



IDP. Revista de Internet, Derecho y Política

E-ISSN: 1699-8154

dbindexing@uoc.edu

Universitat Oberta de Catalunya

España

Jacovkis, David

El software libre: producción colectiva de conocimiento

IDP. Revista de Internet, Derecho y Política, núm. 8, 2009, pp. 4-13

Universitat Oberta de Catalunya

Barcelona, España

Disponible en: <http://www.redalyc.org/articulo.oa?id=78813256006>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto



<http://idp.uoc.edu>

Monográfico «Software libre»

ARTÍCULO

El software libre: producción colectiva de conocimiento

David Jacovkis

Fecha de presentación: noviembre de 2008

Fecha de aceptación: diciembre de 2008

Fecha de publicación: marzo de 2009

Resumen

El software libre ha dejado de ser dominio exclusivo de los expertos en informática. La mayoría de los usuarios de Internet conocen el navegador web Firefox o el paquete ofimático OpenOffice.org, y muchos los utilizan cada día. Pero en el sector de las tecnologías de la información y la comunicación, ya hace muchos años que se utilizan programas libres por diversos motivos: la calidad técnica, la falta de costes por licencias, la seguridad, la independencia tecnológica, etc. En algunos sectores, como las administraciones públicas o la educación, las libertades que el software libre garantiza a sus usuarios son especialmente relevantes. En este artículo presentaremos una breve historia de este movimiento en sus vertientes tecnológica e ideológica. Veremos qué quiere decir que un programa sea libre y qué importancia tiene eso para sus usuarios desde el punto de vista técnico, económico y social.

Palabras clave

software libre, propiedad intelectual, Internet

Tema

Software libre

Free Software: Collective Production of Knowledge

Abstract

Free software is no longer the exclusive domain of computer experts. Most Internet users know about the Firefox web browser or the OpenOffice.org suite, and many use them every day. But in the ICT sector, free software applications have been used for many years for a number of reasons: technical quality, absence of license fees, security, technological independence, etc. In some sectors, such as public bodies or education, the freedoms that free software guarantees are specially important. In this article we present a brief history of the technological and ideological aspects of the free software movement. We discuss what is free software and why it is important for users, from the technical, economical and social points of view.

Keywords

free software, intellectual property, internet

Topic

Free software

1. Introducción

El siglo XX fue testigo de cambios revolucionarios en muchas disciplinas científicas: la física cuántica, el psicoanálisis o la genética son sólo algunos ejemplos de las nuevas ramas del conocimiento que trajo esta edad dorada de las ciencias. Durante el siglo XX también se vivieron grandes avances en las telecomunicaciones: el uso masivo de la telefonía, la radio, la televisión, y finalmente la red de redes a la que llamamos Internet han hecho que el mundo sea hoy mucho más pequeño de lo que era hace 100 años.

De estas nuevas ciencias, la informática es quizás la que ha propiciado cambios más visibles en nuestra sociedad. A caballo entre la matemática y la electrónica, la aplicación de la informática a la fabricación de ordenadores y al diseño de programas informáticos está transformando todos los aspectos de nuestra vida cotidiana, unas veces de forma radical y otras más sutilmente.

En este artículo mostraremos cómo los ordenadores pasaron de ser dominio exclusivo de unos pocos académicos e ingenieros, cuando la informática estaba dando sus primeros pasos, a convertirse en una industria que suministra productos y servicios a la práctica totalidad de los sectores económicos de nuestra sociedad. Este crecimiento espectacular ha ido acompañado de grandes cambios, tanto en la forma de desarrollar y distribuir el software como en los modelos de negocio de las empresas que lo producen.

A la mayoría de usuarios de programas de ordenador de hoy en día les parece perfectamente normal pagar al distribuidor no por el programa en sí, sino por el derecho a hacer un uso limitado –por un tiempo limitado o en una máquina determinada. Pero no siempre ha sido así, y no siempre lo es en la actualidad. Al principio de la década de 1980, cuando el mercado de los programas de ordenador empezaba a crecer, surgió un movimiento social para defender el derecho de los usuarios a tener control total sobre los programas que utilizan. Este movimiento ha resultado en una opción perfectamente viable desde el punto de vista técnico, económico y político para muchos sectores. Es lo que conocemos como *software libre* o *software de código fuente abierto*.

Pero antes de entrar en los detalles de esta historia necesitamos definir una serie de términos y conceptos básicos que nos permitirán alcanzar una visión general del pasado, presente y futuro del software libre.

1.1. Hardware y software

Los primeros ordenadores eran unos gigantes mecánicos que ocupaban habitaciones enteras, mientras que hoy en día cualquier teléfono móvil tiene una potencia de cálculo muy superior a la de aquellos monstruos. Aun así, hay una serie de conceptos que son comunes a todos los ordenadores, y que introduciremos a continuación.

