

# Documentación y software libre

**José Ramón Pérez Agüera**

**Rodrigo Sánchez Jiménez**

Universidad Complutense de Madrid (España)

## 0.1. Resumen

El término “software libre” se ha convertido recientemente en uno de los más utilizados en Internet. Sin embargo el uso que se hace de este término es muchas veces poco adecuado. En este artículo se define con exactitud el concepto, se analizan las ventajas de su utilización en la empresa y la universidad, y, por último, se reseñan sus principales aplicaciones al ámbito de la documentación.

**Palabras clave:** Software libre. Aplicaciones. Investigación. Empresa. Documentación.

## 0.2. Abstract

The term “free software” has lately become very usual on the Internet. Nevertheless, the use made of this term is very often inaccurate. In this paper the concept is defined, the main advantages of its use in the enterprise and university environments are indicated, and its main applications to the field of Information Science are sketched.

**Key Words:** Free software. Applications. Research. Enterprise. Information Science.

## 1. Introducción

Generalmente, cuando adquirimos un programa de ordenador, adquirimos una serie de obligaciones y derechos mediante la aceptación de una licencia de uso. Esto se resume fundamentalmente en que tenemos el derecho de utilizar el software bajo condiciones determinadas —independientemente de que estas sean más o menos restrictivas— y que el vendedor nos proporciona una serie de garantías o servicios adicionales en algunos casos.

En el carácter de estas obligaciones y derechos expresados en la licencia de uso es donde reside la diferencia entre el software libre y el software propietario. La diferencia fundamental entre ambos tipos de publicación de software reside en que en el caso del software propietario los usuarios no tienen derecho a modificar

el software (independientemente de los motivos), por lo que no se les proporciona el código fuente (1); y no tienen derecho a efectuar copias del mismo, salvo aquellas destinadas a la preservación del programa para el propio uso, su utilización para un mismo usuario en varias máquinas, etc.

Podemos ofrecer una visión más precisa de lo que significa software libre, para lo cuál utilizaremos la definición de Richard Stallman a este respecto, quien postula que el software libre lo es si el usuario (Stallman, 1998) tiene libertad para: ejecutar el programa con cualquier propósito; modificar el programa para adaptarlo a sus necesidades; redistribuir copias, tanto gratis como por un canon; y distribuir versiones modificadas del programa, de tal manera que la comunidad pueda beneficiarse con sus mejoras. Por tanto la diferencia fundamental entre el software libre y el propietario reside no tanto en la gratuidad de los programas como en la libertad para distribuirlos y modificarlos con cualquier objetivo. La forma en la que se reflejan los derechos de copia y distribución en la legislación española es esta (España, Ministerio de Cultura, 1996):

[...] cuando se produzca cesión del derecho del uso de un programa de ordenador, se entenderá, salvo prueba en contrario, que dicha cesión tiene carácter no exclusivo e intransferible, presumiéndose, asimismo, que lo es para satisfacer únicamente las necesidades del usuario. La primera venta en la Unión Europea de una copia de un programa por el titular de los derechos o con su consentimiento agotará el derecho de distribución de dicha copia, salvo el derecho de controlar el subsiguiente alquiler del programa o de una copia del mismo.

Esta perspectiva legislativa se adapta bastante bien al ámbito y los objetivos del denominado software propietario, aunque no se adapta tanto a las del software libre. Existen múltiples tipos de licencias que tienen cabida bajo la denominación común de software libre; y la categoría de la legislación española que mejor se adapta a ellas es la de una obra colectiva (Usano, 2001). Bajo esta perspectiva es posible producir software libre con bastante tranquilidad.

