

artículo

Marco jurídico y oportunidades de negocio en el software libre

Jordi Mas

Resumen

El software libre plantea un nuevo modelo de entender las libertades del usuario, la propiedad intelectual y la creación y distribución de software. Ha demostrado ser técnicamente viable, económicamente sostenible y socialmente justo, y ha cambiado la forma de entender la industria del software. Linux, un sistema operativo desarrollado por voluntarios, es el buque insignia de esta revolución social y tecnológica que atrae el interés de usuarios, empresas y administraciones públicas de todo el mundo.

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 e instalando tecnologías libres, y también dando formación y soporte en estas tecnologías, y su número irá indudablemente en aumento.

Este artículo repasa los principales aspectos que debemos conocer del marco jurídico que regula la industria del software y profundiza en las oportunidades de negocio que el software libre ofrece.

Palabras clave

software libre, propiedad intelectual, código abierto

Abstract

Free software leads to a new way of understanding user freedom, intellectual property, and the creation and distribution of software. Changing the way we understand the software industry has been shown to be technically viable, economically sustainable and socially fair. Linux, an operating system developed by volunteers, is the standard bearer for this social and technological revolution which is attracting interest from users, companies and public administrations around the world.

Companies and administrations are starting to invest in free software based solutions that are tailor made and better adapted to their needs. Each day, thousands of businesses and professionals are employed in developing, installing, training and supporting free software and this number is sure only to increase.

This article reviews the main aspects that should be known with regard to the legal framework regulating the software industry and takes a closer look at the business opportunities offered by free software.

Keywords

free software, intellectual property, open source

Introducción

El software libre plantea un nuevo modelo de entender las libertades del usuario, la propiedad intelectual y la creación y distribución de software. Ha demostrado ser técnicamente viable, económicamente sostenible y socialmente justo, y ha cambiado la forma de entender la industria del software. Linux, un sistema operativo desarrollado por voluntarios, es el buque insignia de esta revolución social y tecnológica que atrae el interés de usuarios, empresas y administraciones públicas de todo el mundo.

La libertad de comerciar con el software libre, de venderlo y dar servicios sobre él, es una de sus características básicas. Muchas empresas e individuos aprovechan esta libertad para hacer negocios con el soft-

ware libre, tanto vendiéndolo como ofreciendo servicios de formación y consultoría.

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 que se incorporaron 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 de 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 que la propiedad intelectual 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, u oligopolistas en el mejor de los casos, debido a los derechos exclusivos que el fabricante tiene sobre el software, que se traducen en un control exclusivo sobre la mejora y la distribución del propio software. En este sentido, la evidencia empírica demuestra que la mayoría de los mercados horizontales en la industria del software están 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. Éste es un hecho importante si tenemos en cuenta que aproximadamente el 75%¹ 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.

1. La industria del software

El software libre ha invertido la lógica en lo relativo a 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 (por medio de una licencia). Aunque ninguna de las licencias de software libre imponen restricciones en la venta del software,² 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, afrontan unos costes marginales muy bajos, donde gran parte del precio se convierte en beneficio. Los costes fijos que tienen, principalmente el desarrollo de nuevas versiones del producto, 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. Ello produce el efecto de que, tras haberse instaurado un vendedor en posición dominante en un mercado, si lo cree conveniente, éste pueda fácilmente vender a un precio muy bajo, lo que dificulta 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 *best seller*. 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%³ del total de su negocio, mientras que el 70% restante procede de la prestación de servicios.

2. Marco legal

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 e 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 hay que entender cómo podemos ceder nuestros programas a nuestros usuarios y qué derechos y obligaciones les estamos ofreciendo.

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

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

3. Véase el libro *Secrets of Software Success: Management Insights from 100 Software Firms Around the World*, escrito por empleados de la consultora McKinsey en Alemania.

2.1. 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 que 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 o distribuir, o hacer trabajos derivados de ella, 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. Desde un punto de vista global, podemos señalar que ya en 1886 se firmó el convenio de Berna, que fue ampliado en 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. Ello 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 a la persona del autor y sus derechos personales.

