales con empresas locales y software libre. Diferentes países, como Francia, Brasil, Argentina o Perú<sup>146</sup> han llevado a trámite leyes para favorecer el uso del software libre en la Administración pública.

El Parlamento Europeo adoptó recientemente una resolución que insta a los países miembros a promocionar los proyectos de código abierto o software libre<sup>147</sup>. En España ha habido una única proposición de ley<sup>148</sup>, de Esquerra Republicana de Catalunya (ERC), que fue rechazada en el Congreso de los Diputados<sup>149</sup> en septiembre de 2002. Esta misma proposición de ley fue también presentada por ERC en el Parlamento de Cataluña y rechazada el 25

de octubre de 2002. En el primer caso tuvo los votos en contra del Partido Popular, y en el caso catalán tuvo los votos en contra del Partido Popular y de Convergència i Unió.

La propuesta de ley de ERC instaba al Gobierno y a las empresas participadas por el mismo a utilizar software libre con la única excepción de los casos en donde no existieran soluciones, en los que podría utilizarse software propietario. Esta propuesta también reclamaba al Gobierno que realizara tareas de difusión y de cooperación con experiencias relacionadas con el software libre en España y en otros países.

También ha habido un gran número de mociones a favor del software libre, como la de Sant Bartomeu del Grau en Cataluña<sup>150</sup>, la presentada por el grupo socialista<sup>151</sup> en el Parlamento navarro instando al Gobierno de Navarra a poner en marcha un plan de despliegue del uso de software libre en el ámbito de la Administración, o la presentada por Coalición Canaria<sup>152</sup> en el Parlamento de Canarias favorable al uso del software libre. Sin embargo, la mayoría de estas mociones y propuestas no se han materializado en hechos concretos por la falta de partidas presupuestarias y de planes específicos de ejecución.



### 7.3 Caso Universidad de Lleida

La Universidad de Lleida<sup>153</sup> (UDL) es la universidad pública pionera en el uso de software libre en Cataluña. Durante años ha utilizado sistemas libres con resultados excelentes. En el 2003 accedió al rectorado un nuevo equipo que tenía como uno de sus objetivos potenciar el uso del software libre y que inició un proceso de migración de todos los sistemas. Durante el año 2003 el gasto en licencias de software en la UDL ascendió a 107.185 euros<sup>154</sup>. Entre los numerosos objetivos de la migración destacaba reducir este gasto. El proceso está previsto que finalicé en el año 2007 y se ha dividido en varias fases que se están llevando a término simultáneamente:

Migración de la infraestructura. Contempla la migración de todos los equipos que forman la infraestructura de la universidad (servidores de correo, nombres, web, etc.). Todos los nuevos proyectos que se abordan tienen que ser, si es técnicamente viable, desplegados e implementados empleando exclusivamente software libre. El nivel de implantación del software libre en esta área a finales

de 2004 era ya muy alto, próximo al 95% de los sistemas existentes en toda la universidad.

Migración del *BackOffice*. El *BackOffice*, o lo que podemos llamar software de gestión, presenta unas dificultades particulares que lo convierten en una de las fases técnicamente más difíciles de la migración, pero a su vez es el área donde los beneficios derivados del uso del software libre son más importantes. En la UDL el *BackOffice* es bastante extenso e incluye programas de gestión académica, gestión de investigación, recursos humanos, contabilidad, y unas cuantas aplicaciones desarrolladas a medida.

Los proyectos que se han iniciado en los últimos años se han llevado a cabo exclusivamente empleando software libre, incluyendo el proyecto estratégico más importante de la universidad: el campus virtual. Sólo en aquellos casos en que el proyecto depende de software procedente de terceros y en el que no hay alternativa libre se está utilizando software propietario. Debido a los largos ciclos de vida que presenta el *BackOffice*, donde un programa puede tener una vida de hasta 15 años, el proceso de cambio y migración es largo.

Migración de los escritorios. Se ha iniciado un proceso progresivo de cambio de los sistemas operativos del escritorio y de Microsoft Office por alternativas libres. Dadas las diferencias existentes, tanto a nivel técnico como nivel organizativo, entre los escritorios del personal docente e investigador y el del personal de administración y servicios, se ha optado por migrar los dos colectivos de forma separada (pese a que las iniciativas que puedan ser comunes serán llevadas a término de forma conjunta).

