



# SOBRE EL 'CÓDIGO ABIERTO' (OPEN SOURCE)

## Tabla de contenidos

¿Qué es el 'Software de Código Abierto'?

Sí, pero ¿qué ventajas aporta el código abierto a un cliente?

¿Cuál es el modelo de desarrollo del código abierto?

¿Cuál es el modelo de financiación del código abierto?

Principales hitos en la historia del 'Software de Código Abierto'

Otras dudas...

Sobre este documento



Versión 1.2 - Copyright (c) 2003-2006 Luis Miguel Armendáriz (<http://servi3.com>)

Está permitido copiar, distribuir y/o modificar este documento bajo los términos de la GNU Free Documentation License, Version 1.2

Para obtener una copia de la licencia "GNU Free Documentation License" visite

<http://www.fsf.org/licenses/fdl.txt>.

Todas las marcas pertenecen a sus respectivos propietarios.

## ¿Qué es el 'Software de Código Abierto'?

Lo primero que hay que tener en cuenta es que cuando 'compramos un programa' lo que realmente estamos haciendo es pagar por una licencia de uso del programa. El programa sigue perteneciendo al desarrollador<sup>1</sup>. Por ejemplo, pese a los millones de licencias de uso vendidas, Ms-Windows pertenece a Microsoft.

El Software de Código Abierto, también llamado simplemente OS (por 'Open Source'), se diferencia del software de código cerrado en su licencia. La licencia de código abierto permite explícitamente:

- utilizar el programa para cualquier propósito y sin limitaciones.
- estudiar como funciona el programa.
- redistribuir copias del programa (no se paga por la licencia).
- modificar el programa.

En oposición a esto, las distintas licencias de software cerrado explícitamente:

- limitan el uso del programa<sup>2</sup>.
- prohíben intentar conseguir el código<sup>3</sup>.
- prohíben realizar (y distribuir) copias del programa.
- prohíben modificar el programa<sup>4</sup>.

Para garantizar los derechos de los usuarios de software de código abierto, toda copia del software DEBE distribuirse con la nota de copyright, una copia de la licencia y el código fuente.

El copyright indica el propietario del software, para que no se lo pueda apropiar un tercero.

La licencia garantiza su condición de OS.

El código fuente es la base que permite estudiar y modificar el software.

## Sí, pero ¿qué ventajas aporta el código abierto a un cliente?

### Ventajas económicas:

- El usuario no paga por la licencia de uso del programa.
- El proveedor cobra únicamente por los servicios que presta.
- Normalmente el software asociado a una aplicación OS es también OS. Con el software cerrado suele ser necesario comprar licencias de herramientas que complementen la que ya tenemos.

### Independencia del proveedor y mejora de servicios:

- El cliente es independiente del proveedor ya que, al disponer del código fuente, cualquier proveedor puede proseguir donde terminó el anterior.
- Puesto que el proveedor solo cobra por sus servicios y el cliente no tiene ninguna atadura, el proveedor concentra sus esfuerzos en dar un buen servicio al cliente -el único modo de mantenerlo-.

### Disponibilidad de los datos e Integración de los procesos:

- Estando todo el código disponible, cualquier nuevo desarrollo puede utilizar los datos y procesos del cliente, integrando los distintos programas.
- Los datos generados siempre serán accesibles, sin obligar para ello al cliente a invertir en licencias<sup>5</sup>.
- Al conocerse el funcionamiento de los programas, podrán operar entre ellos sin restricciones.

### Transparencia y seguridad

- Al poderse estudiar el código, las empresas pueden estar seguras respecto al uso que se hace de sus datos<sup>6</sup> y los procesos que se utilizan. No en vano algunos programas han resultado tener puertas traseras o realizar envíos de información sin conocimiento del usuario<sup>7</sup>.

1 En casos particulares de desarrollos a medida puede no ser así

2 P.e.: solo se pueden usar en una CPU, solo hasta 2 procesadores, solo 5 usuarios, solo en algunos países...

3 Ingeniería inversa

4 P.e.: La mayoría de licencias de Microsoft prohíbe la traducción

5 P.e.: Un documento en formato cerrado .doc a veces no puede abrirse en versiones anteriores de MsWord. Datos en formato .mdb solo pueden abrirse correctamente con MsAccess (y también dependiendo de la versión)

6 P.e.: La cámara de gobierno británica dejó de usar el formato .doc en septiembre de 2003 al comprobar que éstos guardaban información sobre autores, revisiones, etc.