Desde la perspectiva del usuario de software libre existen todavía algunas matizaciones que hacer. Es importante no confundir software libre con software gratuito, error que se fundamenta en la traducción del término originario del inglés *free software*, ya que *free* significa al mismo tiempo libre y gratis. El software libre no tiene por qué ser gratuito, y es más, es frecuente que se pague por obtenerlo. Debido a los derechos de distribución y modificación adquiridos por el usuario es muy fácil encontrar copias adaptadas a múltiples sistemas de los programas originales, muchas veces de forma gratuita, por lo que surge una pregunta: ¿por qué comprar software libre? Fundamentalmente, para cubrir tres necesidades: instalación, personalización y mantenimiento.

Otro punto que es necesario explicar es la diferencia entre el software libre, el *shareware* y el *freeware*. En este sentido éstos últimos no proporcionan el código

fuente ni el derecho a modificarlo. Un programa freeware garantiza el derecho a distribuir copias del mismo, lo que está expresamente prohibido en la mayor parte del software propietario, así como el derecho para usarlo de forma gratuita. La fórmula del *shareware* es la que se utiliza generalmente para distribuir versiones limitadas en el tiempo o la funcionalidad — *trials*, *demos*, versiones de prueba, etc. — de software propietario al uso. Generalmente, una vez transcurre un determinado periodo de tiempo, es necesario elegir entre pagar una suma para seguir utilizándolo o desinstalarlo.

Como ya hemos adelantado existen multitud de licencias que tienen cabida bajo el marco del software libre, aunque existen algunos matices que es importante tener en cuenta. Un ejemplo de esto lo constituyen las importantes diferencias entre el software libre y el software de código abierto u *open source*. Esta última categoría de software se deriva de los esfuerzos de una parte de la comunidad de usuarios y creadores de software libre por generar una alternativa altamente comercial para el desarrollo de software. La diferencia fundamental estriba en que dentro del software de código abierto caben cosas como el software semi-libre — para el que se establece una retribución económica sólo en el caso de que el usuario sea una organización con ánimo de lucro — o la capacidad de variar los términos de la licencia original de forma que el software termine por ser propietario. Bajo el punto de vista del software libre cualquier cosa que utilice en todo o en parte software libre debe ser software libre a su vez. Sin embargo los impulsores de la iniciativa por el software abierto estiman que ésta es una desventaja comercial que es necesario superar, por lo que las licencias de este estilo permiten que el software termine por convertirse en propietario, impidiendo su posterior modificación o distribución por parte de terceros. Probablemente la licencia más conocida dentro del panorama del software libre sea la *General Public Licence* (GPL) (Free Software Foundation, 1991). Lo importante de esta licencia es que está protegida por copyleft siendo imposible cambiar la licencia al programa, al distribuirlo tal cual o incluso cuando hemos introducido modificaciones sustanciales en él. La contrapartida de software de código abierto es la *Open Software Licence* (OSL), muy utilizada en ámbitos comerciales (Rosen, 2004), aunque existen multitud de licencias generadas por empresas como Sun, Apple, IBM o Nokia que también entran dentro de la categoría.

De todo esto se extraen tres conclusiones: a) Si somos usuarios sin ánimo de lucro, no tenemos por qué preocuparnos por estas disquisiciones desde un punto de vista práctico e inmediato — podemos utilizar cualquier software independientemente de que sea libre, de código abierto o semiprivado con todas las ventajas que esto supone —. b) Si somos un usuario con ánimo de lucro (como una empresa relacionada con los servicios de documentación), deberíamos preocuparnos por el tipo de licencia del software que adquirimos, ya que es posible que

tengamos que pagar un canon en alguno de los casos. c) Si vamos a desarrollar software — como por ejemplo en una universidad como resultado de una investigación o en una empresa para resolver algún tipo específico de necesidades —, será muy importante que reflexionemos acerca de la necesidad de acogernos a uno u otro tipo de licencia.