El hardware es el conjunto de piezas que forman el ordenador. La parte más importante del hardware es el procesador central, el dispositivo donde se ejecutan todas las instrucciones que hacen funcionar el equipo. El procesador es capaz de realizar miles de operaciones por segundo, pero sólo puede entender un número limitado de instrucciones sencillas.

El software es el conjunto de programas que se ejecutan en un ordenador y nos permiten escribir un texto, navegar por Internet o retocar una fotografía. El programa más importante de cualquier ordenador es el sistema operativo. En realidad se trata de un conjunto de programas que permiten al usuario interactuar con el hardware y ejecutar o desarrollar otros programas. Mac OS, Windows XP y GNU/Linux son ejemplos de sistemas operativos.

Cuando se escribe un programa de ordenador se utiliza un lenguaje de programación, como Java o C. Se trata de lenguajes formales que pueden ser leídos y escritos por cualquier persona con la formación adecuada y que permiten expresar de forma estructurada las instrucciones que el ordenador tiene que ejecutar para llevar a cabo una tarea o resolver un problema. Este conjunto de instrucciones en un lenguaje determinado es lo que denominamos *código fuente de un programa*.

Sin embargo, ¿cómo es posible que el procesador central, que sólo entiende un conjunto limitado de instrucciones sencillas, pueda ejecutar un programa escrito en un lenguaje de programación? La respuesta es simple: hay otro programa, un compilador o un intérprete que se encarga de transformar el código fuente del programa en un listado de instrucciones que el procesador puede ejecutar. Cuando un programa se encuentra en este formato que puede ser directamente ejecutado por el ordenador, decimos que se encuentra en *formato binario*. El inconveniente del formato binario es que sólo el procesador es capaz de entenderlo, ni siquiera quien lo ha programado

podría leerlo fácilmente. Si queremos estudiar o modificar un programa, necesitamos su código fuente, el programa escrito en un lenguaje de programación.

1.2. Derechos de autor, licencias y patentes

Otro conjunto de conceptos que hay que introducir antes de continuar tiene que ver con los derechos de propiedad del software. Una discusión en torno a los derechos de propiedad sobre bienes intangibles está fuera del alcance de este artículo, pero hay aspectos de los derechos de autor y de las patentes que son muy relevantes para los programas de ordenador en general y para el software libre en particular.

Los programas de ordenador están protegidos por los derechos de autor. Muy a menudo, los derechos económicos son cedidos por el programador a una empresa que produce o distribuye software, la cual puede decidir de qué forma distribuye la obra, de la misma manera que se hace con una novela. Tal como pasa en la industria editorial, la forma más habitual de distribuir software reserva todos los derechos para el titular, prohibiendo al usuario, por ejemplo, hacer copias del programa para sus amigos. Además, en la gran mayoría de los casos el usuario sólo recibe el programa en formato binario, lo cual hace imposible que lo pueda estudiar o modificar. Decimos que los programas que se distribuyen de esta forma son privados, ya que privan a los usuarios de ejercer ciertas libertades, como la de estudiar el programa o la de hacer copias. Pero como veremos más adelante, hay otras formas de distribuir software, en las cuales el titular de los derechos cede una parte al usuario del programa.

¿Qué tienen que ver, pues, las patentes con el software? La respuesta, en un mundo ideal, sería sencilla: absolutamente nada.

Una patente es un monopolio concedido por el Estado sobre la explotación de una determinada tecnología, a cambio de que el inventor haga públicos los detalles. El objetivo de las patentes es pues incentivar el progreso tecnológico compensando el esfuerzo que representa la innovación. Pero hay ciertas cosas que no son patentables en la mayoría de los países, como las teorías científicas y los métodos matemáticos. Los programas de ordenador no son más que algoritmos matemáticos que

realizan una tarea determinada, y como tales, no tendrían que ser patentables. Además, la inversión necesaria para desarrollar un programa de ordenador es muy baja comparada con las inversiones necesarias en otros sectores industriales, y no justifica la concesión de patentes en este sector.

Sea como sea, en los últimos años se han concedido miles de patentes de software en Estados Unidos y también en la UE, práctica que algunos entienden contraria a la propia regulación europea de patentes. Además, las supuestas innovaciones que cubren estas patentes de software son, en muchos casos, de uso generalizado o no representan realmente un nuevo desarrollo, como las que cubren el doble clic del ratón [1] o el carro de la compra virtual [2]. Hay muchas organizaciones no gubernamentales y asociaciones de profesionales que han puesto en marcha campañas para alertar del peligro que representan las patentes de software para la innovación en este campo [3,4].