Para el personal docente e investigador se ha establecido un proyecto de migración voluntario. Las acciones concretas para reforzar y apoyar a los que deseen migrar a software libre se basan en establecer un mecanismo de apoyo para encontrar y evaluar las alternativas libres existentes al software científico y de investigación y crear un repositorio con todas estas alternativas.

Para el Personal de Administración y Servicios (PAS) se ha establecido un programa de migración progresiva de los ordenadores de escritorio a Linux/OpenOffice.org. Este programa seguirá dos caminos simultáneos: por una parte los equipos nuevos que se vayan adquiriendo serán instalados con Linux como sistema operativo y con OpenOffice.org

como software ofimático y por otra parte se irá instalando progresivamente OpenOffice.org en los equipos antiguos restantes.

Terminales públicos, aulas y bibliotecas. Un gran número de equipos informáticos de la UDL se encuentran en aulas de usuarios y aulas de informática, en bibliotecas, etc. donde son empleados por los estudiantes de la UDL. Se migrarán todos a Linux con OpenOffice.org.

#### **7.4 Caso de Extremadura**

En el año 1999 la Junta de Extremadura inició el Proyecto Global de Sociedad de la Información, que incluye un paquete de acciones en el área de Sociedad de la Información: la difusión de Internet, el desarrollo de la administración electrónica, la educación en red, el apoyo a las pequeñas y medianas empresas en prácticas de la nueva economía e incorporación a la Sociedad de la Información y el apoyo e integración de las zonas más desfavorecidas de la región y grupos socialmente desfavorecidos.

Como parte de este proyecto nació LinEx<sup>155</sup> con el objetivo de proporcionar una distribución Linux,

basada en Debian, personalizada a las necesidades del sector de la educación. LinEx se convirtió en pocos meses en uno de los casos de uso de Linux de más éxito en España. Esta versión personalizada de Linux se ha distribuido a través de los centros de enseñanza secundaria, la Red Tecnológica Educativa y la población en general mediante el Plan de Alfabetización Tecnológica (PAT).

Gracias al despliegue de una solución basada en software libre, la Junta de Extremadura calcula que ha conseguido un ahorro directo de 48.000 euros por cada unidad docente, que está compuesta por 22 ordenadores<sup>156</sup>. Se trata de un coste nada despreciable, si tenemos en cuenta que el software se ha instalado en más de 60.000 ordenadores en lugar del correspondiente software de propiedad. Por su parte la Junta ha invertido unos 300.000 euros en el desarrollo y mantenimiento del proyecto Linex.

Dentro del proyecto Vivernet<sup>157</sup>, que tiene como objetivo fomentar empresas en las nuevas tecnologías, en colaboración con diversas empresas extremeñas, se liberó bajo licencia libre el paquete de gestión Gestionlinex y la herramienta de gestión contable Contalinex. Ambas aplicaciones están siendo promovidas entre las empresas extremeñas ante las

alternativas de software propietario y ya han sido adoptadas por diferentes empresas de la región.

## **7.5 Caso de Brasil**

El Gobierno brasileño anunció en el año 2003 que utilizaría software libre en prácticamente todas las áreas de su Gobierno, por lo que se convertirá en el primer ejecutivo a escala mundial en llevar a cabo un despliegue masivo de este sistema en la Administración Pública.

En Brasil, la industria informática mueve más de 3.000 millones de dólares al año, de los que aproximadamente 1.000 van a parar a Estados Unidos en concepto de licencias de software<sup>158</sup>. El Gobierno de Ignacio Lula da Silva apuesta por la creación de un tejido local de empresas que ofrezcan soluciones y servicios en software libre, para fomentar, así, el desarrollo local y desvincular la inversión pública de las grandes multinacionales.

En diciembre de 2002, el estado de Rio Grande do Sul fue el primero de Brasil en aprobar una ley a favor del uso del software libre en la Administración Pública. Ya hace tiempo que los más de tres millones y medio

de clientes del banco estatal de Rio Grande do Sul se encuentran el pingüino (la mascota del sistema operativo Linux) cada vez que realizan una operación con los cajeros automáticos, y es que en este estado las empresas públicas y universidades empiezan a funcionar exclusivamente con software libre.