## **2. Ventajas del software libre para investigación, empresa y universidad**

Creemos que la utilización de software libre reviste un número importante de ventajas a tener en cuenta. Sin embargo los motivos por los que decidiremos — o no — utilizar y crear aplicaciones basadas en software libre son distintas para el ámbito académico y el empresarial, por lo que las expondremos por separado:

### **2.1. Software libre en el ámbito empresarial**

El motivo fundamental detrás de la creación de la Open Source Initiative y del surgimiento del concepto de Software de Código Abierto es precisamente la instauración de un modelo de negocio distinto, aunque funcional, para el campo de la producción de software. En esta línea, existen algunos motivos por los que tradicionalmente se escoge software propietario para el ámbito empresarial (Di Cosmo, 2003). Un primer factor tiene que ver con la responsabilidad. En cualquier empresa existe una cadena de responsabilidades que determinan quién se hace cargo de la situación resultante si el software falla. Las aplicaciones no comerciales de software libre proporcionan soporte comunitario, lo que escapa al concepto de responsabilidad antes mencionado. Sin embargo, las compañías comerciales de software libre ofrecen soporte a sus clientes. Otro motivo es el dominio de Microsoft en el ámbito comercial, de su sistema operativo y de algunas aplicaciones esenciales como los navegadores o reproductores de medios. Esto se debe en gran parte a que es además el sistema operativo preinstalado en la mayoría de las máquinas. Instalar una alternativa libre significa por lo general un proceso de aprendizaje. Un tercer factor es la relación entre el coste de un producto y la calidad esperada. Generalmente el principio de “lo barato sale caro” es de aplicación en casi cualquier aspecto de la vida. En el ámbito del software esto no es así. El hecho de que un producto sea más caro no significa que sea mejor, aunque no todo el software libre es gratuito. Finalmente, otro motivo aducido es la solidez de las empresas. Se supone que el modelo de empresa que proporciona productos libres o de código abierto es más inestable que las empresas de software tradicional. Esto no es cierto. Sun, Nokia, Redhat y un largo etcétera han optado por mantener un modelo empresarial ligado al software libre o por mantener divisiones sostenidas en estos principios. Estas empresas no tienen más tendencia a la quiebra que las empresas de software propietario. En un caso extremo uno siem-

pre puede hacerse cargo de la actualización, mejora y mantenimiento de un software de cuyo código dispone, algo que no ocurre con el software propietario.

Por otra parte, existen una serie de motivos que se esgrime a la hora de justificar la decisión de optar por software libre, como son sus capacidades de adaptación al modelo concreto de nuestra empresa o la conformidad a las normas y estándares. En este último sentido, conviene señalar que el software propietario, dependiendo de los casos, tiene tendencia a generar desarrollos al margen de los estándares y a hacer incompatibles sus productos con los de otros fabricantes, como medio de proteger su mercado. Este es el caso de OpenGL, por poner un ejemplo. Esto está muy relacionado también con los altos niveles de competencia entre las empresas, lo que les lleva a intentar excluir a sus adversarios, desbancándolos mediante la adopción una tecnología superior que hace obsoleta la de los adversarios. Evidentemente esto genera iniciativas divergentes, cuando no diametralmente opuestas. Un ejemplo muy importante en contra son los acuerdos alcanzados en el seno del World Wide Web Consortium (W3C). Otra razón a favor es la rápida detección de bugs o fallos de programación y la consiguiente posibilidad de corrección inmediata. Sólo hay que observar al respecto las diferencias entre la rapidez de respuesta de los servicios de actualización de seguridad Windows XP y Debian GNU/Linux. Por fin, otro conjunto de motivos tienen que ver con el propio desarrollo general de tecnologías, ya que las empresas propietarias de tecnologías clave pasan a tener un control importante de los sectores en los que éstas son de referencia; sobre todo, si sus tecnologías están en la base del desarrollo de otras. Incluso los que utilizan software pirata pagan en cierto sentido por el uso de software propietario, ya que siguen estando en una posición de dependencia con respecto al oferente.

¿Pero qué ofrecen las empresas de software libre si no pueden restringir la libre circulación de su código? El modelo de negocio basado en software libre ofrece consultoría, adaptación, servicio de atención, formación y un largo etcétera de servicios añadidos por los que sí cobra a sus clientes. La base del negocio se fundamenta por tanto en ofrecer algo más que la copia del programa (que siempre es susceptible de circular de forma ilegal — y por tanto no remunerada— entre los usuarios).

