

Software libre, software de código abierto, licencias.

Donde se propone un procedimiento de distribución de software y datos de investigación

T. Gomez-Diaz

CNRS, Université Paris-Est, Laboratoire d'informatique Gaspard-Monge

77454 Marne-la-Vallée Cedex 2, France

Contacto: Teresa.Gomez-Diaz@u-pem.fr

28 septiembre 2015, V.1.3-esp

Este obra se comparte bajo la licencia Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional (CC-BY-SA v4.0), <http://creativecommons.org/licenses/by-sa/4.0/>

1 Introducción

Con la entrada en funcionamiento del programa europeo H2020 [10] y la adopción, en la comunidad investigadora, de políticas de acceso libre¹, parece necesario revisar algunos conceptos básicos que inciden en la distribución de programas, tales como los de programación o software libre, software de código abierto y licencias; conceptos que deben ser tenidos en consideración antes de tomar cualquier decisión acerca de la difusión de un programa informático. Este artículo propone, asimismo, un procedimiento concreto de distribución para software y datos.

Este documento está dirigido a la comunidad investigadora, considerada en el sentido más amplio posible, y se distribuye bajo la licencia CC-BY-SA v4.0. Esto indica que puede ser modificado libremente, pero también que los autores de las modificaciones aceptan mantener la información sobre el autor inicial e indicar los cambios realizados. La redistribución de este documento o de sus modificaciones debe realizarse bajo la misma licencia con la que se emite inicialmente. Si el lector considera este documento útil, le agradecería que contribuyera a que sea accesible a un público más amplio.

2 Definición de “free software” o software libre

El concepto de software libre fue formalmente precisado por la Fundación del Software Libre (Free Software Foundation, FSF), creada en 1985 por Richard M. Stallman, pero es preciso señalar que, con anterioridad, ya existían programas cuyo protocolo de distribución correspondía a este concepto. Entre los ejemplos más conocidos mencionamos T_EX de Donald Knuth (1978) o la Berkeley Software Distribution (BSD) del Computer Systems Research Group de la Universidad de California (1977-1995).

Reproducimos aquí la definición de software libre que se encuentra en la página <http://www.gnu.org/philosophy/free-sw.es.html>²:

“Un programa es *software libre* si los usuarios tienen las cuatro libertades esenciales:

Libertad 0 La libertad de ejecutar el programa como se desee, con cualquier propósito.

Libertad 1 La libertad de **estudiar** cómo funciona el programa, y cambiarlo para que haga lo que usted quiera. El acceso al código fuente es una condición necesaria para ello.

Libertad 2 La libertad de redistribuir copias para ayudar a su prójimo.

Libertad 3 La libertad de distribuir copias de sus versiones modificadas a terceros. Esto permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.”

Un programa, código informático o software es *libre*, según esta definición, sólo si se distribuye respetando estas cuatro libertades. Declarar que un programa es *libre* no significa nada, excepto si las cuatro libertades

1. En este documento se traduce *open access* por acceso libre.

2. Más información en español se puede encontrar, por ejemplo, en [3].

mencionadas están confirmadas (legalmente) por la licencia (denominada libre) que acompañe a dicho programa. Si alguna de estas libertades no está asegurada, se dice que el programa “no es libre”, o que es “privativo”.

En nuestra opinión, aún quedan, en algunos círculos, errores de interpretación relacionados con la cuarta libertad, así que insistimos aquí sobre el hecho de que la distribución de las modificaciones de un programa libre corresponde a la decisión exclusiva del autor de dichas modificaciones. Una vez que se decide tal distribución, es necesario respetar las cláusulas de reciprocidad de la licencia inicial.

Notesé la importancia de la palabra *estudiar* en la definición anterior, que ha nacido en círculos cercanos a instituciones de investigación. La comunidad FSF afirma que en la enseñanza se debería utilizar únicamente software libre. Nuestra convicción personal es que el software libre está particularmente bien adaptado al entorno de la investigación y al modo en el que la ciencia evoluciona.