2. Un poco de historia

Los ordenadores modernos son una evolución de las máquinas de calcular, como el ábaco o el astrolabio, que han facilitado los cálculos matemáticos durante siglos. Pero mientras que cada máquina de calcular está diseñada para resolver un cierto conjunto de problemas aritméticos, el ordenador puede ser programado para ejecutar cualquier tarea que se pueda expresar en forma de algoritmo. Ya en la primera mitad del siglo XIX, Charles Babbage diseñó su motor analítico [5], un ordenador mecánico programable que nunca pudo ser fabricado debido a dificultades técnicas y al carácter difícil del propio Babbage. Durante todo el siglo XIX y la primera mitad del siglo XX se desarrollaron de forma paralela los aspectos tecnológicos y la teoría matemática que llevaron a la fabricación de los primeros ordenadores digitales en la década de 1940.

La microelectrónica trajo la miniaturización a los ordenadores, haciéndolos más pequeños y asequibles a la vez que se multiplicaba su potencia de cálculo, en una rápida evolución que continúa hoy en día. Pero a finales de los años setenta los ordenadores todavía eran máquinas muy caras, y su uso estaba limitado a universidades, grandes empresas y agencias estatales.

2.1. Los primeros *hackers*

En este periodo, la mayor parte de los programas los desarrollaban los propios usuarios. A las empresas que vendían los ordenadores no les preocupaba lo que los usuarios hicieran con los sistemas operativos que acompañaban a las máquinas, su negocio era vender y mantener el hardware. Los programas, especialmente en las universidades, se copiaban y modificaban sin restricciones como cualquier otra pieza de conocimiento académico. A todo el mundo le parecía natural aprovechar el trabajo de otros y adaptarlo a las necesidades propias en lugar de empezar cada programa desde cero, de la misma manera que se hace con las ecuaciones que describen un sistema físico o con la fórmula química de una sustancia.

Los desarrolladores que formaban esta comunidad, vinculados principalmente a universidades, se denominaban a sí mismos *hackers*, programadores apasionados por resolver problemas informáticos y obtener el reconocimiento del resto de la comunidad.

Pero a principios de los años ochenta, las empresas que distribuían software empezaron a obligar a sus clientes a firmar contratos de no divulgación, impidiendo a los usuarios compartir estos programas o las mejoras que hubieran hecho. Así, cada usuario tenía que pagar por el derecho a utilizar el software bajo unas condiciones cada vez más restrictivas, incluso cuando sólo había recibido una copia en formato binario y no el código fuente, en un esquema muy similar al que siguen hoy las empresas distribuidoras de software privativo. Hacer copias de un programa se había vuelto ilegal, y sólo la empresa que lo distribuía o algún otro con su consentimiento podían introducir cambios en el mismo.

2.2. El proyecto GNU

En este momento crítico, muchos miembros de la comunidad de desarrolladores dejaron las universidades para ir a trabajar a empresas que producían programas privativos, o a fundar otras nuevas. La comunidad de *hackers* parecía condenada a ir desvaneciéndose poco a poco. En el laboratorio de inteligencia artificial del prestigioso MIT trabajaba desde 1971 Richard M. Stallman, un miembro activo de esta comunidad cooperativa y muy consciente de la amenaza que representaba el software privativo.

Stallman partía de la convicción moral de que los usuarios tenían que tener el derecho a compartir el software, a estudiarlo y a modificarlo libremente como habían hecho durante años.

Con esta convicción como bandera, Stallman dejó su trabajo en el MIT para poner en marcha el proyecto GNU en el año 1984, con la publicación del «GNU Manifesto» [6]. Su objetivo era ambicioso: desarrollar un sistema operativo totalmente libre. El sistema GNU sería compatible con Unix, un sistema operativo desarrollado por AT&T en 1969 y que era el más popular en aquel momento, y por eso lo llamó «GNU no es Unix», siguiendo la tradición *hacker* de utilizar acrónimos recursivos. En el año 1985 se creó la Fundación por el Software Libre (Free Software Foundation, FSF) para dar apoyo y conseguir financiación para el proyecto.

Stallman y un grupo de voluntarios empezaron a desarrollar los primeros programas para el sistema GNU, pero también pretendían incorporar tantas aplicaciones existentes como fuera posible, aunque no a cualquier precio. Estos programas tenían que ser libres, es decir, tenían que garantizar al usuario las siguientes libertades [7]:

1. La libertad de utilizar el programa sin ninguna restricción.
2. La libertad de estudiar el programa y adaptarlo a sus necesidades.
3. La libertad de hacer copias del programa para regalarlas o venderlas.
4. La libertad de mejorar el programa y hacer públicas estas modificaciones para que el resto de usuarios se pueda beneficiar de ello.