## 2.2. Ventajas del software libre para la investigación y la universidad

El software libre tiene su origen en ámbitos científicos y de investigación, lo que no es de extrañar, ya que se adapta a la perfección a los modelos de desarrollo típicos de la Ciencia, que requiere la posibilidad de colaboración en el trabajo y de aprovechamiento de los progresos realizados (factor acumulativo). De esta forma, nace de forma natural para aquellos que consideran la programación o el software como herramientas fundamentales para el desarrollo de su labor de investigación.

Digamos que la concepción de compartir el trabajo y los conocimientos ya existía en la Ciencia antes de que se comenzara a desarrollar y distribuir software, y que la llegada de éste a la tarea cotidiana del científico bajo condiciones restrictivas de distribución y uso chocó contra una costumbre firmemente establecida.

Fue precisamente en ámbitos científicos y de investigación en los que se comenzaron a organizar iniciativas que promovían el uso y el desarrollo del software libre; de manera que el origen del proyecto de software libre más importante hay que buscarlo precisamente en el MIT, y en uno de sus empleados, Richard Stallman, que decide encauzar los esfuerzos que él y sus compañeros ya habían llevado a cabo en los laboratorios del Instituto Tecnológico de Massachussets hacia la creación de un sistema operativo libre a principios de los años ochenta. Este proyecto, denominado GNU, pretende la creación de la base necesaria para el desarrollo ilimitado de software libre para cualquier aplicación. El nombre del proyecto proviene de un complejo juego de palabras de programadores, que hace referencia al sistema operativo en el que está inspirado (Unix), y a la condición de software libre que no comparte con éste. GNU significa *GNU is not Unix*. Posteriormente, se creó la Free Software Foundation, que proporciona soporte legal, administrativo, y organizativo al proyecto, así como a otras iniciativas que desarrollan software libre.

Al contrario que en el ámbito empresarial, donde creemos que subsisten argumentos que justifican tanto la opción propietaria como la opción del software libre, en el caso de la universidad y la investigación la idoneidad de la utilización de software libre parece muy clara. Simplemente, se adapta mejor a las necesidades de este ámbito. Al igual que los resultados de cualquier investigación del ámbito universitario se publican en revistas científicas para que obtengan la mayor difusión posible, y sean criticados y mejorados por la comunidad científica, debería ocurrir lo mismo con los avances en investigación que impliquen la utilización o generación de software. Es imposible criticar y mejorar una aplicación resultado de los esfuerzos de un grupo de investigación en el ámbito de la indización automática si no se puede estudiar el código fuente (por poner un ejemplo). Debido a que los esfuerzos de los distintos programadores se van acumulando a lo largo del tiempo, un software generado por investigadores del ámbito universitario puede ser mejorado debido a que cualquiera puede modificar el código original o uno ya modificado. Esto es, el factor acumulativo de la ciencia se debe poder aplicar a las investigaciones en las que el software utilizado tenga cierta importancia. Además, disponer del código fuente del programa nos permite ir más allá en nuestras investigaciones, adaptarlo a nuestras necesidades específicas.

Desde el punto de vista de la educación —la otra esfera de preocupación del ámbito universitario—, la utilización de software libre parece inexcusable por dos motivos. En primer lugar, facilita el acceso rápido a una versión completa-

mente funcional del software, sin pasar por pasos previos como las versiones de evaluación o la firma de contratos, etc. No sólo tenemos garantizada la utilización del software en prácticas, sino que podremos decidir sin necesidad de atravesar ningún otro trámite si nos conviene sobre la base de la aplicación real, y no de una versión de capacidades reducidas. En segundo lugar, podemos estudiar el propio software y aprender de él. Es fundamental tener acceso al funcionamiento del propio software para que éste no se convierta en una mera caja negra. Es mucho más interesante desde el punto de vista pedagógico hacer patentes los principios en los que se basa que limitarse a utilizarlo.