3 Definición de “open source software” o software de código abierto

Hacia el final de los años 90, la comunidad de desarrollo de programas sufrió una escisión, que concluye con la creación de la Iniciativa de Código Abierto (Open Source Initiative, OSI) en 1998 y la creación de un nuevo concepto: el software open source o de código abierto, que contiene diez condiciones. La definición completa en inglés se encuentra en la página <http://www.opensource.org/docs/osd>, y se expone aquí parcialmente.

“Código abierto no significa únicamente que se tiene acceso al código fuente. La distribución del *software de código abierto* debe cumplir estas normas:

1. Free Redistribution
The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.
2. Source Code
3. Derived Works
4. Integrity of The Author’s Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral”

Incluimos el primer punto “Free Redistribution” en su forma completa, puesto que muestra que el concepto de código abierto está más cercano a intereses comerciales que a conceptos de libertad o de estudio (del código fuente). Insistimos de nuevo en el hecho de que, para poder decir que un programa verifica esta definición, la licencia que le acompaña debe asegurar el respeto a los diez principios enunciados arriba. Si uno de ellos falta, no puede afirmarse que se trate de un programa de código abierto.

4 Diferencias, terminología, licencias

Estas dos definiciones (software libre, software de código abierto) corresponden a filosofías muy diferentes, pero en realidad, la mayor parte de los programas de los que hablaremos aquí van a ser a la vez libres y de código abierto, puesto que las licencias más utilizadas cumplen a la vez las dos definiciones. En términos legales, estas licencias son contratos que participan en la construcción del marco legal en el que se permite la utilización, la copia, la modificación, y la redistribución del software. El software libre o de código abierto no está “libre (o exento) de derechos”: los derechos de autor no van a desaparecer porque el software sea libre o de código abierto. De hecho, las licencias extienden el marco legal inicial definido por la ley (*copyright law, code de la propriété intellectuelle, derechos de autor...*).

Sin embargo, existen ejemplos de programas que son de código abierto pero no libres. Aquí mencionaremos dos ejemplos diferentes:

- Software bajo licencia NASA Open Source Agreement v1.3:

- esta licencia está en la lista de licencias OSI (<http://opensource.org/licenses>), ver <http://www.opensource.org/licenses/nasa1.3>
- pero está incluida en la lista FSF de licencias no libres (<http://www.gnu.org/licenses/license-list.es.html>), ver http://directory.fsf.org/wiki/License:NASA-OSA_v1.3

La NASA Open Source Agreement, versión 1.3, no es una licencia de software libre porque incluye una previsión que requiere que los cambios hechos sean un «diseño original». ...

- El segundo ejemplo es cada vez más común, dado que cada vez hay más programas que se utilizan en teléfonos y todo tipo de dispositivos electrónicos. Así ocurre que estos dispositivos pueden incluir versiones ejecutables de programas libres, pero se impide la realización de cambios en dichos ejecutables, vulnerando uno de los requisitos de los programas libres³.

Un poco más de terminología: un *software comercial* es un software con una licencia que requiere un pago. El software libre o de código abierto puede ser comercial, libre no quiere decir gratis⁴ y también puede darse lo contrario: gratis no corresponde siempre a libre o a código abierto. Por otra parte, un software distribuido con el código fuente no es, necesariamente, libre o de código abierto. Finalmente, el software distribuido sin licencia corresponde a una aplicación directa de la ley, es decir: “Todos los derechos reservados”.

En la comunidad de programación, el término Free/Open Source Software (FOSS) o Free/Libre/Open Source Software (FLOSS) (donde *Libre* refuerza el ángulo *libertad* de la palabra *free*) se utiliza cada vez más por programadores que prefieren no elegir entre ambas comunidades o filosofías.

5 Una clasificación de las licencias

Las licencias conceden derechos (y libertades) pero también pueden contener cláusulas que deben ser respetadas, como, por ejemplo, la de no atribuir a las versiones modificadas el mismo nombre de la versión original, o la de hacer referencia explícita a ciertas publicaciones. Incluimos aquí una clasificación extraída de “A Practical Guide to Using Free Software in the Public Sector” de T. Aimé ([6], p.9 ó [7]).

