

# Software libre para seres humanos

---

Una introducción al software libre para no informáticos

Por Luis Alberto Fernández Vallejo

Esta obra está bajo una licencia Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 España de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/2.5/es/> o envíe una carta a Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

## **Introducción**

En este artículo trataremos de explicar qué es el software libre a personas sin grandes conocimientos técnicos (los informáticos siempre se han considerado unos bichos raros, ¿no crees?). Por todo lo que el software libre significa e implica, creemos que tiene mucho que aportar a la sociedad. Aunque nunca llegues a saber realmente cómo funcionan los programas de ordenador, o cómo se escriben esos complicados lenguajes de programación es importante que, al terminar de leer este artículo, comprendas por qué el software libre juega un papel fundamental para todos nosotros.

### **Por qué los ordenadores son necesarios**

La informática es una rama del conocimiento técnico que se ha hecho un hueco en nuestra vida diaria, y de una manera consciente accedemos a ella y hacemos uso de sus recursos. La mayoría de nosotros tiene una dirección de correo electrónico con la que se comunica con sus amigos y familiares, usa un procesador de textos para escribir cartas, informes o documentos, consulta páginas web para buscar información sobre un tema de interés e incluso para hacer la compra. Usamos los ordenadores día a día y queremos que cuando nos sentemos delante del monitor, consigamos hacer aquello que nos propusimos de una forma productiva, es decir, en menos tiempo y con un mejor resultado final. Para esto tenemos que aprender a manejar los programas informáticos que utilizemos. De la misma manera que para poder conducir un coche necesitamos ir a una auto-escuela para adquirir unos conocimientos básicos de conducción (además de las consabidas horas de práctica al volante), para usar bien el ordenador de nuestra casa o del trabajo es necesario aprender cómo hacerlo.

Más allá de nuestro uso diario, sabemos que la informática y los ordenadores juegan un importante papel en las empresas (tanto pequeñas como grandes multinacionales), en las administraciones públicas, organizaciones, y un gran número de organismos. Si en algún momento las redes de comunicaciones entre ordenadores o las propias máquinas dejarasen de funcionar, podemos dar por seguro que sobrevendría un gran caos. No como en las películas de Hollywood, con explosiones incluidas, pero seguro que más de un quebradero de cabeza dará.

Los ordenadores forman parte de nuestra vida. Creemos que es necesario que tomemos consciencia de este fenómeno, de esta manera conseguiremos ser una parte activa del mismo. No seremos simples receptores pasivos de las decisiones empresariales, sino que tendremos capacidad propia de análisis y decisión. Y es aquí donde el software libre tiene mucho que decir.

## Una nueva palabra: software

Para seguir en nuestra explicación, es necesario que nos detengamos un momento para explicar un par de conceptos: *software* y *código fuente*.

El [software](#) de un ordenador es todo el conjunto de programas que usamos: desde el procesador de textos que estoy usando para escribir este artículo hasta el propio sistema operativo que controla todo el ordenador para que funcione como más o menos quiero que lo haga. Estos programas son realizados por programadores, quienes usan complicados lenguajes de programación para escribir los programas que usamos. Esta tarea que realizan los programadores da como resultado el [código fuente](#) del programa: las líneas de código que describen el funcionamiento del software, los métodos y pasos que lleva a cabo para realizar lo que se espera del mismo. Así, es lógico suponer que si miramos el código fuente de un programa, y conocemos el lenguaje en el que está escrito, podremos saber cómo está hecho, cómo funciona, y si el programador es un verdadero genio o un completo chapuzas.

Tras escribir el programa, para que éste sea útil y funcione, debe sufrir un proceso conocido como *compilación*, tras el cual se obtiene el programa “traducido” a ceros y unos, que es lo único que entiende una máquina<sup>1</sup>. Este paso no puede darse al revés, es decir, no es posible “traducir” de ceros y unos a código fuente (o muy complicado, y nunca se obtiene el código fuente original<sup>2</sup>)

Hoy en día, la constante que mueve la mayor parte del mundillo informático, por parte de las empresas desarrolladoras de software, es “yo te doy el programa ya compilado, para que puedas usarlo en tu ordenador, pero no el código fuente, el cual me reservo”.

## Software libre

Esto no siempre ha sido así. Antes, durante los años 60 y 70, el software era distribuido libremente, el código fuente pasaba de mano en mano para ser estudiado, copiado, modificado o, simplemente, curioseado. El negocio estaba en vender las máquinas, en enseñar a usarlas y en repararlas cuando se estropeaban. A partir de finales de los 70, las compañías productoras de software empezaron a poner restricciones al uso del mismo: no se podía modificar, no se podía ver el código fuente, o no se podía copiar el programa. Esto provocó una reacción en algunas mentes inquietas, y a principios de los 80 se creó el movimiento del *software libre*, auspiciado por [Richard M. Stallman](#), quien defendía la libertad del usuario para usar, copiar, estudiar, modificar y redistribuir libremente el software. Stallman creó el concepto de [copyleft](#), en contraposición al *copyright*, y la [Fundación del Software Libre](#) (Free Software Foundation, FSF). Al mismo tiempo, él y otros que pensaban de la misma manera, iniciaron el [movimiento GNU](#), un intento por crear un sistema informático completamente libre.