### 3. Software libre para documentalistas

Todo lo dicho con anterioridad se aplica por igual a una empresa y a una universidad, ámbitos en los que se producen las actividades propias del área de la documentación. Sin embargo, puede ser de utilidad enfocar más la cuestión y señalar algunos ejemplos de utilidad específica para el campo de la documentación. No incluimos software exclusivo de bibliotecas, que es probablemente el más abundante, aunque podemos proporcionar una buena referencia para todo lo que tiene que ver con este ámbito: Open Sources System for Libraries.

Existen algunos ejemplos de campos en los que los documentalistas hacen uso extensivo de aplicaciones propietarias, pero para las que existen alternativas libres. Un primer campo son los Sistemas de Gestión de Bases de Datos (SGBD), donde MySQL (MySQL AB, 1995-2005) y PostgreSQL (PostgreSQL Global Development Group, 1996-2005) son alternativas válidas para sistemas ligeros-medianos y medianos-pesados respectivamente. Son muy conocidos y utilizados en el ámbito de Internet. Una buena forma de conseguir MySQL —sin duda uno de los sistemas de bases de datos más comunes hoy día— es mediante la aplicación AppServ (AppServ Open Project, 2005) para Windows, que incluye un servidor web y un gestor para esta base de datos.

Otro campo son los sistemas de portales y de gestión de contenidos (CMS), donde destacan PostNuke (Haakenson, 2005), Plone (Limi, Runyan y Andersen, 200-2005), Typo3 (TYPO3 Association, 2002-2005), Jboss Portal (JBoss Inc, 2002-2005) o Mambo (Miro International Pty, 2000-2005). Por su parte, entre los sistemas complejos para el soporte de CMS, hay que señalar los servidores de aplicaciones Jetspeed (Apache Software Foundation, 2000-2005), Zope (Zope Corporation, 2003-2005.) y Jboss Application Server (JBoss Inc, 2002-2005).

En el campo de las bibliotecas virtuales, Phronesis (Tecnológico de Monterrey, 2000-2005) y Greenstone (New Zealand Digital Library Project, 2000-2005) son las dos alternativas más conocidas para generar desde cero una biblioteca virtual.

En el área de los sistemas de recuperación de información destacan Isite, “Java Search Engine” de Javaproxy y NGO Search.

Finalmente en el campo de los sistemas de publicación de revistas electrónicas, el sistema OJS (Public Knowledge Project 2002-2005) es probablemente uno de los mejores. Una de las ventajas de este sistema es que está basado en el estándar Dublin Core para metadatos y en el Open Access Initiative para su disseminación. En su página web se pueden consultar referencias a otras iniciativas similares.

Estos son sólo algunos de los ejemplos más comunes, ya que existe una miríada de aplicaciones para los más diversas necesidades documentales. Sin embargo un buen sitio por el que comenzar es el heterogéneo aunque muy interesante directorio de recursos del portal *R020 Bibliotecología y Ciencias de la Información* (Ferreira, 2005).

#### 4. Notas

- (1) Un programa de ordenador es diseñado en un lenguaje que los programadores comprenden, aunque para que éste programa sea funcional es necesario convertirlo al lenguaje de la máquina que lo va a ejecutar. La práctica habitual del software propietario es la de proporcionar sólo la versión ejecutable por la máquina (binarios), y no la versión legible por humanos (código fuente), lo que en la práctica hace secreto el contenido y la estructura de los programas de ordenador, e impide su modificación.

