



El concepto de software libre

Jesús M. González-Barahona

RESUMEN

El software libre es cada vez más habitual en casi cualquier entorno informático. Sin embargo, es también un gran desconocido. En este artículo se exponen las condiciones que tiene que cumplir un programa para ser considerado como software libre, compiladas en varias definiciones. A partir de ellas se repasa brevemente su historia y se exploran algunas de sus características y consecuencias. También se comentan algunos aspectos relacionados con las licencias de software libre que son tan fundamentales para su existencia.

Palabras clave: software libre, software de fuentes abiertas, licencias de software, comunidades de desarrollo de software

RESUM (*El concepte de programari lliure*)

El programari lliure és cada vegada més habitual en qualsevol entorn informàtic. Tanmateix, és també un gran desconegut. En aquest article s'exposen les condicions que ha de complir un programa per a ser considerat com a programari lliure, compilades en diverses definicions. A partir d'elles es repassa breument la seva història i s'exploraven algunes de les seves característiques i conseqüències. També es comenten alguns aspectes relacionats amb les llicències de programari lliure que són tan bàsiques per a la seva existència.

Paraules clau: programari lliure, programari de codi obert, llicències de programari, comunitats de desenvolupament de programari

ABSTRACT (*The notion of free software*)

Free, open source software is increasingly more usual in almost any computing environment. However, it is also a great unknown: not many people knows what it truly is. This paper presents the conditions that some piece of software has to comply with to be considered as free or open source software, which are compiled in several definitions. Starting from them, its history is briefly exposed, and some of its characteristics and consequences are explored. In addition, some aspects related to free software licensing are commented, since they are so fundamental for its very existence..

Keywords: free software, open source software, software licences, software development communities



1. Introducción

Hace ya tiempo que el software libre dejó de ser algo marginal, para convertirse en una realidad muy habitual para muchos usuarios. Incluso aunque no sepan que están usándolo, cada vez más usuarios realizan gran parte de sus tareas informáticas gracias a él. Bien sea porque lo tienen instalado en su ordenador de escritorio o en su móvil, porque lo usan intensivamente las empresas que les dan servicios vía web, o porque hace que funcionen sus aparatos electrónicos, el software libre es una realidad ubicua.

El navegador Firefox o el entorno ofimático LibreOffice/OpenOffice son, por ejemplo, programas libres populares que funcionan en muchos entornos de escritorio. También hay sistemas completamente o casi completamente libres, como Debian o Ubuntu, que poco a poco van siendo más conocidos. En el caso de los móviles, es bien conocido que Android está compuesto en gran medida por software libre. La infraestructura de grandes empresas que proporcionan servicios vía web, como Facebook, Google o Yahoo está compuesta en gran medida por programas libres. Muchos sintonizadores/grabadores de TDT o bases WiFi funcionan gracias a sistemas en gran medida libres, basados en Linux.

En resumen, es muy difícil que cualquiera de nosotros tenga una vida normal, y no se haya encontrado software libre al menos varias veces al día.

Y sin embargo, muchas veces se nos escapa qué tienen de especial estos programas. El objetivo de este artículo es justamente aproximar la realidad del software libre al lector, presentando con cierto detalle algunas de sus características principales. Aunque este texto se está publicando en un contexto dirigido al mundo de la traducción, no se pretende aquí entrar en el detalle de cómo se puede utilizar software libre para la traducción, o cómo se traducen los programas libres. Ambos temas se tratarán, sin duda mucho mejor de lo que este autor podría haberlo hecho, en otros artículos de este mismo número.

2. Definición

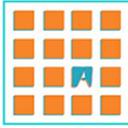
El concepto de software libre es fundamentalmente legal: es un software con el que se pueden hacer cierto tipo de cosas, porque su autor da permiso para ello.

Antes de entrar en más detalles, es conveniente recordar que los derechos de autor en el caso del software funcionan, de cara a los usuarios finales, de forma similar a como funcionan con las obras literarias o musicales. Si no tienes permiso explícito del dueño del programa, no puedes hacer casi nada con él. Desde luego no puedes repartir copias de él. Pero tampoco puedes, por ejemplo, modificarlo, ni siquiera para corregir un error que hayas encontrado. Todos los que utilizan software no libre (privativo) suelen conocer esta situación: si quieres un programa, tienes que “comprarlo” (en realidad, comprar una licencia de uso). Y cuando lo has hecho, puedes usarlo como te permita la licencia, y poco más.

