

## Las ventajas de la apertura

Por Jesús M. González-Barahona

**Resumen:** Una de las principales consecuencias de publicar un programa como software libre es que su código fuente queda a disposición de quien quiera examinarlo. Además, mucha otra información sobre el proceso de desarrollo suele estar también fácilmente accesible en internet. En este artículo se presentan algunas de las consecuencias de esta “apertura” del proceso de construcción de software libre, y se explora brevemente cómo estos mismos mecanismos de apertura podrían comportarse en otros campos.

**Palabras clave:** Software libre, Software abierto, Código abierto

**Title:** The advantages of openness

**Abstract:** One of the main consequences of publishing a program as libre (free, open source) software is that its source code is made available to anyone who wants to look at it. In addition, much other information on the development process is often easily accessible on the internet. Some of the consequences of “opening up” the process of building libre software are discussed, briefly exploring how these open mechanisms could behave in other areas.

**Keywords:** Free software, Open source software, Open software

**González-Barahona, Jesús M.** “Las ventajas de la apertura”. En: *El profesional de la información*, 2008, enero-febrero, v. 17, n. 1, pp. 5-7.

DOI: 10.3145/epi.2008.ene.01



**Jesús M. González Barahona** es profesor en la Universidad Rey Juan Carlos (URJC) de Madrid. Comenzó a trabajar en la promoción del software libre en 1991 y colabora con varios proyectos de software libre (entre ellos Debian), con asociaciones como HispaLinux y EuroLinux. Asesora a empresas y administraciones públicas en sus estrategias relacionadas con este tema. Imparte asignaturas de doctorado sobre software libre en programas de la Univ. Politécnica de Madrid y la URJC y es miembro del comité científico del máster en software libre de la Univ. Oberta de Catalunya y codirector del master en software libre de la URJC, impartido en colaboración con Igalia y Caixa Nova. Ha participado y participa en proyectos internacionales: Calibre, FlossWorld, Flossdel software libre en 1991 y colabora con varios proyectos: Calibre, FlossWorld, FlossMetrics, Qualoss, Qualipso, FlossImpact.

### 1. Software libre, código abierto

El software libre lo es porque se puede estudiar, modificar, y también redistribuir como versión modificada<sup>1</sup>. Por lo tanto, el que un programa sea libre o no viene marcado por la licencia bajo la cual ha sido publicado, que es la que otorga (o no) estos permisos a quien reciba una copia.

Una de las principales consecuencias de esta definición es que el código fuente (que viene a corresponder, en líneas generales, con los “planos” detallados que indican cómo está construido el programa) tiene que estar disponible. En otras palabras, cualquier experto puede entender, con todo el detalle, cómo está construido el programa, hasta el punto de que podrá modificarlo sabiendo lo que hace. Naturalmente, este experto podrá también reconstruir completamente el programa a partir de ese código fuente, si así lo desea.

A primera vista, esta característica del software libre parece de interés sólo para los expertos que pueden

entender el código fuente, y saben cómo modificarlo, por ejemplo para corregir un error. Sin embargo es uno de los aspectos del software libre con más impacto en los usuarios finales. Veamos por qué...

### 2. El círculo virtuoso de la innovación distribuida

Cuando un usuario de software libre percibe un problema con un programa (quizás un funcionamiento erróneo, pero también una funcionalidad que echa de menos, por ejemplo), puede hacer algo más que quedarse esperando a ver si la situación mejora. Puede enviar un informe al grupo de desarrollo del programa. Pero también puede contratar a alguien que le arregle su problema. O puede arreglarlo él mismo si sabe trabajar con el código fuente del programa.

El que el código fuente esté disponible es justamente lo que permite que pueda elegir cualquiera de las

dos últimas opciones. O sea: la mejora del programa no depende sólo del grupo que hace el desarrollo, sino que cualquiera que sepa puede mejorar el programa, o si no tiene conocimiento para ello pero tiene recursos puede encargar que terceras partes realicen la mejora.

Y si algo de esto ocurre, quien realiza la mejora está además muy motivado para contribuir con ella de vuelta al grupo de desarrollo del programa, porque así es muy posible que tal mejora sea incluida en nuevas versiones, sin tener que volver a realizar el esfuerzo correspondiente. Y no hay que olvidar que estas nuevas versiones quedan a disposición de todo el mundo, por lo que hay un potente sentimiento de estar ayudando a un esfuerzo colectivo, incluso si el producto lo elabora una empresa o un grupo relativamente cerrado.

Así, gracias a que el código fuente está disponible, cualquiera puede innovar sobre el programa. Si el grupo de gente interesado en él es suficientemente grande, este mecanismo asegura un flujo constante y muy activo de mejoras e innovaciones. Y cualquier usuario del programa se beneficia de ellas (incluso si no sabe realizar ninguna de ellas).

Naturalmente, si el código fuente no estuviera disponible, nada de esto sería posible, y tendríamos el modelo de innovación centralizada clásico en la industria del software privativo: sólo quien produce un programa puede mejorarlo.

### 3. No sólo el código

El mecanismo descrito anteriormente se ve muy potenciado si no sólo el código, sino toda la información relacionada con el desarrollo del programa, está disponible para terceros. Los debates de diseño, los informes de error, la historia de modificaciones del programa, y otras fuentes de información permiten que cualquiera que se aproxime al proyecto pueda tener con relativa rapidez una visión muy detallada y completa de lo que está ocurriendo en él. Esto no sólo permite hacer evaluaciones detalladas (y muy exactas) de la “salud” de la comunidad que está manteniendo un programa, sino que también permite obtener fácilmente una gran cantidad de conocimiento sobre cómo se desarrolla.