7 Algunos sistemas -como Windows2000- envían información del cliente al fabricante. El problema es que al no conocerse el código, no puede saberse exactamente qué información envía. Otros parecen tener puertas traseras

- Los errores del software pueden ser detectados y corregidos por el usuario. Evidentemente, también se puede pagar a una empresa por ello o esperar a una nueva versión como ocurre con el código cerrado.

## Versatilidad

- Dado que el código abierto se puede modificar, cualquier necesidad del usuario se puede desarrollar. Con el código cerrado el usuario depende de que la empresa desarrolladora decida implementar la funcionalidad que necesita. Por ejemplo en aplicaciones de código abierto existen traducciones a idiomas minoritarios que para el código cerrado no son rentables.

## **¿Cuál es el modelo de desarrollo del código abierto?**

El modelo de desarrollo del software de código abierto se basa en compartir el conocimiento.

Este es el modelo tradicional propio de los campos científicos y por ello fue el modelo inicial con que se desarrollo lo que hoy conocemos como internet, al igual que las herramientas que lo hicieron posible (tcp/ip, UNIX, C...)<sup>8</sup>.

En el modelo de desarrollo de código cerrado una empresa contrata a desarrolladores para realizar un proyecto y después vende en el mercado licencias de uso con restricciones (no se puede copiar, no se puede estudiar, no se puede modificar...).

En el modelo de desarrollo de código abierto una persona u organización (puede ser una empresa) coordina una amplia comunidad de desarrolladores independientes distribuidos por todo el planeta y el software se puede ver, probar y modificar aún antes de que salga una primera versión completa.

Al proceder los usuarios y desarrolladores de distintos entornos económicos, sociales y legales, el resultado obtiene mayor flexibilidad, adaptabilidad y versatilidad.

## **¿Cuál es el modelo de financiación del código abierto?**

Como ya se ha apuntado, las empresas que se dedican al OS no ingresan dinero por las licencias del software sino por el servicio que prestan a sus clientes.

Más concretamente, las empresas realizan sus ingresos por los servicios asociados al software, tales como formación, asesoría y consultoría, certificación, desarrollos a medida...

De esta manera los clientes no pagan por poder utilizar un programa (lo use mucho o poco), los clientes invierten en tener un proveedor de servicios. Así optimizan su inversión, ya que solo pagan por los servicios recibidos y se mantienen independientes del proveedor.

Por otra parte uno de los servicios que puede proporcionar un proveedor es desarrollar una funcionalidad inexistente en una aplicación.

De hecho con el código cerrado, mediante la venta de licencias, los clientes financian el desarrollo de un sistema y, generalmente, siguen pagando por su uso cuando ya se ha financiado completamente el desarrollo.

En el código abierto un cliente únicamente financia, si lo desea, la funcionalidad que necesita y no existe.

De este modo contribuirá además al desarrollo del sistema, beneficiando a otras empresas en la misma medida que el desarrollo financiado por otras empresas le ha ayudado.

## **Principales hitos en la historia del 'Software de Código Abierto'**

- A principio de los ochenta, Richard M. Stallman fue el primero en formalizar la filosofía del "Software Libre". Para ello creó la "Free Software Foundation" en 1984 y comenzó el proyecto GNU.

La aportación básica y más importante es la licencia GNU-GPL (GNU General Public License) que garantiza y protege la libertad de utilizar, estudiar, copiar y modificar el software.

- En 1991 un estudiante, Linus Torvald, comienza el desarrollo de 'Linux' bajo la licencia GPL, lo que acaba suponiendo un gran impulso para el proyecto GNU y el Software Libre en general.

- En 1998 Bruce Perens y colaboradores escriben la 'Definición de Código Abierto' ("Open Source Definition"). Su objetivo era describir las propiedades técnicas del Software Libre.

La misma 'Definición de Código Abierto' es un derivado de las 'líneas maestras del Software Libre Debian' ("Debian Free Software Guidelines"), que deriva de la GPL.

