
The **El 9º HACKERS MEETING** tendrá lugar en Rennes (Francia), del 18 al 20 de agosto.

Esta es una traducción de la página original en inglés.

Por qué el «código abierto» pierde de vista lo esencial del software libre

por **Richard Stallman**

Cuando decimos que el software es «libre», nos referimos a que respeta las libertades esenciales del usuario: la libertad de utilizarlo, ejecutarlo, estudiarlo y modificarlo, y de distribuir copias con o sin modificaciones. Es una cuestión de libertad y no de precio, por lo tanto piense en «libertad de expresión» y no en «barra libre». ^[1]

Estas libertades son de vital importancia. Son esenciales no solamente para el bien del usuario individual sino para la sociedad entera, porque promueven la solidaridad social: compartir y cooperar. La importancia de estas libertades aumenta a medida que nuestra cultura y nuestras actividades cotidianas se vinculan cada vez más con el mundo digital. En un mundo de sonidos, imágenes y palabras digitales, el software libre se vuelve cada vez más esencial para la libertad en general.

Decenas de millones de personas alrededor del mundo ahora utilizan software libre; las escuelas públicas de algunas regiones de India y España enseñan a todos los estudiantes a utilizar el sistema operativo libre GNU/Linux. Sin embargo, la mayoría de estos usuarios nunca han oído las razones éticas por las cuales desarrollamos este sistema y construimos la comunidad del software libre, porque este sistema y esta comunidad son descritos como «de código abierto» y atribuidos a una filosofía diferente que rara vez menciona estas libertades.

El movimiento del software libre ha hecho campaña por la libertad de los usuarios de ordenador desde 1983. En 1984 iniciamos el desarrollo del sistema operativo libre GNU, para poder evitar el uso de sistemas operativos que no son libres y que niegan la libertad a los usuarios. Durante los años ochenta desarrollamos la mayor parte de los componentes esenciales del sistema GNU, y diseñamos la Licencia Pública General de GNU (GNU GPL, por sus siglas en inglés) para usarla en la distribución de dichos componentes; una licencia diseñada específicamente para proteger la libertad de todos los usuarios de un programa.

Sin embargo, no todos los usuarios y programadores de software libre estaban de acuerdo con los objetivos del movimiento del software libre. En 1998 una parte de la comunidad del software libre se bifurcó y dió inicio a una campaña para promover el «*open source*» (código abierto). La expresión se propuso originalmente para evitar un posible malentendido con el término «*free software*» ^[2] (software libre), pero pronto se asoció con posiciones filosóficas diferentes a las del movimiento del software libre.

Algunos de los defensores del «código abierto» lo consideraron una «campaña de marketing para el software libre», con el objetivo de atraer a los ejecutivos de las empresas enfatizando los beneficios prácticos sin mencionar conceptos de lo que es correcto e incorrecto —y que quizá los empresarios no deseaban oír—. Otros defensores rechazaban rotundamente los valores éticos y sociales del software libre. Cualesquiera que hayan sido sus puntos de vista, cuando hacían campaña por el «código abierto» no mencionaban ni abogaban por esos valores. La expresión «código abierto» fue rápidamente asociada con ideas y argumentaciones basadas únicamente en valores de orden práctico, tales como desarrollar o usar software potente y confiable. La mayoría de los partidarios del «código abierto» llegaron al movimiento después de entonces y hacen la misma asociación de conceptos.

Ambas expresiones describen casi la misma categoría de software, pero representan puntos de vista basados en valores fundamentalmente diferentes. El código abierto es una metodología de programación, el software libre es un movimiento social. Para el movimiento del software libre, el software libre es un imperativo ético, respeto esencial por la libertad de los usuarios. En cambio la filosofía del código abierto plantea las cuestiones en términos de cómo «mejorar» el software, en sentido meramente práctico. Sostiene que el software privativo no es una solución óptima para los problemas prácticos que hay que resolver. En la mayoría de los casos, cuando se discute sobre «código abierto» no se toma en consideración el bien y el mal sino únicamente la popularidad y el éxito; he aquí un ejemplo típico.

Para el movimiento del software libre, sin embargo, el software que no es libre es un problema social y la solución consiste en dejar de usarlo, migrar al software libre.