En aquel momento había nacido el movimiento del software libre, fruto de la añoranza de la antigua necesidad de compartir código e información durante los años 60 y 70, y abanderando los principios de libertad y libre intercambio de información.

### **Y todo esto, ¿qué importa hoy?**

Hoy en día, el mundo del software puede dividirse, a riesgo de simplificar la situación, en privativo y libre<sup>3</sup>. El software privativo es el desarrollado por organizaciones que venden o regalan su producto sin publicar el código fuente del mismo. De esta manera nadie sabe cómo funciona el programa internamente. Además de esto, al usar este programa, aceptas un acuerdo de licencia que establece que no podrás copiar, modificar, distribuir o regalar el programa. Dependiendo de la licencia, se limitan o permiten ciertas acciones (¿has leído las licencias de los programas que usas? No estaría de más que lo hicieras; puede que te llesve alguna sorpresa. Una vez tropecé con una licencia que prohibía, explícitamente, hablar mal del programa en público<sup>4</sup>). En definitiva, la licencia sólo permite *usar* el programa, te impide regalar una copia a un amigo o modificarlo si te sientes capaz. ¿Comprarías un coche con el capó cerrado, al que no pudieras cambiar ninguna pieza? Aún peor: ¿comprarías un coche que sólo pudieras reparar en un taller concreto y sólo cuando los mecánicos de ese taller decidieran que el coche necesita un arreglo? Entonces, ¿por qué usas programas privativos?

En la otra cara, está el software libre, que permite usar, copiar, estudiar, modificar y redistribuir libremente los programas, además de permitir el libre acceso al código fuente para saber cómo está hecho. Puedes usar el programa sin ningún tipo de limitación, compartirlo entre tus amigos, modificarlo, adaptarlo a tus gustos y contribuir a la mejora del software devolviendo tu trabajo a la comunidad, con lo que todos nos beneficiamos. La licencia de los programas libres posibilita todo esto, defendiendo y protegiendo los derechos del usuario<sup>5</sup>.

Es importante recalcar que los programas libres no siempre tienen por qué ser gratuitos, aunque usualmente lo son (de la misma manera que los programas privativos no tienen por qué costar dinero) Hay que recordar que *libre* no es lo mismo que *gratis* (esta confusión se debe a que en inglés, la palabra “free” de *free software* tiene ambas connotaciones). De cualquier forma, siempre será legal el regalar copias de un programa libre, a pesar de que hayas pagado por él.

Hoy, el software libre es una realidad cada vez más asentada<sup>6</sup>. Se usa en las grandes corporaciones, en las administraciones, en pequeñas y grandes empresas, en los súper-ordenadores e, incluso, en el ordenador personal de tu vecino. El 70% de las veces que te conectas a una página web, el programa que se encarga de “servirte” es libre (el servidor web [Apache](#)). Algunos usan el software libre porque creen que es técnicamente superior que el privativo. Otros lo usan porque piensan que es una forma ética o moral de usar los ordenadores. Hay algunos que lo usan, simplemente, para alimentar su orgullo de *frikies*. Todos

tienen sus propias y respetables razones.

¿Qué aporta el software libre? O, mejor dicho, ¿qué nos puede aportar el software libre a nosotros, simples usuarios de ordenadores? Se puede responder esta pregunta desde varias perspectivas diferentes, cada persona debe elegir aquella que considere más importante.

En primer lugar, nos aporta ética. Sabemos que éste es un concepto muy devaluado y tergiversado hoy en día por algunos (a quienes recomendaríamos un par de lecturas<sup>7</sup>), pero para otros sigue siendo importante. Es la razón fundamental por la que se inició todo este movimiento. Es ético y justo que la información deba ser libre, que todo el mundo tenga acceso a la misma, y esto incluye el código fuente de los programas de ordenador. Es uno de los principios fundamentales del [movimiento hacker](#), de la [cultura libre](#) y de las licencias [Creative Commons](#).

En segundo lugar, aporta avance tecnológico. Los programas de software libre son desarrollados por una persona, o un grupo más o menos numeroso de personas (dependiendo de la envergadura del proyecto). Sin embargo, como el código fuente de los programas es accesible por todo el mundo, literalmente cualquier persona con los conocimientos suficientes puede corregir algún error, incorporar alguna nueva y fascinante funcionalidad o, simplemente, leer el programa para constatar que está bien hecho y recomendarlo a sus amigos. Es simple matemática: a mayor número de personas colaborando en algo, más probabilidades hay de que los errores se subsanen rápidamente y/o el producto se mejore. Y de una mejora o corrección se beneficia, automáticamente, toda la comunidad, debido a su carácter abierto.