#### 5. Referencias

- Apache Software Foundation (2000-2005). Apache Portals: Jetspeed. Apache Software Foundation 2000-2005. URL: <<http://portals.apache.org/jetspeed-1/>>.
- AppServ Open Project (2005). AppServ Open Project. Chiang Mai (Tailandia): AppServ Open Project. 2005. . URL: <<http://www.appservnetwork.com/>>.
- Di Cosmo, Roberto (2003). Software Libre para las empresas. Di Cosmo, Roberto. 2003. URL: <<http://www.pps.jussieu.fr/~dicosmo/TALKS/Cordoba2003.pdf>>.
- España. Ministerio de Cultura (1996). Título VII. Programas de Ordenador. // Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el Texto Refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia. // BOE de 22 Abril 1996. Madrid: Boletín Oficial del Estado, 1996. URL: <<http://www.boe.es/g/es/boe/dias/1996-04-22/seccion1.php#00002>>.
- Ferreira, Diego (2005). R020 Bibliotecología y Ciencias de la Información. [S. l.: Diego Ferreira, 2005]. URL: <<http://www.089020.com.ar/recursos.php?total=total>>.
- Free Software Foundation (1991). GNU General Public License. Versión 2ª. Massachussets: Free Software Foundation, 1991. URL: <<http://www.gnu.org/copyleft/gpl.es.html>>.
- Haakenson, Vanessa (2005). PostNuke: Content Management System. PostNuke, 2005. URL: <<http://www.postnuke.com/>>.

- JBoss Inc (2002-2005). JBoss Application Server. JBoss Inc, 2002-2005. URL: <<http://www.jboss.org/products/jbossas>>.
- JBoss Inc (2002-2005b). JBoss.com: The Professional Open Source Company. JBoss Inc, 2002-2005. URL: <<http://www.jboss.org/products/jbossportal>>.
- Limi, Alexander; Runyan, Alan; Andersen, Vidar (200-2005). Plone: A user friendly and powerful open source Content Management System. Plone, 2000-2005. URL: <<http://plone.org/>>.
- Miro International Pty (2000-2005). Mambo. Miro International Pty, 2000-2005. URL: <<http://www.mamboserver.com/>>.
- MySQL AB (1995-2005). MySQL: The World's Most Popular Open Source Database. Uppsala: MySQL AB, 1995-2005. URL: <<http://www.mysql.com>>. Consultado: 2005-05-19.
- New Zealand Digital Library Project (2000-2005). Greenstone Digital Library Software. Nueva Zelanda: University of Wikato, 2000-2005. URL: <<http://www.greenstone.org/>>.
- Open Sources System for Libraries. [S. l.]: The oss4lib Community, 1999-2005. URL: <<http://oss4lib.org/>>.
- PostgreSQL Global Development Group (1996-2005). PostgreSQL: The world's most advanced open source database. PostgreSQL Global Development Group, 1996-2005. . URL: <<http://www.postgresql.org/>>.
- Public Knowledge Project (2002-2005). PKP Open Journal Systems. Vancouver: University of British Columbia, 2002-2005. . URL: <<http://www.pkp.ubc.ca/ojs/>>.
- Rosen, E. Lawrence (2004). Open Software License v. 2.1. Open Source Initiative, 2004. URL: <<http://opensource.org/licenses/osl-2.1.php>>
- Stallman, Richard (1998). El Proyecto GNU. [en línea]. Free Software Foundation. URL:<<http://www.gnu.org/gnu/thegnuproject.es.html>>. Consultado: Febrero 2004.
- Tecnológico de Monterrey (2000-2005). Phronesis Biblioteca Digital. Monterrey (México), Tecnológico de Monterrey, 2000-2005.. URL: <<http://copernico.mty.itesm.mx/phronesis/project/>>.
- TYPO3 Association (2002-2005). TYPO3 Open Source management: the web in your hands. TYPO3 Association, 2002-2005. URL: <<http://typo3.com/>>.
- Usano, Pablo (2001). El "Software Libre" como obra Colectiva. GNU España, 2001. URL: <<http://es.gnu.org/licencias/softlibr.html>>. Consultado: Enero 2005.
- Zope Corporation (2003-2005). Zope.org. Zope Corporation, 2003-2005. URL: <<http://www.zope.org/>>.