2.2. 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 informá-

tico 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. Este contrato es denominado 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 de las condiciones de la misma y las acepte, los proveedores le obligan a aceptar la licencia cuando se instala el programa –licencia *click-wrap*– o a veces, incluso, en algunos programas propietarios, cuando simplemente abre el sobre o paquete que lo contiene –licencia *shrink-wrap*–; la validez legal de ambos procesos 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,⁴ uno de los mayores depósitos de proyectos libres en Internet, en julio de 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%, licencias del tipo BSD. Como vemos, es abrumador el uso y la extensión de las licencias GPL y LGPL creadas por la Free Software Foundation.⁵

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 las licencias⁶ que ceden los derechos de copia, distribución y modificación del programa bajo las condiciones que definen el 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 véricas 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 el 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, lo que garantiza 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*⁷ (conocida por su acrónimo GPL), que ha sido formulada por la Free Software Foundation y que goza de amplia popularidad. Es una licencia que

4. <http://sourceforge.net/>

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

6. <http://www.gnu.org/copyleft/copyleft.html>

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

ubre con gran detalle los diferentes aspectos del software. Existe una recopilación larga pero exhaustiva de preguntas frecuentes con sus respuestas sobre la GPL.⁸

Licencias de código abierto o permisivas. Son las 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 en 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,⁹ con lo que sólo los clientes de Netscape podían beneficiarse de estas mejoras a pesar de 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 incluso permiten crear software propietario de un proyecto libre, y otros que argumentan que este tipo de licencias restan libertad a los usuarios, dado que no garantizan que futuras versiones sean libres.

La licencia BSD¹⁰ es una de las más comunes de este tipo. Lo más destacable es que permite el uso y la 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.¹¹ 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.¹² Esta licencia permite que el software de Berkeley DB sea hoy

en día usado 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 que 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.

2.3. 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 período limitado de 20 años, tras el que todo el mundo puede hacer uso de la misma. Originalmente fueron creadas para que las personas pudieran recuperar las grandes inversiones efectuadas en investigación, a cambio de que adelantaran al público los detalles de su creación, a fin de producir 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 escala 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 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 aplicación técnica suficiente. Actualmente, en la Unión Europea aún se está debatiendo un cambio de ley, mientras se permiten y se reconocen 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 otro, la innovación informática es acumulativa y basada en experiencias e información previas. Faltaría, así pues, el requisito de «inventividad».

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

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

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

11. <http://www.sleepycat.com>

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

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 para 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 el hecho de 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 con un único clic, «1-click». Éstos 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, a fin de minar 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, con lo que violan un aspecto esencial de la patentabilidad y la idea de 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 Proinnova¹³ 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.

2.4. 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 claramente que un producto o servicio proviene 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 e incluso 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 mane-

ra, 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 noventa.

3. Modelos de negocio

Frank Hercker escribió en 1998, mientras era empleado de la empresa Netscape, un detallado ensayo¹⁴ 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 esta 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 de 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 los venden por medio de un canal en forma de producto. Son empresas que tienen presupuestos importantes destinados a financiar la investigación y el desarrollo de estos proyectos libres, al contrario que las empresas de servicio, que normalmente 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; IBM, que incluye el entorno de desarrollo Eclipse como parte de su solución WebShepere. 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, 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 pro-

13. <http://proinnova.hispalinux.es>

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

yectos 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 esté dispuesto a publicar el código de sus mejoras y, por otro, 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, el desarrollo a medida de soluciones, la formación y el 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 las que se especializan en un área concreta de conocimiento, como puede 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 pueden acelerar enormemente la creación de soluciones competitivas.

Según LibroBlanco.com,¹⁵ una iniciativa que hace un seguimiento del software libre en el ámbito del sector público, en julio de 2004 había 289 empresas en España contabilizadas que trabajaban y daban 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 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 Zaurus¹⁶ 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 y 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 ello representa para el importante sector asiático de fabricantes de electrónica de consumo. Durante el verano de 2004 se presentó¹⁷ 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 setenta, 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* (macroordenadores), o en los años noventa en la industria del entretenimiento, donde se ha convertido en un líder indiscutible en el desarrollo de software de videojuegos.

3.1. 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 y cómo se enfocaron, e intentar entender por qué no tuvieron éxito. Es importante reconocer el riesgo que tomaron porque, aunque fallidas, fueron innovadoras al tratar de poner en marcha experiencias de negocio que antes no se habían intentado.

SourceGear

En 1998 la empresa SourceGear empezó a trabajar en el desarrollo del procesador de textos libre Abiword.¹⁸ 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 los ayudara a crear su producto de forma más rápida, lo que disminuiría los costes de desarrollo del mismo. Desgra-

15. <http://www.libroblanco.com/>

16. <http://www.myzaurus.com/>

17. http://news.com.com/Asian+Linux+gaining+momentum/2100-1011_3-5278304.html

18. <http://www.abiword.com>

ciadamente, 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 proyectos de software.