Brasil tiene un 22%<sup>159</sup> de la población que vive en la pobreza. Además tiene una gran extensión geográfica, lo que dificulta la comunicación entre diferentes lugares y crea comunidades que pueden quedar potencialmente aisladas. Estos factores combinados motivaron al gobierno a iniciar en paralelo diversos proyectos de inclusión digital en diferentes zonas basados en el establecimiento de telecentros. El objetivo de los mismos es reducir la exclusión digital, aumentar la capacidad profesional, difundir software libre y aumentar la participación popular en las nuevas tecnologías. El software libre ha jugado un papel fundamental en la creación de estos centros y está permitiendo que los nuevos incluidos digitales puedan hacerlo con software legal. Los responsables de los centros crearon una distribución Linux llamada Sacix<sup>160</sup>, basada en Debian que incluye de forma integrada todos los programas libres que se necesitan en estos centros y que además distribuyen en la población.

Un parte importante de la responsabilidad del avance de este movimiento en Brasil recae en la iniciativa "Software libre"<sup>161</sup>, que impulsa de un tiempo a esta parte el uso del software libre en la Administración pública y ha venido organizando el Foro Internacional de Software Libre, que en el año 2004 celebró su quinta edición. El proyecto, de carácter no gubernamental, trabaja juntamente con la alcaldía de Porto Alegre, el Gobierno federal y otras secciones de la Administración pública de Brasil para la expansión del software libre en las diferentes áreas del Gobierno.

-----

<sup>134</sup><http://www.linex.org>

<sup>135</sup><http://www.microsoft.com/resources/sharedsource/Licensing/GSP>.

px

<sup>136</sup>Coste de un Windows XP y un Microsoft Office dependiendo de si la versión es profesional.

<sup>137</sup><http://www.cia.gov/cia/publications/factbook/geos/vm.html>

<sup>138</sup><http://global.bsa.org/usa/press/newsreleases/2001-05-21.566.phtml>

<sup>139</sup>[http://www.newsfactor.com/story.xhtml?story\\_title=Budget\\_Versi](http://www.newsfactor.com/story.xhtml?story_title=Budget_Versi)  
o of\_Windows\_Headed\_for\_Russia&story\_id=27219

<sup>140</sup>[http://www.ethnologue.com/show\\_country.asp?name=Viet+Nam](http://www.ethnologue.com/show_country.asp?name=Viet+Nam)

<sup>141</sup><http://www.hispalinux.es/noticias/172>



<sup>142</sup><http://www.fundehumano.org/modules.php?name=linexcol>

<sup>143</sup><http://www.noticiasdot.com/publicaciones/2004/0604/0406/noticias040604/noticias040604-16.htm>

<sup>144</sup><http://news.bbc.co.uk/2/hi/science/nature/1749441.stm>

<sup>145</sup><http://www.redflag-linux.com/eindex.html>

<sup>146</sup>Ved propuesta en Francia,

<<http://www.senat.fr/consult/loglibre/texteloi.htm>>, y Argentina, <<http://www.grulic.org.ar/proposicion/proyecto/ley-dragan/index.html>>.

<sup>147</sup>[http://www.cyberrights.org/interception/echelon/European\\_parliament\\_resolution.htm](http://www.cyberrights.org/interception/echelon/European_parliament_resolution.htm)

<sup>148</sup>Ved documento "Publicaciones VII Legislatura: BOCG del Congreso"; serie: B, número: 244-1, fecha: 24 de mayo de 2002, en <<http://www.senado.es/>>.

<sup>149</sup>[http://www.congreso.es/public\\_oficiales/L7/CONG/DS/PL/PL\\_188.PDF](http://www.congreso.es/public_oficiales/L7/CONG/DS/PL/PL_188.PDF)

<sup>150</sup><http://www.softcatala.org/admpub/grau.htm>

<sup>151</sup><http://www.parlamento->

[navarra.es/castellano/boletines/2002/b2002111.pdf](http://www.parlamento-)

<sup>152</sup><http://www.parcn.es/pub/Bop/5L/2001/122/bo122.pdf>

<sup>153</sup><http://www.udl.es>

<sup>154</sup><http://www.udl.es/rectorat/viti/guia.html>

<sup>155</sup><http://www.linex.org/>

<sup>156</sup>Ver el artículo "Software libre en Extremadura: LinEx", publicado en el número 162 de la revista *Novatica*.