En el caso del software libre, sin embargo, el dueño del programa (normalmente quien lo ha hecho) tiene una visión bastante diferente sobre lo que los usuarios deberían poder hacer. Por eso, si recibes un programa libre, el autor te está permitiendo que:

- lo uses como mejor te parezca
- puedas estudiar cómo funciona, y modificarlo si quieres
- lo redistribuyas a quien quieras
- distribuyas copias modificadas, si quieres

Estas son las llamadas “cuatro libertades” del software libre. Estas libertades se plasman en la licencia que acompaña al programa. En ella, el autor garantiza a quien reciba el programa que puede ejercerlas. Por eso, en el fondo, lo que define si un programa es libre o no es este detalle legal: que la licencia garantice estas libertades.



Para poder ejercer las cuatro libertades es preciso recibir el código fuente (algo así como los planos del programa), que permite hacer modificaciones de forma cómoda. De esta característica del software libre viene su otro nombre en inglés, “open source software”. En español, es común verlo traducido como “software de fuentes abiertas” o “código abierto”.

Este “segundo nombre” del software libre fue propuesto porque había quien percibía que “free software” (el nombre original) sugería “software gratis”, dado que en inglés “free” significa tanto “gratis” como “libre”, y por ello no era apropiado para introducir el concepto en empresas que estaban interesadas en obtener beneficios con su producción. Esto dio lugar a una cierta división en la comunidad del software libre, hasta el punto de que hay quien quiere ser identificado como parte de la “open source software community”, mientras que otros insisten en que la suya es la “free software community”. En cierto sentido, los primeros suelen estar más interesados en las consecuencias prácticas (de negocio, técnicas) del modelo de software libre, mientras que los segundos suelen valorar más los aspectos éticos, las ya mencionadas libertades. En la práctica, cada individuo tiene su propia combinación de motivos, normalmente en algún punto intermedio entre estos dos extremos.

Sin embargo, a pesar de estas diferencias, el software al que se refieren los dos términos es, a efectos prácticos, casi el mismo. Así podemos constatarlo si comparamos las tres definiciones detalladas más habituales: la Free Software Definition¹ de la Free Software Foundation, la Debian Free Software Guidelines² del proyecto Debian y la Open Source Software Definition³ de la Open Source Initiative.

Es interesante ver cómo en las comunidades de desarrollo de software libre se pueden mezclar ambas sensibilidades. En muchas de ellas pueden encontrarse desarrolladores con motivaciones muy diversas: voluntarios que colaboran por motivos éticos, otros que están fundamentalmente interesados en el desafío técnico, otros que buscan recompensa económica... Y todos ellos colaborando para realizar el mismo programa. Hay pocos otros ámbitos donde se vean situaciones semejantes.

Es importante destacar cómo las definiciones de software libre no hacen ninguna referencia al precio a que puede conseguirse el programa. Como ya se ha indicado, la palabra “free” en “free software” no tiene que ver con el precio, sino con las libertades que pueden ejercer quienes lo reciben. El software libre y el software gratuito son por tanto cosas bien distintas. Pero dicho esto conviene recordar también que debido a la tercera libertad cualquiera puede redistribuir un programa sin pedir contraprestación económica ni permiso. Esto hace prácticamente imposible obtener grandes ganancias simplemente por la distribución de programas libres: cualquiera que los obtenga puede a su vez redistribuirlos a precio más bajo, o incluso gratis. De esta forma, el precio de redistribución tiende hacia el coste marginal de copia.

Aún así hay modelos de negocio basados precisamente en vender software⁴⁵⁶, porque son muchas las circunstancias donde el consumidor está dispuesto a pagar si recibe ciertas contraprestaciones, como por ejemplo una cierta garantía, aunque sea subjetiva, sobre el software que recibe, un valor añadido en forma de selección, actualización y organización de un conjunto de programas, o un acceso a servicios adicionales (como mejoras y mantenimiento).

¹Free Software Definition <http://www.gnu.org/philosophy/free-sw.html>

²Debian Free Software Guidelines http://www.debian.org/social_contract#guidelines

³Open Source Software Definition <http://www.opensource.org/docs/osd>

⁴Economic aspects and business models of Free Software <http://ftacademy.org/materials/fsm/5>

⁵The emerging economic paradigm of open source <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/viewArticle/1470/1385>

⁶Free/Libre Open Source Software: a guide for SMEs <http://guide.conecta.it>



3. Un poco de historia

Para entender cómo es el software libre hoy, es bueno conocer un poco de su historia. Cuando nació el software, en las décadas de los 1950 y 1960, los programas no eran algo que se comercializase por separado. Se compraba un ordenador (en la época, normalmente un “mainframe” o similar), y con él se tenía acceso al catálogo de programas del fabricante, y a otros realizados por usuarios. Porque era habitual que si conocía a alguien que tenía un programa que me interesaba, se lo pidiera y simplemente me daba una copia. Por supuesto yo podía mejorar ese programa, y a su vez distribuir las modificaciones a terceros.