Eric Sink, fundador de SourceGear, explica¹⁹ cómo su empresa fracasó al intentar posicionar Abiword como solución en diferentes nichos de mercado. El error principal de SourceGear fue intentar obtener beneficios tal y como lo han hecho tradicionalmente los vendedores de software propietario. Tras 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,²⁰ que implementó una parte muy importante de la interfaz 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 en su haber, y con un nivel de financiación que era envidiable para el resto de la 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 un amplio 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. Pero nunca llegó a acumular suficientes usuarios para hacer la empresa mínimamente rentable.

Eazel cerró sus puertas en mayo de 2001.²¹ 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 de 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 han construido a su alrededor y han impulsado su desarrollo.

3.2. Experiencias de éxito

Existen en el mundo miles de empresas y profesionales que se ganan la vida cada día desarrollando, instalando tecnologías libres, y también ofreciendo formación y soporte en estas tecnologías. Todos ellos son casos de éxito, pero son casi imposibles de documentar. Veremos cuatro casos concretos de empresas que usan el software libre como parte de su estrategia de negocio o que incluso están totalmente basadas 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 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 lo era 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 siempre ha intentado 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 hay seguridad, desarrollo de Kernel y sistemas empujados, gestión avanzada de redes o desarrollo web. Su certificación *Red Hat Certified*

19. <http://software.ericssink.com/20011228.html>

20. http://en.wikipedia.org/wiki/Andy_Hertzfeld

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

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 constituyen un área en la que 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 por medio 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 la de otros distribuidores de Linux, como la empresa alemana Suse (actualmente parte de Novell) o la francesa Mandrake Software.

IBM

A finales de 1997, IBM²² estaba considerando cómo reorganizar su oferta de servicios web. Con menos del 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 continua dinero y recursos al desarrollo y la 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 el 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. Así mismo, 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 de 2003, empezó a demandar a empresas usuarias de Linux, entre ellas IBM, por una supuesta violación de propiedad intelectual. IBM ha demostrado que está del lado del software libre y de defender a Linux ante 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 hacia 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.²³

Ándago

Ándago²⁴ 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 importante durante ese año 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,²⁵ la distribución de Linux adaptada para su divulgación a toda la ciudadanía extremeña.

Ándago ha desarrollado plataformas y 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 AB²⁶ 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.

22. <http://www.ibm.com>

23. http://news.zdnet.com/2100-3513_22-275388.html

24. <http://www.andago.com>

25. <http://www.linex.com>

26. <http://www.mysql.com/>

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 estipuladas. Si redistribuyen una solución basada en MySQL, el código 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 facilitar 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 seguir desarrollando el producto libre MySQL.

Conclusión

Hemos visto cómo el software libre ya ha cambiado la forma de entender la industria del software. Las empresas han entendido la libertad y

el control que proporciona el software libre para realizar modificaciones, distribuir las y desvincularse de 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 incluso 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.

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, ofreciendo formación y soporte en tecnologías libres, y su número irá indudablemente en aumento.

Citación recomendada:

MAS, Jordi (2005). «Marco jurídico y oportunidades de negocio en el software libre». *UOC Papers* [artículo en línea]. N.º 1. UOC. [Fecha de consulta: dd/mm/aa]. <<http://www.uoc.edu/uocpapers/1/dt/esp/mas.pdf>>
ISSN 1885-1541



Esta obra está bajo la licencia Reconocimiento-NoComercial 2.5 de Creative Commons. Puede copiarla, distribuirla y comunicarla públicamente siempre que especifique su autor y UOC Papers; y no la utilice para fines comerciales. La licencia completa se puede consultar en: <http://creativecommons.org/licenses/by-nc/2.5/es/deed.es>



Jordi Mas
Ingeniero de software Ximian
Coordinador general de Softcatalà
jmas@softcatala.org

Ingeniero técnico por la UOC. Actualmente trabaja como ingeniero de software en la empresa de código abierto Ximian/Novell, donde desarrolla la implementación del proyecto libre Mono. Es co-coordinador académico del máster de Software libre de la UOC. Como voluntario, colabora en el desarrollo del procesador de textos Abiword y en la ingeniería de las versiones en catalán de los proyectos Mozilla y Gnome. Así mismo, es coordinador general de Softcatalà. Como consultor ha trabajado para empresas como Menta, Telépolis, Vodafone, Lotus, eresMas, Amena y Terra España.