<sup>157</sup><http://www.vivernet.com/i>

<sup>158</sup><http://www.softwarelivre.org>

<sup>159</sup><http://www.cia.gov/cia/publications/factbook/geos/br.html>

<sup>160</sup><http://www.telecentros.sp.gov.br/interna.php?id=1612>

<sup>161</sup><http://www.softwarelivre>

## 8. A MODO DE CONCLUSIÓN

No me gustaría acabar sin hacer una pequeña conclusión sobre el software libre, su impacto en nuestro entorno, y su futuro más inmediato. Hemos visto cómo el software libre ya ha cambiado la forma de entender la industria de la programación. Empresas y administraciones empiezan a invertir en soluciones basadas en software libre más personalizado y mejor adaptado a sus necesidades. Cada día miles de empresas y profesionales se ganan la vida desarrollando, instalando, formando y soportando programas libres y su número irá indudablemente en aumento.

La idea de colectivizar los derechos de autor introducida por el software libre ha traspasado ya la programación y ha sido adoptada por creadores de contenidos digitales, como música, vídeo o libros con iniciativas como *Creative Commons*.

El sector público ha entendido que la libertad y el control que proporciona el software libre para realizar modificaciones, distribuirlas y desvincularse de un único proveedor que son atractivas para cualquier Administración, que bajo ningún concepto debería aceptar soluciones tecnológicas para las que sólo

existiera un único proveedor. Con la llegada de Internet y los teléfonos móviles, el sector público aumentará en los próximos años su inversión tecnológica de forma muy considerable, por lo que cobrarán aún mucha más importancia las políticas relacionadas con la adquisición de nuevas tecnologías por parte de la Administración. El software libre representa una oportunidad sin precedentes para el desarrollo de la sociedad de la información. Los ejemplos de administraciones públicas que empiezan a combinar software de propiedad con libre o a usar software libre exclusivamente, como los casos de Brasil o Extremadura, serán cada vez más frecuentes.

El modelo de producción del software libre ha demostrado tener un éxito considerable y ser altamente productivo en el desarrollo de programas tan complejos como el sistema operativo Linux o servidor Apache. Veremos cómo este modelo de producción cooperativo se extiende a otras áreas del conocimiento como ya está sucediendo actualmente con proyectos como Wikipedia o Open Directory.

Con la mejora en el acceso a la Red y la conexiones de banda ancha aumentará el número de usuarios conectados a la Red y su agilidad en el uso de la misma. Existe una correlación directa entre el número

de usuarios que tiene una aplicación libre y la comunidad que la respalda. Cuanta más gente utilice el programa, más gente reportará errores y sugerencias, y nuevas instituciones, empresas, y voluntarios se sumarán a los esfuerzos ya existentes.

El precedente creado por SCO, que a principios del 2003 empezó a demandar a empresas usuarias de Linux, entre ellas IBM, por una supuesta violación de propiedad intelectual, ha mostrado la necesidad de protección legal para desarrolladores, usuarios, y proyectos de posibles reclamaciones legales infundadas. A medida que el software libre gane en adopción y más empresas hagan negocio con él aumentará el riesgo de que aparezcan fabricantes de software propietario que vayan a juicio para reclamar posibles violaciones de derechos de autor o patentes.

El software libre es técnicamente viable, económicamente sostenible y socialmente justo. Existe un mundo lleno de posibilidades. Aprovecharlo depende de nosotros.

# APÉNDICES

## Apéndice 1 – Webs relacionadas con el software libre

A continuación vamos a enumerar unos pocos sitios de noticias sobre software libre que nos pueden ayudar a mantenernos al día sobre noticias, documentación y acontecimientos.