Software \ Licencia	Non Copyleft	Weak Copyleft	Strong Copyleft
Original	A	A	A
Modified	A' B	A' B	A' B

TABLA 1 – Una clasificación de las licencias.

Copyleft⁵ fuerte: la licencia del programa inicial *A* se hereda en *A'* (con o sin modificación de *A*) y en el software enlazado (*linked*) *B*. La cláusula de reciprocidad indica que la licencia inicial se hereda en la distribución del software modificado. Ejemplo: GNU GPL.

Copyleft débil: la licencia del programa inicial *A* se hereda en *A'*, pero los nuevos programas enlazados pueden tener cualquier licencia. Ejemplos: MPL, GNU LGPL.

Sin copyleft: la licencia inicial de *A* puede desaparecer en *A'* (con o sin modificación de *A*) y los nuevos programas pueden tener cualquier licencia. Ejemplos: Apache, BSD, MIT.

Dado el impacto de la cláusula de Copyleft, incluimos aquí la frase de la licencia GPLv2 que activa la herencia de la licencia: “*You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.*”

Es necesario prestar una atención particular al caso en el que licencias diferentes impongan obligaciones diversas que pueden entrar en conflicto cuando se incluyen y/o modifican diversos programas para producir un nuevo software. En la página <http://www.gnu.org/licenses/gpl-faq.html#AllCompatibility> se da una tabla de compatibilidad entre las licencias FSF.

6 El doble papel de las licencias

Por una parte, las licencias establecen un marco legal⁶ en el que se permite la utilización, la copia, la modificación y la redistribución del software. En Francia, la ley indica que la utilización o la modificación de

3. Más información en <https://www.gnu.org/philosophy/open-source-misses-the-point.es.html>

4. *Free* puede traducirse como *gratis*, lo que es muy diferente de *libre*.

5. <http://es.wikipedia.org/wiki/Copyleft>

6. Se debe evitar toda modificación de informaciones legales (autores, instituciones, fechas, licencias...) de componentes recuperados en Internet.

un software es ilegal salvo si se tiene un acuerdo (escrito), ver por ejemplo [6], p.3. Dar acceso a programas en una página web no produce ni software libre ni de código abierto, ni programas que son legalmente utilizables: es importante dar licencias. Estas licencias proporcionan un marco legal “por defecto”, y si ese marco no se ajusta al uso previsto, se puede contactar con los autores para establecer otro tipo de colaboración (quizás pagando algunos servicios).

Por otra parte, cuando ciertas licencias, como las de software libre, se deciden al nivel de instituciones, laboratorios o agencias de financiación de la investigación, cuando se requieren en documentos como la declaración de Berlín [5] o la declaración de Budapest [4], esto corresponde a políticas en las que está en juego el libre acceso a la investigación, teniendo como consecuencia, por ejemplo, el aumento de la reproductibilidad de los resultados publicados.

La palabra *open* se ha extendido rápidamente fuera del marco inicial del software, estamos en tiempos de acceso, educación, innovación o gobiernos que se califican como “open”; es una palabra de moda que se utiliza incluso en contextos que no están relacionados con el concepto inicial de “open”: es necesario, por ello, comprobar cuidadosamente las definiciones, licencias y políticas que se asocian a esta palabra.

7 Un procedimiento de distribución para software o datos de investigación

La siguiente lista puede ser adaptada para la distribución del software producido en los laboratorios de investigación [8]; igualmente, puede ser adaptada a la distribución de datos. Los items marcados con (*) se deben revisar regularmente en las versiones importantes.