Es importante destacar que las libertades 1 y 3 requieren que el usuario tenga acceso al código fuente del programa, tal como hemos visto anteriormente. Esta definición de software libre es la aceptada de forma mayoritaria por la comunidad, y la que utilizaremos en lo sucesivo en este artículo.

Pronto surgió un problema con esta definición: algunos programas libres utilizaban licencias que permitían distribuir modificaciones sin publicar el código fuente correspondiente. Estas licencias permisivas permiten que un programador aproveche el código fuente como quiera, ya

sea para incorporarlo a un programa libre o para un programa privativo.

Las licencias permisivas no garantizan que el software que hoy es libre lo sea también en el futuro: si una empresa mejora un programa libre y lo redistribuye con una licencia privativa, la comunidad no puede disfrutar de estas mejoras. Por eso, desde el proyecto GNU se creó el concepto de *copyleft*, que hace uso de los derechos de autor para proteger las libertades de los usuarios. Así, lo que normalmente es «todos los derechos reservados» se convierte en «todos los derechos invertidos», el autor concede las cuatro libertades antes mencionadas al usuario, añadiendo una condición: si se hace público el programa con modificaciones, se tiene que publicar bajo la misma licencia, de modo que continúe siendo libre. La licencia pública general (GPL, General Public License) es la que utiliza el proyecto GNU para sus programas, y una de las más utilizadas por la comunidad.

2.3. En busca de un núcleo

A finales de los años ochenta, los participantes en el proyecto GNU habían reunido casi todas las piezas necesarias para alcanzar su objetivo, un sistema operativo totalmente libre. Pero todavía les faltaba una parte fundamental: el *kernel*, el núcleo del sistema operativo. El núcleo es un programa muy especial que se encarga de la comunicación entre el hardware y el resto de programas, sin el cual el software GNU se tenía que ejecutar sobre sistemas Unix privativos. GNU estaba desarrollando su propio núcleo libre, el Hurd, pero todavía estaba muy lejos de poder ser utilizado de forma generalizada.

En el año 1991, el estudiante finés Linus Torvalds empezó a trabajar en un núcleo compatible con Unix, y lo llamó Linux [8]. Desde muy pronto publicó su trabajo bajo la licencia GPL, y programadores de todo el mundo empezaron a contribuir de forma voluntaria. En 1992, se empezó a utilizar la combinación del núcleo Linux y el sistema GNU dando lugar al sistema operativo GNU/Linux, que se ha convertido en uno de los productos más destacados del movimiento del software libre. Linus Torvalds todavía está al frente del desarrollo de Linux, coordinando el trabajo de centenares de programadores, mientras que Richard Stallman dedica casi todo su tiempo a promover el software libre desde la FSF.

2.4. La madurez de un modelo de producción de conocimiento

A medida que el sistema operativo GNU/Linux se hacía más fiable y robusto, sus desarrolladores dejaron de ser sólo *hackers* voluntarios. Muchas empresas de diferentes sectores empezaron a ver en el nuevo sistema una alternativa a los Unix privativos, que les permitiría adaptar el sistema a sus necesidades a la vez que se ahorraban las licencias del software privativo. Proveedores de acceso a Internet, distribuidores de hardware para servidores y proveedores de servicios en red fueron de los primeros en explotar comercialmente el nuevo sistema operativo libre, junto con otros programas libres muy populares, como el servidor de páginas web Apache o el servidor de correo electrónico Sendmail.

Muy pronto surgieron las primeras distribuciones de GNU/Linux, que empaquetaban el sistema operativo junto con muchas otras aplicaciones libres. Algunas distribuciones, como Debian GNU/Linux o Slackware, son mantenidas por una comunidad colaborativa, mientras que otras como RedHat Linux o SuSE dependen de una empresa que vende el producto resultante junto con manuales, certificaciones, apoyo técnico y otros servicios asociados.

Así, ha ido tomando forma un ecosistema en el que voluntarios, empresas y centros académicos mantienen de forma cooperativa Linux, el software GNU y centenares de programas libres. En algunos casos, especialmente en proyectos grandes y populares en ámbitos muy diferentes, como Linux, la comunidad de desarrolladores la forman individuos y organizaciones muy variados. En otros casos es una única organización -a veces una sola persona- quien saca adelante el proyecto. Algunas comunidades son más abiertas a admitir nuevos participantes, y otras no lo son tanto, pero en todos los casos cualquiera tiene la libertad de estudiar el código, y si no le gusta cómo funciona el proyecto original, nadie le puede impedir crear una nueva versión con sus propias modificaciones.