### Sitios de Noticias

<a href="http://www.libroblanco.com">http://www.libroblanco.com</a>	Seguimiento de la implantación de iniciativas públicas de software libre. Incluye listados de grupos de usuarios de Linux
<a href="http://barrapunto.com/">http://barrapunto.com/</a>	Informaciones sobre software libre y ciencia.
<a href="http://libertonia.escomposlinux.org">http://libertonia.escomposlinux.org</a>	Noticias y artículos sobre Linux y software libre

<a href="http://www.libroblanco.com">http://www.libroblanco.com</a>	Seguimiento de la implantación de iniciativas públicas de software libre. Incluye listados de grupos de usuarios de Linux
<a href="http://www.laflecha.net/canales/softlibre">http://www.laflecha.net/canales/softlibre</a>	Canal de noticias de software libre
<a href="http://www.lapastillaroja.net">http://www.lapastillaroja.net</a>	La pastilla roja

## Documentación en español

<a href="http://es.tldp.org/">http://es.tldp.org/</a>	Recopilación de documentos de instalación de Linux en español.
<a href="http://www.uoc.edu/masters/softwarelibre/esp/materiales.html">http://www.uoc.edu/masters/softwarelibre/esp/materiales.html</a>	Materiales libres del máster en software libre de la Universitat Oberta de Catalunya.

## Organizaciones

<a href="http://www.fsf.org">http://www.fsf.org</a>	<i>Free Software Foundation</i>
---	---------------------------------

<a href="http://www.fsf.org">http://www.fsf.org</a>	<i>Free Software Foundation</i>
<a href="http://www.fsfeurope.org">http://www.fsfeurope.org</a>	Capítulo europeo de la <i>Free Software Foundation</i>
<a href="http://www.opensource.org">http://www.opensource.org</a>	<i>Open Source Initiative</i>
<a href="http://www.hispalinux.es">http://www.hispalinux.es</a>	Principal grupo de usuarios de Linux en España

## **Apéndice 2 - Declaración de Barcelona para el avance del software libre**

A mediados de mayo del año 2004 el consejo científico del máster en software libre de la UOC se reunió en Barcelona y como resultado creó la Declaración de Barcelona donde se analizan los retos del software libre en los siguientes años.

### **Declaración de Barcelona para el avance del software libre**

#### 1. Contexto histórico



## Internet es software libre

La mayoría de la infraestructura de Internet está basada en software libre y protocolos abiertos. Actualmente más del 60% de los servidores web utilizan Apache, un gran número de servidores de correo usan Sendmail para gestionar el envío de correo electrónico y prácticamente la totalidad de los servidores de nombres (DNS), esenciales en el funcionamiento de la Red, utilizan el programa BIND o derivados de su código fuente. Es indiscutible la importancia que ha tenido el software libre en la extensión y desarrollo de Internet desde sus inicios, y la influencia mutua de estos dos ámbitos tecnológicos es un hecho contrastado. Por lo tanto, el éxito del software libre va mucho más allá de la disponibilidad de una enorme cantidad de programas con licencias libres (entre los cuales el sistema operativo GNU/Linux, el navegador Mozilla o el paquete ofimático OpenOffice son ejemplos notables).

## Historia

Aunque los orígenes del software libre se sitúen en la década de los 60, con los primeros desarrollos de software, el movimiento como tal no se formalizó

hasta los 80, cuando tuvieron lugar, entre otros, los siguientes hechos:

- \* La creación del proyecto GNU is Not Unix (GNU), liderado por Richard Stallman.

- \* La constitución de la *Free Software Foundation* (FSF).

- \* La publicación de la primera versión de la *GNU Public License* (GPL).

- \* El desarrollo de BSD UNIX por parte de la Universidad de California en Berkeley.

- \* La libre circulación e intercambio del software a través de Internet.