- Elegir un nombre o título para identificar el objeto, evitar las marcas y otros nombres registrados, se puede asociar a dicho nombre la fecha, el número de versión...
- (*) Establecer la lista de los autores y sus afiliaciones. Asociar un porcentaje de participación, completar con los contribuidores menores. Si la lista es larga mantener la información al día en una página web.
- (*) Establecer la lista de las funcionalidades más importantes.
- (*) Establecer la lista de los programas o datos incluidos, e indicar las licencias o todo documento que permita su acceso, copia, modificación o redistribución.
- Elegir una licencia, con el acuerdo de todos los autores o titulares de los derechos de autor, obtener un acuerdo firmado si es posible. Considerar la utilización de licencias libres y licencias CC (v4.0) para los datos (por ejemplo).
- Elegir una página web, forge o depósito para distribuir el producto, con las licencias y las condiciones de uso, copia, modificación o redistribución claramente mencionadas, así como la mejor forma de citar el trabajo. Tanto unos buenos metadatos como el respeto a estándares abiertos van a ser importantes en la distribución de nuevas componentes a una comunidad numerosa: ayudará a los demás a utilizar vuestro trabajo e incrementará su longevidad. Dar licencias a la documentación (GNU FDL, CC, LAL...) y a las páginas web. Utilizar identificadores persistentes si es posible.
- (*) Archivar regularmente ficheros tar.gz para trazar funcionalidades nuevas e importantes.
- Informar a los laboratorios e instituciones de los autores (si no se hizo en el paso de licencia).
- Crear e indicar claramente una dirección de contacto.
- Distribuir el software o los datos.
- Informar a la comunidad, considerar la publicación de Software o Data Papers.

Cómo dar una licencia. Dar una licencia no es un paso difícil, pero debe ser efectuado antes de la distribución del software. La idea básica es la identificación de los ficheros con un encabezamiento comentado que incluya la información más relevante sobre los autores, las fechas, las licencias y el añadir, al conjunto de los ficheros, un nuevo fichero (llamado COPYING, LICENCE o README, por ejemplo) con el texto completo de la licencia (o un enlace al texto completo). Es necesario indicar la licencia del software en la documentación, en la página web, así como los autores y su afiliación. Aquí damos un ejemplo para el encabezamiento de los ficheros:

- nombre: nombre del fichero, nombre y version del software,
- titulares de los derechos de autor : Copyright (©), año(s), instituciones, personas...,
- autores: la lista de los autores y sus afiliaciones (o un enlace),
- licencia: nombre y versión,
- fechas: fecha de creación, fecha de última modificación.

8 Conclusión

El objetivo principal de este documento es ayudar a la comunidad de investigación al mejor entendimiento de los conceptos básicos de la distribución de software y las licencias asociadas. Si usted considera que este documento es útil, por favor contribuya a que sea accesible a un público más amplio.

Agradecimientos. Mi sincero agradecimiento a los correctores que han contribuido a la mejora de este documento en sus diversas versiones y lenguajes.

Referencias

- [1] Free Software Foundation, <http://www.fsf.org/>
- [2] Open Source Initiative, <http://opensource.org/>
- [3] Introducción al software libre, J. González Barahona, J. Seoane Pascual, G. Robles, Barcelona: Fundació per a la Universitat Oberta de Catalunya, 2003, <http://bibliotecalibre.org/bitstream/001/344/4/84-9788-028-5.pdf>
- [4] Budapest Open Access Initiative, 2002, <http://www.budapestopenaccessinitiative.org/read>
- [5] Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities, 2003, <http://openaccess.mpg.de/286432/Berlin-Declaration>
- [6] A Practical Guide to Using Free Software in the Public Sector, T. Aimé, 2010, PDF file available from <https://www.projet-plume.org/ressource/guide-logiciels-libres-administrations>
- [7] Licence & copyright pour les développements de logiciels libres de laboratoires de recherche, T. Gomez-Diaz, 2009, <https://www.projet-plume.org/ressource/faq-licence-copyright>
- [8] Diffuser un logiciel de laboratoire : recommandations juridiques et administratives, T. Gomez-Diaz, 2010, <https://www.projet-plume.org/ressource/diffuser-logiciel-recomm-juridiques-admin>
- [9] Thème PLUME: “Patrimoine logiciel d’un laboratoire”, <http://www.projet-plume.org/patrimoine-logiciel-laboratoire>
- [10] Towards better access to scientific information: Boosting the benefits of public investments in research, European Commission, 2012, http://ec.europa.eu/research/science-society/document_library/pdf_06/era-communication-towards-better-access-to-scientific-information_en.pdf
Background page: <http://ec.europa.eu/research/science-society/index.cfm?fuseaction=public.topic&id=1301>