«Software libre». «Código abierto». Si es el mismo software (o casi), ¿importa acaso qué nombre se utiliza? Sí, porque las diferentes palabras expresan ideas diferentes. Aunque un programa libre con cualquier otro nombre le dará hoy la misma libertad, establecer la libertad de manera perdurable depende sobre todo de enseñar a las personas a valorar la libertad. Si desea ayudar en esto, es esencial que use la expresión «software libre».

Nosotros, en el movimiento del software libre, no vemos el ámbito del código abierto como al enemigo; el enemigo es el software privativo, el que no es libre. Pero queremos que la gente sepa que defendemos la libertad, así que no aceptamos que se nos identifique como partidarios del código abierto.

Diferencias prácticas entre software libre y código abierto

En la práctica, el código abierto sostiene criterios menos estrictos que los del software libre. Por lo que sabemos, todo el código fuente de software libre existente que se ha publicado se podría calificar como código abierto. Casi todo el software de código abierto es software libre, con algunas excepciones. En primer lugar, algunas licencias de código abierto son demasiado restrictivas, por lo que no se las puede considerar como libres. Por ejemplo, «Open Watcom» no es libre porque su licencia no permite hacer versiones modificadas y utilizarlas de forma privada. Afortunadamente, son muy pocos los programas que llevan tales licencias.

En segundo lugar, y lo que en la práctica es más importante, muchos productos que funcionan como ordenadores verifican las firmas de sus programas ejecutables para impedir que los usuarios instalen ejecutables diferentes; solo una compañía tiene el privilegio de elaborar ejecutables que funcionen en el dispositivo y de acceder a todas las prestaciones del mismo. A estos dispositivos los llamamos «tiranos» y la práctica se denomina «tivoización», por referencia al producto (Tivo) donde por primera vez descubrimos su implementación. Aun cuando el ejecutable esté hecho a partir de código fuente libre, los usuarios no pueden ejecutar versiones modificadas, de modo que el ejecutable no es libre.

Los criterios del código abierto no contemplan esta cuestión, solo les interesa la licencia del código fuente. De modo que estos ejecutables no modificables, si están hechos a partir de un código fuente como Linux, que es de código abierto y libre, son de código abierto pero no son libres. Muchos productos Android contienen ejecutables tivoizados de Linux, que no son libres.

Errores frecuentes sobre el significado de «software libre» y «código abierto»

La expresión «software libre» puede dar lugar a un malentendido. El significado no intencional —«software que se puede obtener sin costo alguno»— corresponde, y también corresponde el significado con el que lo usamos: «software que da al usuario ciertas libertades». Resolvemos este problema publicando la definición de software libre, donde decimos: «Piense en libertad de expresión, no en barra libre». No es una solución perfecta, no puede eliminar completamente el problema. Un término correcto e inequívoco sería mejor, si no presentase otros problemas.

Lamentablemente, todas las alternativas en inglés presentan algún problema. Hemos analizado muchas alternativas que nos han sugerido, pero ninguna es tan claramente «correcta» como para adoptarla. Por ejemplo, en ciertos contextos se puede usar la palabra española y francesa «libre», pero en India no la reconocerán en absoluto. Todas las alternativas propuestas para «software libre» tienen algún tipo de problema semántico, incluso «software de código abierto».

La definición oficial de «software de código abierto» (publicada por la *Open Source Initiative* y demasiado larga como para citarla aquí) se derivó indirectamente de nuestros criterios para el software libre. No es la misma, es un poco más amplia en algunos aspectos. No obstante, dicha definición concuerda con la nuestra en la mayoría de los casos.

Sin embargo, el significado obvio de la expresión «software de código abierto» es «puede mirar el código fuente», y pareciera que muchos opinan que eso es lo que significa. Ese es un criterio mucho más débil que la definición de software libre, y también mucho más débil que la definición oficial de código abierto. Incluye muchos programas que no son ni libres ni de código abierto.

Debido a que el significado obvio de la expresión «código abierto» no es el que sus defensores quieren darle, la mayoría la interpreta erróneamente. Según el escritor Neal Stephenson, «Linux es software de 'código abierto',

lo que significa simplemente que cualquiera puede obtener copias de los archivos del código fuente». No pienso que su intención haya sido rechazar o cuestionar deliberadamente la definición oficial. Pienso que simplemente aplicó las convenciones del idioma inglés para obtener el significado de la expresión. El estado de Kansas publicó una definición similar: «Utilicen software de código abierto (OSS). OSS es el software cuyo código fuente está disponible pública y libremente, aunque los términos específicos de licenciamiento pueden variar con respecto a lo que se permite hacer con el código».