En el año 1998, un grupo de destacados *hackers* y activistas de la comunidad fundaron la Iniciativa por el código fuente abierto (Open Source Initiative, OSI). Con Eric S. Raymond y Bruce Perens al frente, este grupo quería desvincularse del estilo de Stallman y la FSF, según ellos demasiado radical en sus planteamientos y sus acciones. En un tono mucho más conciliador y atractivo para el

mundo empresarial, apostaron por el uso del término *open source*, insistiendo en la superioridad técnica de este modelo de desarrollo y al mismo tiempo evitando la confusión que origina en inglés el término *free*, que quiere decir «libre» pero también «gratis». La definición de *open source* de la OSI es idéntica al contrato social de la distribución Debian GNU/Linux, que fija las condiciones que tiene que cumplir la licencia de un programa para que éste pueda ser incluido en Debian. Uno de los principales problemas de la OSI es que algunas empresas de software privativo utilizan el término *open source* para referirse a otras cosas, como dar acceso limitado a partes del código fuente de un programa sin derecho a hacer modificaciones y bajo contratos de confidencialidad, cosa que no tiene nada que ver con los objetivos de la OSI.

La FSF respondió a la creación de la OSI criticando que se dejara de lado la libertad de los usuarios en beneficio de la tranquilidad de los empresarios, y defendiendo el uso del término *free software*. A pesar de estas diferencias de fondo, los objetivos a corto plazo de ambas organizaciones son similares, y las definiciones de software libre de la FSF y de software de código fuente abierto de la OSI son equivalentes prácticamente en todos los casos: la mayoría de licencias aceptadas por la FSF lo son también por la OSI, o podrían serlo, y a la inversa. La OSI y la FSF, el software de código fuente abierto y el software libre, representan las dos vertientes de la comunidad, una más interesada en la eficiencia y la superioridad tecnológica y la otra en los principios morales y la libertad de los usuarios.

3. El software libre e Internet, una relación simbiótica

La idea de una red de comunicaciones que conectaría máquinas de todo el mundo ya formaba parte del imaginario de los autores de ciencia-ficción en los años cuarenta, cuando los primeros ordenadores empezaban a funcionar. Durante las décadas de los sesenta y setenta, a la vez que los ordenadores se hacían más potentes, se desarrollaron diferentes tecnologías para el intercambio de datos. Uno de los desarrollos más importantes fue el juego de protocolos de Internet, de los que cabe destacar el protocolo de control de transmisiones (*transmission control protocol*) y el protocolo de Internet (*Internet protocol*), que combinados denominamos TCP/IP. Este protocolo, que se continúa utilizando en Internet hoy en día, es el que hace posible que

la información que se quiere transmitir se divida en pequeños paquetes que viajan por rutas independientes dentro de la Red y se vuelven a unir cuando llegan a su destino. Desde el comienzo, este protocolo se desarrolló de forma abierta y libre de *royalties*, lo cual facilitó mucho su adopción generalizada.

La primera red de un tamaño considerable en utilizar el protocolo TCP/IP fue ARPANET, en el año 1983. Desde aquel momento otras redes de ordenadores fueron migrando desde protocolos de comunicación más antiguos para adoptar el TCP/IP, integrándose en la Red ya existente. Esta tendencia se consolidó con la incorporación de diferentes redes comerciales a la infraestructura que se empezaba a conocer como *Internet*.

En el año 1991 Tim Berners-Lee, miembro del centro de investigación europeo CERN, publicó su proyecto *World Wide Web* para crear una red de documentos conectados mediante hipervínculos, que acabó siendo la aplicación más popular de la Red junto con el correo electrónico. Durante la década siguiente el número de máquinas conectadas a Internet se duplicó anualmente. Este espectacular crecimiento se suele atribuir al hecho de que Internet no tiene una administración centralizada, y al carácter abierto y no privativo de los protocolos que gobiernan la Red.

Se pueden establecer muchos paralelismos entre la evolución del software libre y el crecimiento de Internet. Ambos procesos surgieron en ambientes académicos y se desarrollan de una forma abierta y descentralizada, y ambos se caracterizan por su capacidad de agregar recursos ya existentes para acelerar su crecimiento.

El software libre facilitó mucho la expansión de Internet. Desde sus inicios, los programas más utilizados para mantener la infraestructura de Internet y ofrecer servicios son programas libres: el servidor de nombres BIND hace funcionar la práctica totalidad de los servidores de nombres de dominio (*domain name system*, DNS), que permiten localizar los servidores de Internet utilizando nombres de dominio, como *uoc.edu*, en vez de direcciones numéricas como *194.224.66.57*. El servidor libre de páginas web Apache sirve más del 50% de las páginas existentes, y el 60% de los servidores de correo electrónico utilizan software libre. Además, el hecho de disponer de un sistema operativo libre permite que muchas organiza-

ciones puedan estar presentes en la Red con unos costes muy bajos.