Estos efectos son bien conocidos en la comunidad del software libre, y por ello la idea de “apertura” normalmente se extiende no sólo al código, sino a muchísimos otros aspectos de la producción del programa. Es importante darse cuenta de que, aunque el caso del código venía obligado por la licencia (si el software es libre, el código fuente ha de estar disponible), el publicar o no más información es algo que los proyectos pueden hacer o no, sin por eso dejar de ser software libre.

Los proyectos de software libre suelen tomar esta actitud abierta hacia la información hasta extremos poco concebibles en otras disciplinas, porque consideran que las ventajas de hacerlo son muchas. Son conscientes de que eso está ayudando a que terceros tengan más información sobre las características, y eso potencia el que se animen a realizar innovaciones, y a que esas innovaciones sean útiles, y en línea con las necesidades y requisitos del proyecto.

Cuanto más abierto es el proyecto, más fácil es captar recursos externos para tareas de mejora e innovación, así como atraer personas potencialmente interesadas que pueden acabar incorporándose al grupo de desarrollo, y atraer talento externo que estudie el programa, destaque fallos y problemas, y quizá incluso proponga soluciones.

Así, la cultura de la mayor parte de los proyectos de software libre incluye esta máxima apertura: “publiquemos toda la información que podamos, porque eso nos beneficia”.

### 4. Aprender de lo que ya se sabe

Pero esto no es tan extraño. Los beneficios del escrutinio público son bien conocidos en otros dominios. Uno de los que primero viene a la mente es el ámbito científico. Se espera que los investigadores publiquen todos los detalles posibles de cualquier avance que hayan logrado, con la idea de que otros lo puedan reproducir, criticar, y si es posible, lo mejoren. Casi exactamente con los mismos efectos que en el mundo del software libre: mejora de la calidad, innovación distribuida, detección temprana de errores y problemas, etc.

Sin embargo, en uno de los dominios donde podría esperarse más aplicación, se encuentra todavía muy poca. Entre los productores de obras culturales libres (música, documentos, películas) es todavía muy poco habitual publicar información que no sea la propia obra liberada.

Algunos grandes proyectos, como la *Wikipedia*, sí se han preocupado (y se están preocupando cada vez más) de que toda la información relacionada con el proceso de creación colectiva quede documentado públicamente. Las páginas de discusión o las listas de correo públicas, por ejemplo, son grandes pasos en este sentido.

Pero la *Wikipedia* es hoy día más una excepción que una norma. Para muchos documentos libres que circulan por la Red es hasta difícil encontrar una versión editable (un formato que permita cambiarlo de forma simple). Es casi imposible conseguir los materiales separados (distintas pistas, por ejemplo) con los que

Register for free at <https://www.scipedia.com> to download the version without the watermark

se ha editado una obra musical. Y no hablemos de los materiales fuente (planos, bandas de sonido, imágenes estáticas, etc.) para una película.

Hasta que todo esto esté disponible será difícil que la innovación por terceros ocurra sobre esas obras. Y si además, el proceso de desarrollo no está muy documentado públicamente (por ejemplo, las discusiones del equipo de creadores, las pruebas que dieron lugar a una cierta versión de una banda sonora, los borradores de los guiones de una película), más difícil será atraer talento externo que colabore en la mejora (o simplemente, en la evolución) de la obra.

En el caso de obras relativamente utilitaristas, como es el caso de la *Wikipedia*, quizá esto se está reconociendo antes, y por eso el camino está más avanzado. En las obras más puramente artísticas queda un largo camino por andar.

En estos aspectos, como en muchos otros, el software libre tiene ya una historia que merece repasar, para aprender ella. Y aprovecharse de lo aprendido allí...

### Notas

1. Esta definición rápida y poco rigurosa es una simplificación de la de “free software” de la *Free Software Foundation*, y de la de “open source software” de la *Open Source Initiative*, que son mucho más detalladas, y en términos generales definen el mismo concepto al que en este artículo me referiré como “software libre”.

2. Copyright 2008 Jesús M. González Barahona. Algunos derechos reservados. Este artículo se distribuye bajo la licencia “Reconocimiento-CompartirIgual 2.5 España” de *Creative Commons*, disponible en: <http://creativecommons.org/licenses/by-sa/2.5/es/deed.es> Este artículo y otros del mismo autor está disponible en: <http://sinetgy.org/jgb>

**Jesús M. González-Barahona**  
[jesus.gonzalez.barahona@urjc.es](mailto:jesus.gonzalez.barahona@urjc.es)



**DocuMenea** es un sistema de noticias sobre Biblioteconomía, Documentación, Archivística, Tratamiento de la Información, Periodismo, Internet y Nuevas tecnologías basado en el software de Menéame.

No estés ni un día más sin leer las novedades y votar las que consideres importantes para hacerlas más visibles:

<http://www.documenea.com>

# SCIPEDIA

Register for free at <https://www.scipedia.com> to download the version without the watermark



The screenshot shows the Scipedia website interface. At the top, there's a navigation bar with 'DocuMenea' logo and search options. Below is a section for 'últimas noticias' (latest news) with several articles listed, each with a date, title, and a brief description. On the right side, there are interactive buttons for 'Vota las últimas noticias recibidas' and 'enviar una historia'. At the bottom right, there's a 'figsona' section and a 'categorias' (categories) list including topics like 'Archivos', 'Bibliotecas', 'Educación y formación', etc.