El *New York Times* publicó un artículo que amplía el significado del término para referirse a las pruebas de los programas beta por parte de los usuarios (se permite a unos cuantos usuarios probar una versión inicial para que den sus impresiones de forma confidencial), algo que los programadores de software privativo han hecho durante décadas.

El término se ha extendido para incluir los planos de máquinas que se publican sin patentes. Los planos de máquinas libres de patentes son una loable contribución a la sociedad, pero la expresión «código fuente» no se aplica en este caso.

Los partidarios del código abierto intentan afrontar este problema refiriéndose a su definición oficial, pero ese enfoque correctivo es menos efectivo para ellos que para nosotros. El término «software libre» tiene dos significados naturales, uno de los cuales es el que le damos, de manera que una persona que ha comprendido la idea de «libertad de expresión, no barra libre» no se equivocará de nuevo. Al contrario, el «código abierto» tiene solamente un significado natural, el cual es diferente del que sus partidarios desean darle. Así, no hay una manera concisa de explicar y justificar la definición oficial de «código abierto», lo que causa aún más confusión.

Otra mala interpretación de «código abierto» es la idea de que significa «no usar la GPL de GNU». Esto tiende a provocar otro malentendido: «software libre» equivale a «software que está bajo la GPL de GNU». Ambas interpretaciones son incorrectas, ya que la GPL de GNU se califica como licencia de código abierto, y la mayoría de las licencias de código abierto se consideran licencias de software libre. Existen muchas licencias de software libre además de la GPL de GNU.

El término «código abierto» se ha extendido aún más debido a su aplicación en otras actividades tales como el gobierno, la educación y la ciencia, todos campos en los que no existe nada parecido al código fuente, y donde los criterios aplicables a las licencias de software no son pertinentes. El único elemento que estas actividades tienen en común es que, de algún modo, se invita a las personas a participar. Fuerzan tanto el término que llega a significar únicamente «participación» o «transparencia», o aún menos que eso. En el peor de los casos, se ha convertido en una trivial expresión de moda.

Valores diferentes pueden llevar a conclusiones similares, pero no siempre

Los grupos radicales de los años sesenta tenían la reputación de estar divididos en facciones: algunas organizaciones se apartaban debido a desacuerdos sobre detalles de estrategia, y los dos grupos resultantes se consideraban enemigos aunque tuvieran metas y valores básicos similares. El ala derecha de la política se aprovechó de esto y lo utilizó para criticar a la izquierda en general.

Algunos intentan desacreditar el movimiento de software libre poniendo nuestro desacuerdo con el código abierto en el mismo plano que los desacuerdos entre aquellos grupos radicales. Lo entienden al revés. Estamos en desacuerdo con el código abierto en lo que respecta a los objetivos y valores básicos, pero su perspectiva y la nuestra conducen en muchos casos al mismo comportamiento práctico, por ejemplo, programar software libre.

Como resultado, personas del movimiento del software libre y del ámbito del código abierto a menudo trabajan conjuntamente en proyectos prácticos tales como el desarrollo de software. Es notable que posiciones filosóficas tan diferentes puedan tan a menudo motivar a diferentes personas a participar en los mismos proyectos. Sin embargo, hay situaciones en las que estos puntos de vista fundamentalmente distintos dan como resultado acciones totalmente diferentes.

La idea del código abierto es que permitiendo que los usuarios modifiquen y redistribuyan el software se obtienen programas más potentes y confiables. Pero no hay ninguna garantía de que esto sea así. Los programadores de software privativo no son necesariamente incompetentes. A veces producen algún programa potente y confiable, aunque no respete la libertad de los usuarios. La reacción de los activistas del software libre y de los entusiastas del código abierto frente a esa situación será muy diferente.

Un entusiasta puro del código abierto, uno que no esté influenciado para nada por los ideales del software libre,

dirá: «Estoy sorprendido de que haya logrado que su programa funcione tan bien sin haber utilizado nuestro modelo de desarrollo, pero lo logró. ¿Cómo puedo obtener una copia?» Esta actitud premia los esquemas que nos quitan la libertad, llevándonos a perderla.