Durante los últimos 1960, y sobre todo durante los años 1970, la situación cambió radicalmente. El software pasó a ser un sujeto económico por sí mismo, y nacieron empresas interesadas en comercializar sólo programas, y no los equipos en los que funcionaban. Muchas de estas empresas eligieron modelos de negocio basados en vender licencias de uso de sus programas, a la vez que trataban de restringir por todos los medios su redistribución y modificación. El ámbito de los derechos de autor (copyright) se amplió a los programas de ordenador, y la idea predominante en la industria del software fue “controla al máximo tu programa, limita al máximo lo que otros puedan hacer con él”.

Como reacción a esta situación, en los años 1980 varios grupos comenzaron a poner los pilares del software libre. En particular, Richard Stallman⁷, que había sido programador en el MIT, tuvo un papel fundamental. Por un lado, sentó las bases filosóficas del software libre, escribiendo varios ensayos sobre el tema y fundando la Free Software Foundation. También contribuyó a las bases legales, colaborando en la escritura de una de las primeras licencias de software libre, la GPL. Y en que el software libre fuera una realidad, construyendo varios programas (un editor, un compilador, un depurador) que fueron el origen del proyecto GNU, una de las primeras comunidades de desarrollo de software libre. Otros grupos que tuvieron también un papel fundamental fueron los participantes en la comunidad Unix BSD⁸, que también usaban licencias libres para sus mejoras y ampliaciones del que entonces era el mejor sistema operativo disponible, y los desarrolladores de la primera Internet⁹, que permitieron que las primeras comunidades de software libre dispusieran de las herramientas de coordinación que necesitaban.

La década de 1990 fueron años de crecimiento. Linux (que nació hacia 1992, de la mano de Linus Torvalds) se estableció como una opción viable en el mercado de sistemas operativos, sobre todo en servidores. El software libre se convirtió en la opción preferente para proporcionar servicios web. Comenzaron a aparecer sistemas de escritorio completos basados en software libre, y grandes empresas incluyeron el software en su estrategia.

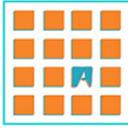
La última década ha sido sin duda la de la madurez. El software libre ha penetrado en prácticamente todos los nichos informáticos, convirtiéndose en una opción habitual en varios de ellos (servidores, sistemas empujados, o últimamente, móviles). Es difícil encontrar una empresa que no haya al menos probado las soluciones libres, y muchas de ellas están montando sus sistemas informáticos teniéndolas en cuenta en todos los lugares posibles.

Hoy el software libre es una opción a considerar, que en general se valora no solo por su coste, sino también por las ventajas a medio y largo plazo que puede ofrecer. Aún así, sigue siendo relativamente desconocido en muchos entornos, y muchos de sus usuarios potenciales desconocen los detalles del modelo lo suficiente como para beneficiarse de ellos.

⁷Free as in Freedom: Richard Stallman's Crusade for Free Software <http://oreilly.com/openbook/freedom/>

⁸Berkeley Software Distribution (Wikipedia) http://en.wikipedia.org/wiki/Berkeley_Software_Distribution

⁹Where wizards stay up late <http://www.chick.net/wizards/wizards.html>



4. Algunas consecuencias para los usuarios

¿Y cuáles son estos detalles? Veamos a continuación, sin pretender ser exhaustivos, algunos de ellos.

Quizás el principal efecto para el usuario es la aparición de mecanismos de verdadera competencia en el mercado. Hay que tener en cuenta que en gran medida los mercados de aplicaciones informáticas tienden al monopolio de producto. Esto es, una gran parte de los usuarios encuentran ventajas en utilizar el producto mayoritario. Hay muchas razones para ello: por ejemplo, la tendencia a utilizar el mismo software que la gente de nuestro entorno, la facilidad para encontrar documentación y expertos en él, o la simple comodidad que produce en muchos casos usar “lo mismo que todo el mundo”. En el mundo del software privativo esto supone automáticamente la aparición de un monopolio de empresa, pues cada producto tiene un solo proveedor último, su fabricante. En el mundo del software libre, sin embargo, cualquier empresa puede proporcionar un producto libre, y si quiere, proporcionar servicios para él, sin ninguna restricción: el código fuente de la aplicación está disponible para cualquier empresa que quiera trabajar con él.