Al mismo tiempo, es innegable que el movimiento del software libre debe gran parte de su éxito a Internet. Durante los años ochenta el proyecto GNU enviaba por correo las copias de sus programas en cintas magnéticas. Pero cuando Torvalds publicó Linux en 1991, el anuncio se hizo en un grupo de noticias de Internet, y el código se podía descargar de un servidor FTP. Desarrolladores de todo el mundo se pusieron en contacto con él por correo electrónico, enviándole fragmentos de código, sugerencias o críticas. Así es como funcionan la gran mayoría de proyectos de software libre, con una serie de herramientas de colaboración en red que permiten el trabajo simultáneo de un gran número de programadores y usuarios que reportan errores y contribuyen con fragmentos de código y sugerencias.

4. Situación actual del software libre

Ya hace años que el software libre dejó de ser dominio exclusivo de *hackers*, expertos en informática y aficionados con mucho tiempo libre. El sistema operativo GNU/Linux, la base de datos MySQL, el servidor de páginas web Apache y el sistema Samba para compartir archivos y recursos en red son sólo algunos ejemplos de programas libres con una cuota de usuarios muy importante, y todos ellos continúan creciendo. Por eso no tiene que sorprender a nadie que todas las grandes empresas de las tecnologías de la información y la comunicación (TIC) tengan una estrategia específica para el software libre, incluso los grandes productores de programas privativos.

4.1. Modelos de negocio

Hay muchas maneras con las que una empresa puede hacer negocio utilizando software libre, ya sea desarrollándolo, integrándolo en otros productos o vendiendo servicios asociados. A continuación mencionaremos algunos ejemplos concretos.

Los ordenadores, y muy especialmente los servidores –ordenadores muy robustos destinados habitualmente a proporcionar servicios en red–, se venden normalmente con un sistema operativo instalado, de modo que el vendedor pueda certificar que funciona correctamente en su hard-

ware. La mayoría de fabricantes de hardware para servidores, como IBM o Sun Microsystems, ofrecen máquinas con alguna distribución de GNU/Linux pre-instalada y certificada. De esta manera dan a sus clientes un valor añadido –un sistema operativo libre y certificado– a un precio muy competitivo, ya que el cliente no tiene que pagar por licencias de software.

Los productores y distribuidores de software libre, como RedHat o MySQL, tienen modelos de negocio que habitualmente combinan diferentes fuentes de ingresos. La certificación, la venta de servicios asociados y el desarrollo a medida son algunas de las más habituales. Algunas empresas publican el software con licencias tipo *copyleft*, de modo que los programas que se deriven también tienen que ser libres, pero venden licencias especiales para quien quiera integrar su producto en un programa privativo. Otras empresas publican una versión libre del programa que proporciona la funcionalidad básica, y ofrecen ampliaciones privativas bajo licencia.

Muchas otras empresas hacen negocio con el software libre sin contribuir significativamente a su desarrollo. Por ejemplo, muchos proveedores de páginas web usan GNU/Linux y Apache en su negocio sin modificarlos de forma significativa. De esta manera reducen costes y no dependen de un único proveedor de software, ya que si necesitan un nuevo desarrollo o corregir un error en el programa lo pueden hacer ellos mismos o encargar el trabajo a otro. Los proyectos libres se benefician mucho de estos usuarios, pues hacen un uso intensivo del programa en entornos muy variados y pueden proporcionar información muy valiosa sobre su funcionamiento.

4.2. Sectores especiales

Hay dos sectores en los cuales el software libre cobra una importancia especial, independientemente de consideraciones económicas o de sus cualidades técnicas. En estos sectores son especialmente relevantes la transparencia, la independencia y las libertades que proporciona el software libre.

El primero de estos sectores son las administraciones públicas (AA. PP.). En el área de las TIC, se puede decir que las diferentes instituciones del Estado son el principal agente del mercado si las consideramos de forma conjunta. Las AA. PP. producen gran cantidad de programas

hechos a medida, ya sea de forma interna o por encargo, y también utilizan muchos programas ya existentes. Estos programas van desde los típicos paquetes de ofimática hasta grandes bases de datos, pasando por sistemas de seguridad para la protección de datos privados.