El activista del software libre dirá: «Su programa es muy atractivo pero valoro más mi libertad. Así que rechazo su programa. Haré mi trabajo de alguna otra manera y apoyaré un proyecto para el desarrollo de un reemplazo libre». Si valoramos nuestra libertad, podemos actuar para mantenerla y defenderla.

El software potente y confiable puede ser malo.

El deseo de que el software sea potente y confiable deriva de suponer que tiene que estar diseñado para que resulte útil a sus usuarios. Si es potente y confiable, el software será de mayor utilidad.

Pero sólo se puede decir que el software es útil si respeta la libertad de los usuarios. ¿Qué pasa si el software está diseñado para encadenar a los usuarios? En ese caso la potencia hace que las cadenas sean más restrictivas, y la confiabilidad significa que son más difíciles de quitar. Las funcionalidades maliciosas, como espiar a los usuarios, restringir a los usuarios, las puertas traseras y las actualizaciones impuestas, son comunes en el software privativo, y algunos defensores del código abierto se proponen hacer lo mismo en programas de código abierto.

Bajo la presión de las compañías discográficas y cinematográficas, el software que se pone a disposición de los usuarios está diseñado cada vez más específicamente para restringirlos. Esta funcionalidad maliciosa se conoce como «gestión digital de restricciones» DRM (véase DefectiveByDesign.org) y es la antítesis, en espíritu, de la libertad que el software libre busca proveer. Y no sólo en espíritu, puesto que el objetivo del DRM es pisotear su libertad: los programadores de DRM intentan dificultarle, hacer que le resulte imposible o incluso ilegal modificar los programas que implementan el DRM.

Con todo, algunos partidarios del código abierto han propuesto software con «DRM de código abierto». La idea es que publicando el código fuente de los programas diseñados para restringir su acceso a los datos cifrados, y permitiendo que otros lo modifiquen, se obtendrá software más potente y confiable para restringir a los usuarios como usted. Luego el software se le entregará a usted en dispositivos que no le permitirán modificarlo.

Aunque este software sea de código abierto y utilice el modelo de desarrollo del código abierto, no será software libre ya que no respetará la libertad de los usuarios que en la práctica lo ejecutan. Si el modelo de desarrollo del código abierto logra que este software sea más poderoso y confiable para restringirle a usted como usuario, eso lo hará aún peor.

Miedo de la libertad

Al inicio, la principal motivación de los que decidieron separar el código abierto del ámbito del software libre fue que los planteamientos éticos del «software libre» incomodaban a muchas personas. Es cierto: plantear cuestiones éticas como el tema de la libertad, hablar de responsabilidades y de conveniencia, es inducir a las personas a que se cuestionen cosas que quizá prefieran ignorar, por ejemplo preguntarse si su conducta es ética. Esto puede generar malestar y algunos pueden optar simplemente por ignorar estas cuestiones. Pero esto no quiere decir que tengamos que dejar de hablar de ello.

Sin embargo, eso es lo que decidieron hacer los líderes del «código abierto». Pensaron que omitiendo hablar de ética y de libertad, mencionando únicamente los beneficios prácticos inmediatos de cierto tipo de software libre, podrían «vender» el software más fácilmente a ciertos usuarios, especialmente a las empresas.

Cuando los defensores del código abierto mencionan algo más profundo que esto, normalmente es la idea de hacer un «regalo» de código fuente a la humanidad. Presentar esto como un hecho especialmente bueno, como algo que va más allá de lo moralmente exigible, presupone que distribuir software privativo sin código fuente es moralmente legítimo.

Tal enfoque se ha demostrado eficiente, en sus propios términos. La retórica del código abierto ha convencido a muchas empresas y particulares a usar e incluso desarrollar software libre, lo cual ha extendido nuestra comunidad, pero solamente a un nivel práctico y superficial. La filosofía del código abierto, con sus valores puramente prácticos, impide la comprensión de las ideas más profundas del software libre. Trae muchas personas a nuestra comunidad, pero no les enseña cómo defenderla. Eso es bueno, hasta cierto punto, pero no asegura la libertad. Atraer usuarios al software libre los lleva sólo hasta una parte del camino que hay que recorrer para convertirse en defensores de su propia libertad.