Posteriormente, la de los 90 fue la década de la expansión de este movimiento. Dos factores fueron clave para este hecho. Uno es la llegada de los primeros sistemas operativos completos totalmente libres, como 386BSD -que más tarde evolucionaría a NetBSD y FreeBSD (con la contribución decisiva de la Universidad de California en Berkeley)- y GNU/Linux, en el que el trabajo de un estudiante finlandés, Linus Torvalds, permitió disponer de un núcleo libre para el sistema operativo iniciado por Stallman y la FSF. El otro es la popularización del acceso a Internet, que multiplicó la comunicación y la internacionalización de las comunidades encargadas

del desarrollo de software libre, además de facilitar su distribución. En la década actual ha empezado el proceso de consolidación del movimiento, como lo demuestra el hecho de que varios millones de personas usen productos de software libre en todo el mundo. Además, el software libre se usa de manera oficial en varias compañías (desde PYME hasta grandes multinacionales) e instituciones públicas, y el número de usuarios y desarrolladores de software libre no para de crecer. También cabe destacar las valiosas aportaciones de grandes compañías del sector informático, como Netscape, Sun Microsystems, IBM, Novell o Red Hat. Es de esperar que esta consolidación sea palpable en los próximos años, pero no se debe perder de vista que hay una serie de retos que el software libre tendrá que afrontar con éxito si quiere continuar con su crecimiento.

### 1.La aportación decisiva del mundo universitario

El mundo universitario ha tenido un papel muy destacado en el desarrollo de Internet y del software libre. Algunas de las tecnologías base del mundo libre, como los sistemas operativos basados en BSD, el sistema gráfico X-Window, o tantos otros, se han desarrollado y mejorado en las universidades. El propio Richard Stallman proviene del mundo

académico, y el reconocido padre del kernel Linux, Linus Torvalds, desarrolló la primera versión del mismo cuando aún era estudiante universitario.

## 2. Retos y nuevas oportunidades para el software libre

La década actual tiene que estar marcada por la extensión y la consolidación del software libre. Para alcanzar este objetivo debe afrontar diversos retos y algunas amenazas, de los que queremos dejar constancia en este documento. Este conjunto de retos y oportunidades se ha clasificado en siete ámbitos diferentes: académico, técnico, estratégico, legal, social, del voluntariado e institucional.

### \* *Ámbito académico*

El software libre posee ciertas características que lo convierten en un centro de interés para el mundo universitario. Desde el punto de vista de la investigación, el software libre traslada los principios básicos del modelo científico de producción de conocimiento (libre difusión, revisión por parte de expertos, constante búsqueda de mejoras, reproductibilidad de los resultados) al mundo del desarrollo del software, que además se ha convertido en una herramienta esencial para la investigación en

cualquier campo. Desde el punto de vista educativo, el software libre proporciona muchas ventajas (independencia del fabricante, facilidad de compartición del conocimiento, flexibilidad, etc.) que ya han sido identificadas por parte de muchas universidades. Por lo tanto, parece razonable que el software libre adquiera cada vez más importancia para las universidades y que se eliminen las principales barreras para su uso en investigación y docencia.

Aún un poco más allá, el modelo de compartición del conocimiento impulsado por el software libre puede extenderse hacia otras áreas, como por ejemplo la producción de materiales docentes, lo que puede representar toda una revolución en la organización de la enseñanza. Esta tendencia puede conducir a un nuevo modelo docente, dando lugar a un verdadero entorno abierto para la educación.

Muchas de las razones por las cuales el software libre se ajusta extremadamente bien a las necesidades de la educación superior también son aplicables a la enseñanza primaria y secundaria. Por lo tanto, el uso del software libre en esas etapas debería ser promovido e impulsado por las instituciones que tienen responsabilidad en esas áreas, siguiendo el

ejemplo de algunos casos de éxito como el de LinEx en Extremadura.

\* **Ámbito técnico**

Desde el punto de vista técnico, el software libre tiene que disponer de tecnologías que permitan simplificar el desarrollo de programas libres y mejorar la integración entre diferentes sistemas (como Mono o DotGNU). También tiene que continuar la evolución de las tecnologías de escritorio para facilitar el uso de los sistemas actuales (KDE, GNOME, Mozilla, OpenOffice). Además, se debe trabajar para consolidar los estándares (como OASIS) en el mundo del software libre con el fin de garantizar la interoperabilidad de todas las aplicaciones libres.

La integración del software en todo tipo de dispositivos electrónicos, más allá de los ordenadores personales de escritorio (por ejemplo, los dispositivos móviles, lectores de DVD, reproductores de música, etc.), abre una nueva vía de expansión en la cual el software libre debe introducirse como tecnología puntera, proporcionando independencia del proveedor.