Cuando una institución pública desarrolla un programa, ya sea de forma interna o por encargo, es lógico que los resultados se publiquen bajo una licencia libre, de manera que cualquiera los pueda aprovechar. Al fin y al cabo, son los impuestos de todos los ciudadanos los que han pagado este desarrollo. Pero cuando se trata de programas ya existentes, también hay muchos motivos para que las AA. PP. escojan programas libres, como la transparencia, la seguridad o la independencia tecnológica. En estos momentos, por ejemplo, es muy posible que el historial médico del lector esté almacenado en una base de datos privativa, un programa que nadie puede saber exactamente lo que hace excepto la empresa que lo desarrolla. Es como si el Estado, antes de la informatización de estos archivos, hubiera encargado a una empresa privada que guardara los originales en una caja fuerte de la cual sólo la empresa tuviera la llave. En estas condiciones es imposible que las AA. PP. puedan garantizar la privacidad, integridad y durabilidad de los datos de los ciudadanos, tal como es su obligación, ya que ni siquiera pueden realizar una auditoría de los programas que utilizan para manipularlos y no almacenarlos. Una situación comparable sería la de una administración que aceptara un proyecto para construir un puente sin examinar los planos y renunciando a pedir la opinión de sus propios expertos.

Otra gran ventaja del uso de software libre por parte de las AA. PP. es la adaptabilidad y la independencia de los proveedores. En el caso de países como el nuestro, donde la protección de las lenguas propias es una de las obligaciones de las AA. PP., el software libre ofrece la posibilidad utilizar programas completamente traducidos. Y si la comunidad no ha completado la traducción de un programa, la propia Administración que lo necesita se puede encargar de hacerlo. Por contra, es muy frecuente que se utilicen programas privativos en otros idiomas, y que se tenga que pagar a la empresa que lo produce para que los traduzca. Los programas privativos mantienen a su cliente cautivo de la empresa que los produce, en nuestro caso, casi siempre empresas extranjeras. Por contra, el software libre favorece el crecimiento de un mercado local en el sector de las TIC, donde empresas de todas las medidas compiten para

ofrecer mejores servicios. Parece evidente cuál de las dos alternativas es la que más se ajusta a las obligaciones del Estado hacia sus ciudadanos.

Otro sector donde el software libre es especialmente relevante es la educación. Ya hace muchos años que las principales empresas de software privativo se dedican a promover el uso de sus productos en las escuelas, institutos y universidades haciendo publicidad, regalando ordenadores y ofreciendo licencias especiales de sus programas a los estudiantes a unos precios muy bajos. Estas estrategias de marketing están teniendo el efecto deseado: la gran mayoría de estudiantes utilizan programas privativos en el centro educativo, que de esta manera contribuye a convertirlos en futuros clientes de las empresas que los producen.

Por contra, el software libre permite que los estudiantes utilicen libremente los programas que aprenden a utilizar en la escuela. Por ejemplo, los pueden estudiar en clase, llevárselos a su casa sin vulnerar ninguna licencia y ayudar a traducirlos a su idioma. Además, el software libre fomenta los valores de la cooperación, el trabajo en equipo y el reconocimiento del trabajo de los demás.

En los últimos años han surgido muchas iniciativas para promover el uso del software libre en el sector educativo, como el proyecto Linkat [9] o el gnuLinux [10], que han llevado a cabo la tarea de adaptar una distribución de GNU/Linux para los centros educativos y las características de una región determinada. Igual que en el caso de las AA. PP., el software libre permite que los estudiantes trabajen en un entorno informático totalmente traducido, localizado y adaptado a sus necesidades concretas.

5. El futuro del software libre

Las tendencias actuales del mercado de las TIC permiten afirmar que el uso del software libre seguirá aumentando en los próximos años. El crecimiento de este sector ha hecho posible la aparición de un gran número de empresas grandes y pequeñas que ofrecen servicios en torno al software libre. Uno de los argumentos más repetidos en contra del uso de estas tecnologías, la supuesta falta de soporte técnico, es refutado a diario por miles de profesionales que se dedican a instalar, mantener y adaptar aplicaciones de software libre en todo tipo de entornos.

Uno de los ámbitos donde es más difícil la penetración del software libre es el doméstico. Eso es debido principalmente a tres factores: la resistencia al cambio que tienen todos los usuarios, las dificultades asociadas a instalar un sistema operativo diferente al que llevaba originalmente el ordenador, y los documentos almacenados en formatos de archivo privativos. La reciente aparición de una nueva línea de ordenadores portátiles de dimensiones reducidas -los llamados *netbooks*- se ha convertido en una inesperada palanca que puede abrir este mercado para el software libre: una tercera parte de estos dispositivos se venden con alguna distribución de GNU/Linux, adaptada para usuarios noveles, instalada ya de fábrica.