Tarde o temprano estos usuarios se sentirán tentados a volver al software privativo por alguna ventaja práctica. Son innumerables las compañías que buscan ofrecer esa tentación, algunas hasta ofrecen copias gratuitas. ¿Qué motivaría a los usuarios a rechazar esto? Sólo si han aprendido a valorar la libertad que el software libre les brinda, a valorar la libertad como tal en vez de la conveniencia técnica y práctica de algún software libre en particular. Para diseminar esta idea, tenemos que hablar acerca de la libertad. Cierta dosis de «silencio» en el trato con las empresas puede resultar útil para la comunidad, pero es peligroso cuando se vuelve tan común que el amor a la libertad llega a verse como una excentricidad.

Esa peligrosa situación es exactamente la que tenemos. Mucha gente relacionada con el software libre, especialmente los que lo distribuyen, habla poco acerca de la libertad; normalmente porque buscan ser «más atractivos para las empresas». Casi todas las distribuciones del sistema operativo GNU/Linux añaden paquetes privativos al sistema libre de base, y con ello invitan a los usuarios a considerar esto como una ventaja en lugar de un defecto.

Las extensiones de software privativas y las distribuciones GNU/Linux que son parcialmente libres encuentran terrenos fértiles, porque gran parte de nuestra comunidad no insiste en la libertad del software. Esto no es una coincidencia. La mayor parte de los usuarios de GNU/Linux llegaron al sistema por el discurso del «código abierto», el cual no menciona la libertad como una meta. Las prácticas que no sostienen la libertad y las palabras que no hablan de libertad van de la mano, promoviéndose entre sí. Para superar esta tendencia tenemos que hablar más de libertad, y no menos.

«FLOSS» y «FOSS»

Los términos «FLOSS» y «FOSS» se utilizan para indicar neutralidad entre el software libre y el código abierto. Si su objetivo es ser neutral, el más adecuado es «FLOSS», ya que este es verdaderamente neutral. Pero si lo que usted desea es defender la libertad, utilizar un término neutral no es la manera de hacerlo. Defender la libertad implica mostrar a los demás que se está a favor de la libertad.

Rivalidad de ideas

Los términos «libre» y «abierto» rivalizan por las ideas. Las expresiones «software libre» y «código abierto» reflejan ideas diferentes, pero según el punto de vista de la mayoría de las personas en el campo del software, ambas compiten por el mismo espacio conceptual. La costumbre de decir y pensar en términos de «código abierto» constituye un obstáculo para la comprensión y consideración de la filosofía del movimiento del software libre. Una vez que las personas se hayan acostumbrado a relacionar nuestro movimiento y nuestro software con la palabra «abierto», probablemente nos veremos en la necesidad de tener que provocarles un colapso intelectual para que se den cuenta de que lo que nosotros propugnamos es *otra cosa*. Toda actividad que promueve el término «abierto» tiende a ensanchar el telón que oculta las ideas del movimiento del software libre.

Los activistas del software libre, por lo tanto, harán bien en negarse a participar en actividades que se denominan «abiertas». Aunque la iniciativa en sí misma sea positiva, toda contribución a la misma causará un poco de daño colateral, al promover la idea del código abierto. Existen muchísimas actividades provechosas que se denominan «libres», y toda contribución a estos proyectos aportará un poco de beneficio colateral. Con tantos proyectos útiles a disposición, ¿por qué no escoger alguno que aporte mayor beneficio?

Conclusión

A medida que los promotores del código abierto atraen nuevos usuarios a nuestra comunidad, nosotros, los activistas del software libre, tenemos que trabajar aún más para llevar el concepto de libertad a estos nuevos usuarios. Tenemos que decir «¡Es software libre y te brinda libertad!» más fuerte que nunca. Cada vez que usted dice «software libre» en lugar de «código abierto», apoya nuestros esfuerzos.

Notas

Lakhani y Wolf escribieron una ponencia acerca la motivación de los programadores de software libre donde sostienen que una parte considerable está motivada por la opinión de que el software debe ser libre (a pesar de que encuestaron a los desarrolladores de *SourceForge*, un sitio que no opina que este sea un asunto ético).

Notas de traducción

[1]. En inglés el término *free* puede significar «libre» o «gratis».

[2]. En inglés la expresión «*free software*» podría interpretarse como «software gratis», pero esta ambigüedad no existe en español.

Copyright © 2007, 2010, 2012, 2015, 2016 Richard Stallman

Esta página está bajo licencia Creative Commons Reconocimiento-SinObraDerivada 4.0 Internacional.