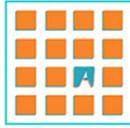
Es destacable también cómo en el software libre es muy fácil realizar pruebas con el producto real. En la mayoría de los casos, basta con bajarlo de Internet, algo que muy habitualmente está incluso automatizado en los canales de descarga de las distribuciones de software libre. Se pueden probar distintos programas con funcionalidad similar, elegir el que más se adapte a las necesidades, y si más adelante se prefiere otro, el cambio es también simple. Y todo ello sin tener que pagar licencias, ni firmar acuerdos con los distribuidores.

Desde un punto de vista más estratégico, el software libre permite independizarse en gran medida de la estrategia del productor. En el caso del software privativo, solo el productor puede hacer evolucionar un producto, por lo que el usuario depende en gran medida de lo que este productor decida. Sin embargo, en el software libre por un lado tenemos la competencia mencionada anteriormente, que hace que si una empresa no hace avanzar un producto por el camino que desean sus usuarios, aparezcan incentivos para que lo haga otra. Y no hay que olvidar que, en el fondo, los ingresos de estas empresas provienen de las labores de mantenimiento y mejora que realizan para los usuarios. Este es un mecanismo muy directo para asegurar que las mejoras y cambios del programa vayan en la línea que desean estos usuarios.

Algunas otras consecuencias destacables son las claras posibilidades de conseguir adaptaciones y mejoras del producto, con costes relativamente bajos, la posibilidad de resolver los problemas que pueda plantear la aplicación, o de contratar a alguien que los resuelva, modificando el programa, o conseguir auditorías completas, realizadas por partes imparciales, sobre cualquier producto que se quiera utilizar.

Por supuesto no todo son ventajas, también hay problemas. Gran parte de ellos provienen del hecho de que aún hoy día el utilizar software libre supone estar en una minoría, con los problemas que esto conlleva: dificultad de encontrar empresas de mantenimiento o expertos en el programa, menos disponibilidad de formación, dificultades para interoperar con los productos mayoritarios, etc. También son destacables los debidos al desconocimiento: muchos usuarios no solo no son conscientes de qué es el software libre, sino que ignoran completamente qué pueden hacer con él. Si al final no aprovechan directamente ninguna de las ventajas que les da el software libre, solo se beneficiarán muy marginalmente de ellas.

En muchos ámbitos, el software libre aún no se ha desarrollado hasta el punto de disponer de aplicaciones que sirvan para los usuarios avanzados. Por ello, antes de comprometerse a utilizar programas libres que quizás no satisfagan todas las necesidades, es necesario realizar siquiera un breve análisis del software disponible, y asegurarse de que se va a poder trabajar con comodidad y de forma productiva en el entorno deseado.



Cada usuario tendrá que decidir en qué situaciones le resulta conveniente, bien por razones éticas, bien por razones prácticas, utilizar software libre. Quizás esta breve exposición le haya ayudado a hacerse una primera idea pero, naturalmente, conviene que profundice más en sus necesidades y cómo el software libre las puede satisfacer, antes de tomar decisiones.

5. Licencias de software libre

Ya se ha mencionado cómo las libertades que proporciona un programa libre están codificadas en su licencia de redistribución. Esta es la herramienta legal que el autor utiliza para garantizarlas a los que reciben su programa.

No todas las licencias de software libre son iguales, ni tienen los mismos efectos. El autor, cuando garantiza las libertades mencionadas, puede hacerlo de forma distinta, y puede imponer también algunas condiciones (siempre que no interfieran con las libertades básicas, o la licencia no sería de software libre). Hay que tener en cuenta que la licencia es lo único que permite la redistribución de un programa libre, por lo que quien quiera realizarla no tiene más remedio que cumplirla. Por ejemplo, la licencia puede exigir que se respeten las indicaciones de autoría, o que se incluya el código fuente si se quiere redistribuir el programa listo para ejecutar.

Solo quien detente los derechos de un programa (los autores o a quienes estos hayan cedido estos derechos) pueden reclamar que se cumpla la licencia. Por este motivo los autores de software libre no renuncian a la propiedad de su creación: solo permiten a terceros que la usen, la modifiquen y la redistribuyan, pero no les ceden sus derechos.

Hay muchas licencias de software, pero por razones prácticas la mayoría de los proyectos utilizan un pequeño conjunto de cinco o seis. Por un lado muchos proyectos no quieren o no pueden dedicar recursos a diseñar una licencia propia, por otro saben que los usuarios prefieren tratar con una licencia ampliamente conocida que leerse y analizar licencias nuevas.