Una oportunidad para el avance del software libre, desde el punto de vista técnico, sería desarrollar una

aplicación libre que se convierta en el referente en su ámbito. Por ejemplo, el campo de la seguridad y la privacidad constituye un terreno donde es posible crear una aplicación que se convierta en un estándar.

#### \* Ámbito estratégico

La liberación del código fuente de los programas, como ocurre con el software libre, es un estímulo decisivo a la competitividad y permite mejorar, a través de la cooperación, la calidad de las aplicaciones desde el punto de vista del usuario.

El software libre tiene que trabajar firmemente para combatir las técnicas FUD (*Fear, Uncertainty and Doubt* - miedo, incertidumbre y duda) que se utilizan en su contra. Para conseguirlo, la información debe ser la mejor herramienta. También hay que realizar estudios en profundidad sobre el coste total de propiedad (*Total Cost of Ownership, TCO*) y documentar casos de implantación con éxito de software libre.

#### \* Ámbito social

En el ámbito social hay que trabajar para generalizar el uso del software libre más allá de los especialistas y

de los usuarios expertos. Se debe trabajar para presentar el software libre como una alternativa dentro del mundo empresarial.

La sociedad de la información debe fundamentarse en la libre circulación de la información, pero este proceso requiere un dominio de la tecnología subyacente si se quiere participar de manera activa en él. El software libre es una herramienta necesaria para evitar una división entre los países que lideran este proceso y los que sólo intervienen en él como meros consumidores de tecnologías propietarias.

Las metodologías usadas en el software libre pueden utilizarse además como fórmula para abordar ciertos problemas sociales. El modelo abierto de los procesos de prueba y mejora de las soluciones puede aplicarse a ámbitos como la sanidad. Este concepto, conocido como web social, se está desarrollando actualmente en el Centro para la Sociedad de la Información de Berkeley.

\* *Ámbito legal*

Las iniciativas para extender el ámbito de lo patentable al software constituyen una gran amenaza para la industria del software en general y para el



software libre en particular. Este problema tiene una mayor relevancia ahora en Europa, donde las patentes de software todavía son objeto de debate, y se está discutiendo sobre la promulgación de una directiva al respecto.

Otro aspecto a considerar es la validez de las diferentes licencias como herramienta para proteger el software libre, sobre todo cuando se aplican en jurisdicciones diversas. Finalmente, convendría tomar medidas para garantizar la propiedad intelectual del software libre, de forma que se puedan evitar situaciones de inseguridad jurídica como las surgidas a raíz de las demandas interpuestas por SCO.

#### \* *Ámbito del voluntariado*

Es preciso seguir encontrando vías para mantener y promover la colaboración y las aportaciones voluntarias al software libre. Se debe preservar el espíritu de colaboración del voluntariado (como en el caso del proyecto Debian) y el papel destacado que tiene éste ante la apuesta de empresas e instituciones. El voluntariado tiene un rol fundamental para garantizar la calidad de los desarrollos y también permite conservar el espíritu y la filosofía del movimiento.

## \* *Ámbito institucional*

El estímulo para el uso y el desarrollo de plataformas libres y de código abierto en las administraciones públicas garantiza una mayor interoperabilidad de los sistemas de información, la posibilidad de auditoría de códigos fuente de los programas con el objetivo de preservar la seguridad y privacidad de los datos, la independencia de los distribuidores estimulando la competitividad y la reducción de costes en licencias de software.

Además el software libre contribuye al desarrollo de una industria de software en la región, sustituyendo la transferencia de royalties al exterior en forma de pagos de licencias por contratos de servicios basados en el nuevo modelo de negocios proporcionado por el software libre. Este hecho incentiva la implantación de nuevas empresas y la creación de puestos de trabajo cualificados, aprovechando el conocimiento local disponible.

Por ésta y otras razones, varios gobiernos en todos los continentes vienen desarrollando políticas públicas en este sentido. Pero es necesario que nos inspiremos en la lógica del principal motivo del éxito de la

comunidad del software libre y que estimulemos la cooperación entre las diferentes administraciones con vistas a disminuir los costes de desarrollo y a compartir experiencias.