En tercer lugar, aporta ahorro económico. Ya sea tecnológicamente superior, o similar, a los programas privativos (en todo caso, casi nunca inferior, más que en contadas ocasiones), tenemos un software que tiene un coste de producto casi cero. Esto no significa que todo sea gratuito, simplemente se apuesta por otro modelo de negocio, basado en el trabajo de instalar, configurar, prestar servicio técnico o asesoramiento. Es cierto que la mayor parte de las personas, no nos engañemos, no paga el software que usa. Aunque tenga un coste monetario bastante elevado, el sistema operativo de las ventanas o los programas ofimáticos más usados tienen un precio, en la práctica, de 0 euros. [Pero no siempre va a ser así](#).

Sea por la razón que sea, cada vez más personas cambian sus programas privativos por alternativas libres. Lo importante es hacerlo con una motivación. Recomendamos a cualquier persona interesada que bucee en la pequeña bibliografía de introducción que aportamos, la cual ampliará muchos de los conceptos presentados aquí y ayudará a asentar otros.

Hemos querido dejar de lado la explicación sobre GNU/Linux, ya que el objetivo de este artículo únicamente es mostrar qué es el software libre. Puede que hablemos sobre este sistema operativo, el más

usado y conocido del software libre (aunque no el único), en un próximo artículo.

## BIBLIOGRAFÍA RECOMENDADA

---

- Richard M. Stallman, “*Software libre para una sociedad libre*”. Ed Traficante de Sueños (también disponible en <http://biblioweb.sindominio.net/pensamiento/softlibre/>)
- Richard M. Stallman, “*El derecho a leer*”. Un pequeño cuento futurista. Puede leerse en <http://www.pps.jussieu.fr/~dicosmo/Piege/ES/www.glug4.linux.org.ar/documentos/derechoaleer.pdf>
- Eric S. Raymond, “*La catedral y el bazar*”. Un artículo que explica los modelos de desarrollo del software libre (el bazar) y el software propietario (la catedral). En la red está disponible en la dirección [http://es.wikisource.org/wiki/La\\_Catedral\\_y\\_el\\_Bazar](http://es.wikisource.org/wiki/La_Catedral_y_el_Bazar) También hay disponible una crítica en [http://biblioweb.sindominio.net/telematica/critica\\_esr.html](http://biblioweb.sindominio.net/telematica/critica_esr.html)
- Roberto Di Cosmo, “*Trampa en el cyberspacio*”. Ensayo en el que se analiza la situación actual del mundo de la informática. Recomendable para entender los peligros del software propietario. En <http://www.pps.jussieu.fr/~dicosmo/Piege/ES/www.glug4.linux.org.ar/documentos/trampacyberspacio.pdf>
- Juanomás García y Alfredo Romeo, “*La pastilla roja*”. Editorial open:service. El libro intenta mostrar el por qué el software libre debería ser el fundamento de la informática en el futuro. Se estudian los principios ideológicos y las repercusiones económicas de su implantación. Es posible solicitar un ejemplar en [http://www.lapastillaroja.net/formulario\\_de\\_pedido.php](http://www.lapastillaroja.net/formulario_de_pedido.php)

## NOTAS

---

1. Existen lenguajes en los que no es necesario compilar para ser ejecutados. Es lo que se conoce como *lenguajes interpretados*.
2. Mediante ingeniería inversa es posible saber, con mayor o menor acierto, cómo funciona un programa compilado. Es un proceso dificultoso, como si se jugara una partida de ajedrez al revés.
3. También puede encontrarse los términos de *software cerrado* y *software de fuente abierta*. Aunque son similares, estos términos tienen algunas diferencias sutiles que a algunas personas les parecen muy importante. Para una definición de *fuentes abiertas* visite [este enlace](#), y para una discusión de ambos conceptos, visite la [web de GNU](#)
4. No es broma, la licencia de SQL Server establece que no se puede criticar el producto en público. En la [web de Kriptópolis](#) se habla de las licencias y lo que se acepta con ellas.

5. Existen muchos tipos de licencias. Aunque la más extendida sea la [Licencia Pública General de GNU \(GPL\)](#), existen [muchas otras](#).

6. El servidor web más usado del mundo es Apache, con [una cuota cercana al 70%](#). El servidor DNS omnipotente en Internet es BIND. El navegador Firefox ya se ha comido un [10% de cuota de mercado](#). El sistema operativo más usado [en los servidores públicos de Internet](#) es GNU/Linux, además de ser el más usado entre los [500 ordenadores más potentes del mundo](#).

7. Para iniciarse, nada mejor que leer “Ética para Amador”, de Fernando Savater. Para profundizar más sobre el tema, recomendamos “Ética para náufragos”, de José Antonio Marina. Aunque de éste último autor sería bueno leerse toda su obra :-). Además, para un estudio sociológico sobre la motivación de los hackers, recomendaríamos sin dudar [“La ética del hacker”](#)