Quizás el sector donde el software libre tiene un futuro más prometedor, aparte de los servidores de Internet, es el de los sistemas integrados en dispositivos electrónicos, como teléfonos móviles, equipos multimedia, sistemas de navegación GPS, ordenadores integrados en los automóviles, etc. Estos sistemas se caracterizan por las limitaciones que impone el hardware, con unas prestaciones muy inferiores a las de un ordenador de sobremesa. Esto obliga a que el software que se instala esté perfectamente optimizado para aprovechar al máximo las características del hardware y consumir el mínimo de recursos. El software libre es ideal en este sentido, ya que permite que cualquiera haga modificaciones en él. Además, el precio de estos dispositivos suele ser más bajo que el de un ordenador, lo cual hace más significativo aún el coste de las licencias de software.

6. Conclusiones

El movimiento del software libre, en paralelo con el crecimiento de Internet, es uno de los pilares de la revolución

tecnológica que representan las TIC. Sin programas libres y estándares abiertos, Internet no sería la red adaptable, flexible y abierta que conocemos, y la brecha digital sería aún mayor. Muchas organizaciones, empresas e instituciones públicas utilizan cada día programas libres, por sus ventajas técnicas y económicas pero también por las libertades que confieren a sus usuarios. De las cuatro libertades fundamentales que definen el software libre se derivan muchas otras, como la libertad de escoger a un proveedor de servicios, la libertad de cambiar de aplicación sin cambiar el formato en el que se almacenan los datos, la libertad de utilizar los programas en la lengua propia, la libertad de construir un negocio sobre productos ya existentes, etcétera.

Pero los principios del movimiento del software libre trascienden el ámbito de la informática. Aparte de sus méritos como método de producción y distribución de software, los principios ideológicos de este movimiento están en sintonía con nuestra tendencia natural a comunicarnos y compartir conocimiento, a desarrollar la ciencia y la cultura aprovechando los cimientos construidos por otros antes que nosotros. ¿Qué sentido tienen las grandes redes de comunicaciones si se restringe el derecho a utilizarlas para compartir el conocimiento?

Desde finales de los años noventa han ganado empuje movimientos que promueven el libre intercambio en otras áreas, como la cultura, los materiales educativos o la ingeniería. Todos estos movimientos tienen en común la convicción de que el conocimiento no ha de ser sujeto de apropiación, cerrado tras licencias y patentes, sino un bien común, desarrollado de forma colectiva y del cual todos podamos disfrutar.

Referencias

- [1] LUI, C. E.; BLUM, J. R. (2004). *Time based hardware button for application launch*. United States Patent: 6727830.
<<http://www.freepatentsonline.com/6727830.html>>
- [2] LEVINE, F. E.; CARTER, B. (2000). *Stateless shopping cart for the web*. European Patent EP0807891.
<<http://www.freepatentsonline.com/EP0807891B1.html>>
- [3] *iNo a las patentes de software!*
<<http://www.nosoftwarepatents.com/es>>

- [4] *Resumen sobre las patentes de software - FFII*
 <<http://es.ffii.org/contraargumentos>>
- [5] SWADE, D. (2002). *The difference engine*. Londres: Penguin.
- [6] *El manifiesto GNU*. Free Software Foundation (FSF).
 <<http://www.gnu.org/gnu/manifiesto.es.html>>
- [7] *La definición de software libre*. Free Software Foundation (FSF).
 <<http://www.gnu.org/philosophy/free-sw.es.html>>
- [8] TORVALDS, L.; DIAMOND, D. (2001). *Just for fun*. Nueva York: Harper Collins.
- [9] Portal LinKat
 <<http://linkat.xtec.cat>>
- [10] gnuLinEx.org
 <<http://www.linex.org>>

Cita recomendada

JACOVKIS, David (2009). «El software libre: producción colectiva de conocimiento». En: «Software libre [monográfico en línea]. *IDP. Revista de Internet, Derecho y Política*. N.º 8. UOC. [Fecha de consulta: dd/mm/aa].

<Dirección electrónica del PDF>

ISSN 1699-8154



Esta obra está bajo la licencia Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons. Así pues, se permite la copia, distribución y comunicación pública siempre y cuando se cite el autor de esta obra y la fuente (*IDP. Revista de Internet, Derecho y Política*) y el uso concreto no tenga finalidad comercial. No se pueden hacer usos comerciales ni obras derivadas. La licencia completa se puede consultar en: <<http://creativecommons.org/licenses/by-nc-nd/3.0/es/deed.es>>

Sobre el autor

David Jacovkis
 david@freeknowledge.eu

David Jacovkis es licenciado en Física y ha cursado posgrados de Ciencia de los materiales (UAB) y de Software libre (UOC). Ha trabajado como administrador de sistemas, consultor en tecnologías de la información y editor de materiales educativos. Cofundador del Free Knowledge Institute, donde trabaja en diferentes proyectos relacionados con el uso del software libre en la educación. Consultor en el máster de Software libre de la UOC.

Free Knowledge Institute
 Wibautstraat, 150
 1091GR Amsterdam, Países Bajos