Para su estudio, pueden dividirse las licencias de software libre en dos grandes familias:

- Licencias permisivas: no imponen condiciones especiales en la *segunda* redistribución. Solo especifican que el software se puede redistribuir o modificar sin imponer condiciones especiales. Esto permite, por ejemplo, que el software pueda redistribuirse (con o sin modificaciones) como software privativo. Las licencias BSD modificada¹⁰, MIT¹¹ o Apache¹² son de este tipo.

- Licencias copyleft o robustas. Son las que imponen condiciones en caso de que se quiera redistribuir el software, forzando a que se sigan cumpliendo las condiciones de la licencia después de la *primera redistribución*. Las licencias GPL¹³ y LGPL¹⁴ son de este tipo.

Mientras las licencias permisivas hacen énfasis en la libertad de quien recibe un programa, ya que le permite hacer casi lo que quiera con él (en términos de condiciones de futuras redistribuciones), las robustas se centran en asegurar la libertad de cualquiera que pueda recibir un trabajo derivado del programa, ya que obliga a que las sucesivas modificaciones y redistribuciones respeten los términos de la licencia original.

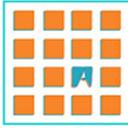
¹⁰The Modified BSD License: An Overview <http://www.oss-watch.ac.uk/resources/modbsd.xml>

¹¹MIT License (Wikipedia) http://en.wikipedia.org/wiki/MIT_License

¹²Apache Licenses <http://www.apache.org/licenses/>

¹³GNU General Public License <http://www.gnu.org/copyleft/gpl.html>

¹⁴GNU Lesser General Public License <http://www.gnu.org/licenses/lgpl.html>



La diferencia entre estos dos tipos de licencias ha sido (y es) tema de debate en la comunidad del software libre. En cualquier caso, es conveniente recordar que todas ellas son licencias libres, dado que cumplen las definiciones ya mencionadas. El lector interesado en más detalles, puede ver recopiladas y comentadas tanto licencias consideradas libres como licencias consideradas no libres desde el punto de vista de la FSF¹⁵. El punto de vista de la *Open Source Initiative* se refleja, por otro lado, en su propio listado¹⁶. En ambas listas el lector podrá comprobar cómo, en general, las licencias que ambas organizaciones reconocen son prácticamente las mismas.

El licenciamiento del software libre es un tema complejo, pero importante. Aspectos como la compatibilidad entre licencias, el relicenciamiento, el licenciamiento dual, y otros muchos tienen gran influencia en el desarrollo de programas libres, y en los modelos de negocio en uso alrededor de ellos. El lector interesado puede consultar textos más especializados para profundizar en este tema¹⁷.

6. Para concluir

El software libre es una realidad consolidada. Más que preguntarse si puede ser o no de utilidad para él, cualquier usuario debería plantearse qué programa libre puede interesarle, pues es más que posible que haya al menos alguno. Y si, siendo un profesional de la traducción, resulta que se está interesado no sólo en utilizar programas libres, sino también en contribuir a su mejora, la localización es sin duda un campo donde la colaboración será apreciada.

En general, los programas libres ampliamente extendidos se desarrollan por comunidades donde la lingua franca es el inglés. Sin embargo es muy común que los desarrolladores, habitualmente ellos mismos procedentes de contextos culturales y geográficos muy diversos, tengan mucho interés en el programa pueda ser utilizado por hablantes de otros idiomas. Por eso se han desarrollado herramientas de ayuda a la traducción y la localización que facilitan en gran medida estas tareas, desvinculándolas de forma casi completa de las tareas de desarrollo software. Muchos proyectos libres han adoptado herramientas de este tipo consiguiendo formar comunidades de traductores muy activas. De hecho, los programas libres con cierto éxito popular son hoy día, probablemente, los que están disponibles en más idiomas distintos.

Para terminar, es conveniente recordar que este artículo tiene un carácter deliberadamente introductorio. El lector interesado en profundizar en el tema puede consultar otros textos¹⁸¹⁹²⁰, donde encontrará sin duda mucho más detalle y profundidad en este mundo tan apasionante.

15Various Licenses and Comments about Them

<http://www.gnu.org/licenses/license-list.html>

16<http://www.opensource.org/licenses>

17Understanding Open Source and Free Software Licensing <http://oreilly.com/openbook/osfreesoft/book/index.html>

18Introducción al software libre <http://curso-sobre.berlios.de/introsobre/>

19Open Sources: Voices from the Open Source Revolution <http://oreilly.com/openbook/opensources/book/index.html>

20Open Sources 2.0 http://commons.oreilly.com/wiki/index.php/Open_Sources_2