En este sentido, proponemos la creación de una red de cooperación internacional entre administraciones públicas que establezca relaciones institucionales con organizaciones del tercer sector (incluyendo ONG), comunidades de desarrolladores de software libre, universidades, organismos de las Naciones Unidas y el sector privado, procurando alcanzar estos objetivos.

Barcelona, 18 de mayo de 2004

Copyright (c) 2004 Manuel Castells, Vinton Cerf, Marcelo D'Elia Branco, Juantomás García, Jesús M. González Barahona, Pekka Himanen, Miguel de Icaza, Rafael Macau, Jordi Mas, David Megías, Òscar del Pozo, Pam Samuelson. Se garantiza el permiso para copiar y distribuir este documento completo en cualquier medio si se hace de forma literal y se mantiene esta nota.

### **Apéndice 3 - Medida de gobierno para el impulso del software libre en el Ayuntamiento de Barcelona**

Medida de Gobierno para el impulso del software libre presentada por Jaume Oliveras de ERC en nombre de los tres grupos de gobierno en el Ayuntamiento de Barcelona (PSC-ERC-ICV) en el Plenario Municipal del 21 de julio de 2004.

#### **MEDIDA DE GOBIERNO PARA EL IMPULSO DEL SOFTWARE LIBRE**

Primero. El Ayuntamiento de Barcelona impulsará, mediante una migración progresiva y controlada, la utilización del software libre y en catalán en la administración municipal, los organismos autónomos públicos y las empresas públicas y de capital mixto bajo control mayoritario del Ayuntamiento. Se exceptuarán aquellas aplicaciones informáticas en que no sea posible disponer de software con licencia libre que incluya las soluciones técnicas objeto de la licitación pública.

Segundo. El avance del software propietario hacía el software libre en catalán se desarrollará mediante una prueba piloto de carácter limitado seguida de un exhaustivo estudio de los costes de implantación,

mantenimiento y eficacia que ofrecen todos los tipos de software.

Tercero. Se fomentará y se priorizará la utilización, el desarrollo, la distribución y la difusión de software libre en catalán en aquellos convenios de colaboración o decretos de otorgamiento de subvenciones que tengan relación con el ámbito de las nuevas tecnologías.

Cuarto. Del mismo modo, se colaborará en la difusión del software libre en catalán a la ciudadanía en general, y entre las asociaciones, organizaciones e instituciones, públicas o privadas, haciendo un especial énfasis en el ámbito educativo, y a la vez fomentando así la industria informática catalana.

Quinto. El Ayuntamiento velará porque en todas las comunicaciones que los ciudadanos puedan realizar informáticamente se garanticen los derechos de los usuarios que ya utilizan software libre. Esta normalización se garantizará especialmente al correo electrónico y a las páginas web, y se complementará el dominio .es con un dominio .net. El Plan Estratégico de Sistemas y el Plan de Administración Electrónica tienen en curso estudios para implementarlo pronto.

Sexto. Para facilitar la conclusión de estos objetivos, se establecerán convenios con otras administraciones de ámbito internacional que se encuentran avanzadas en este proceso, para compartir experiencias, herramientas formativas y software. Asimismo, para fomentar la coordinación y la cooperación al respeto, se promoverán acuerdos con otras administraciones locales, las cámaras de comercio, la Diputación de Barcelona, la Generalitat de Catalunya y el resto de las administraciones públicas que estén interesadas.

## BIBLIOGRAFÍA

Barahona, Jesus. Robles; Gregorio, Soane Joaquin (2003). *Introducción al software libre*, Barcelona, Universitat Oberta de Catalunya.

Cusumano, Michael A (2004), *The business of Software*, Free press.

García, Juan Tomas; Romero, Alfredo, *La pastilla roja: Software libre y revolución digital*, EditLin.

Himanem, Pekka (2001), *La ética del hacker*, Destinolibro.

Lessig, Lawrence (2001), *The future of ideas*, Random House.

Quittener, J. (1998), *Speeding the Net: The inside history of Netscape*, Atlantic montly press.

Young, Robert (1999), *Under the Radar*, Coriolis.

Williams, Sam (2002), *Free as in Freedom*, O'Reilly.