---

(<http://www.cnn.com/TECH/computing/9909/03/windows.nsa.02/>)

8 El Oxford English Dictionary fué desarrollado de forma cooperativa voluntaria y siguen pidiendo colaboraciones (<http://www.askoxford.com/worldofwords/oed/wordsearchers/>)

## Otras dudas...

**¿Quién responde del Software si 'no es de nadie'?**

**¿Me debo fiar simplemente del las buenas intenciones de un altruista?**

No. Hay pequeños desarrollos que son organizados por un desarrollador altruista, pero en proyectos más complejos el mayor impulso viene dado muchas veces por empresas.

Estas empresas, como ya se ha comentado, se benefician de crear un mercado más abierto y de su posición de expertos en dicho proyecto (puesto que son los principales desarrolladores). Es el caso de Zope Corporation, MySQL AB, ComPiere Inc., Red Hat Inc., SuSE Linux AG... o empresas 'tradicionales' como IBM o Novell.

El software siempre tiene un propietario. Está indicado en la nota de *copyright* o *copyleft*. Por eso es imprescindible con cada copia del software distribuir dicha nota.

**¿El OS proporciona realmente independencia del proveedor?**

Sí. El uso de OS proporciona independencia del proveedor.

Hay quien apunta que IBM está impulsando el uso de Linux en las empresas para que más tarde queden atadas a sus aplicaciones.

Por eso lo importante no es un sistema como Linux, sino el uso de licencias de código abierto. Por ello se habla de sistemas GNU/Linux (sistemas basados en GNU con Linux como núcleo).

Mientras un usuario utilice OS se mantendrá independiente del proveedor<sup>9</sup>.

Si decide utilizar código cerrado propiedad de IBM, Microsoft o cualquier otro, dependerá de las condiciones que éstos impongan en sus licencias, de los desarrollos que deseen hacer, etc.

**Pero el código abierto también me obliga a realizar actualizaciones ¿no?**

No. Se ha hecho notar por ejemplo que Red Hat no ofrece soporte para sus distribuciones antiguas, del mismo modo que Microsoft no ofrece soporte para las versiones antiguas de Windows.

Sin embargo hay dos diferencias muy importantes:

- Si Microsoft decide no dar soporte a una versión antigua de Windows, sin su permiso explícito nadie en ningún caso puede hacerlo, mientras que cualquiera puede mantener y soportar (o contratar para que se haga) una versión antigua de GNU/Linux.

De hecho Microsoft puede mantener, copiar y modificar cualquier versión de GNU/Linux mientras no cambie las condiciones de su licencia (mientras la mantenga como OS).

- La programas de código abierto, no requieren el pago de una licencia por la actualización. Cada versión del Windows de Microsoft requiere de un pago para obtener la licencia de uso y/o actualización.

**¿Se puede cerrar un programa que ya está abierto? ¿Y si GNU/Linux deja de ser OS?**

No es posible cerrar un programa abierto. Cuando un cliente recibe una copia de un software OS, recibe el código fuente y una licencia que la permite **sin limitaciones** usar, estudiar, copiar y modificar dicho software manteniéndose como libre.

Si Linus Torvald<sup>10</sup> quisiese cerrar Linux:

1.- No podría porque no todo el código es suyo. Necesitaría la aceptación de toda la comunidad de desarrolladores

2.- Aún en ese caso, cualquiera podría tomar una copia actual de Linux y seguir desarrollándola como software libre sin que nadie pudiese impedirlo. Eso es lo que garantiza tener el código y la licencia GPL.

## Sobre este documento

Este documento pretende únicamente realizar una introducción al 'Open Source' desde el punto de vista empresarial y comercial, una toma de contacto que incite a buscar más información.

Por ello este no es un documento técnico, ni mucho menos, sino que incide en las ventajas y desventajas que para un empresario tiene el OS.

Por último hay que decir que uno de los objetivos era que el texto ocupase únicamente 4 páginas para que fuese más atractivo (para que dé menos pereza leerlo ;). Es por eso que hay omisiones e inexactitudes.

---

<sup>9</sup> Algunas licencias y su compatibilidad con GPL: <http://www.fsf.org/licenses/license-list.html>

Categorías de Software Libre y No Libre: <http://www.fsf.org/philosophy/categories.es.html>

<sup>10</sup> Linus Torvald desarrolló el sistema Linux con licencia GPL. Sigue coordinando el desarrollo y es propietario de la marca Linux®.