



Revista Mexicana de Ciencias Políticas y  
Sociales

ISSN: 0185-1918

articulo\_revmcpys@mail.politicas.unam.mx

Universidad Nacional Autónoma de México

México

Mochi Alemán, Prudencio Óscar  
El movimiento del software libre

Revista Mexicana de Ciencias Políticas y Sociales, vol. XLV, núm. 185, mayo-agosto, 2002, pp. 73-89

Universidad Nacional Autónoma de México

Distrito Federal, México

Disponible en: <http://www.redalyc.org/articulo.oa?id=42118505>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# El movimiento del *software* libre

PRUDENCIO ÓSCAR MOCHI ALEMÁN\*

## Resumen

El objetivo de este trabajo es describir el surgimiento y desarrollo del movimiento de *software* libre. Para ello se analizan algunas características importantes que dan cuenta del origen y desarrollo del mismo, así como de sus momentos más críticos y sus tensiones, sus desafíos y sus retos. Otra de las grandes inquietudes de este trabajo, es intentar descifrar cómo el movimiento del *software* libre y Linux, en su versión más conocida, sobrevivieron y se afirmaron hasta llegar a constituir una verdadera alternativa frente a las grandes empresas de la alta tecnología.

## Abstract

The objective of this work (The Wave of Free Software), is to describe the on going and development of the wave of free software. For that, we analyze some of its most important characteristics that give us the beginning and development of free software; thus like it's most critical and tension moments; its challenges and goals. An other huge part of this work tries to decipher how the wave of free software and Linux, on its most common version, survived and reaffirmed their position on the way to reach a real alternative for high technology big companies.

*Palabras clave:* *software*, Movimiento del Software Libre, industrias de *software*, Model Instrumentation, Telemetry Systems (MITS), Altair, Linux.

---

## El desarrollo del *software* libre

**E**l desarrollo de la producción de *software* estuvo signado, desde sus orígenes, por actores de la sociedad civil que formaron lo que hoy se conoce como el movimiento del *software* libre. En sus orígenes, tanto como en la fase actual, el desarrollo de este movimiento tuvo como protagonistas a un grupo de profesores, jóvenes estudiantes y trabajadores universitarios, creativos y rebeldes, quienes comenzaron a organizarse para realizar aquel sueño utópico de socia-

\* Centro Regional de Investigaciones Multidisciplinarias, Av. Universidad s/n, circuito N° 2, Campus UAEM, Col. Chamilpa, c.p. 62210, Cuernavaca, Morelos.

---

---

lizar la comunicación, de poner “una computadora al alcance de todos”, de crear un *software* para ser distribuido de manera gratuita. Se trataba de un movimiento paralelo y con características similares, en sus orígenes, al de Internet, caracterizado como “movimiento libertario, contestatario, con el que buscaban un instrumento de liberación y de autonomía respecto al Estado y a las grandes empresas” (Castells, 2001).

Las reflexiones que abordaré en este artículo son parte de una investigación más general sobre la industria del *software* en México,<sup>1</sup> en el contexto latinoamericano e internacional. De las primeras aproximaciones a este tema, se desprende que, además del gran desarrollo y protagonismo que involucra a este sector, así como de la importancia que emerge sobre sus datos cuantitativos y cualitativos, es destacado el papel y el protagonismo de las organizaciones de la sociedad civil, las cuales, a través de fundaciones, asociaciones, universidades, ciudadanos aficionados y programadores independientes, contribuyen a la creación, experimentación y producción de *software*. Dentro de este sector, un ámbito importante en la producción de *software*, es el de los voluntarios, quienes con su trabajo cooperativo,<sup>2</sup> contribuyen a la producción de *software* libre, como veremos en este artículo.

Uno de los componentes fuertes de la globalización es la vinculación entre cambios de formas de organización de la sociedad, la economía y la cultura, relacionados con las modificaciones en las nuevas tecnologías de la información y comunicaciones que desembocaron en la revolución tecnológica y su influencia en todos los ámbitos de nuestra sociedad. A partir de los años sesenta y setenta dicha revolución se acentuó e interactuó con el fenómeno de crecimiento de las organizaciones de la sociedad civil y al reclamo que las mismas hacían y hacen por la ampliación de los derechos cívicos, políticos y culturales, por parte de la ciudadanía. Las nuevas relacio-

<sup>1</sup> Los objetivos de esta investigación son el estudio cuantitativo de las industrias de *software* en México, a nivel internacional y en el contexto latinoamericano (producción, exportación, importación, capital accionario, etc.), así como el estudio cualitativo respecto a sus formas de organización, las características y perfil de sus trabajadores, políticas públicas implementadas para su desarrollo.

<sup>2</sup> Para ampliar más este tema se puede consultar el artículo de Aris Papathéodorou y Laurent Moineau, “Cooperation et production immatérielle dans le logiciel libre”, *Revista Multitudes*, marzo de 2000. Versión en línea <[www.samizdat.net/slut/textes/mult\\_coopprod.html](http://www.samizdat.net/slut/textes/mult_coopprod.html)>.

---

---

nes sociales que se crean dentro de estos espacios virtuales, a través de las redes electrónicas, los grupos de discusión, así como toda la comunicación mediatizada por computadoras y las herramientas brindadas por las TIC, constituyen un nuevo espacio público con nuevas características y oportunidades, en el cual se desarrollará el movimiento del *software* libre (Mochi Alemán, 2001).

El objetivo de este trabajo es describir, en un primer momento, el surgimiento de estos actores de la sociedad civil, sus objetivos y su ambiente cultural, que conformarán el movimiento del *software* libre. En un segundo momento, nos detendremos en algunas de sus características consideradas importantes para entender su desarrollo, así como en algunos de sus momentos más críticos y conflictivos, sus desafíos y retos. Otra de las grandes inquietudes de este trabajo, es intentar descifrar cómo el movimiento del *software* libre y Linux, en su versión más conocida, sobrevivieron y se afirmaron hasta llegar a constituir una verdadera alternativa frente a las grandes empresas de la alta tecnología.

Se trata, por lo tanto, de un trabajo de carácter exploratorio, que si bien se desprende de la investigación citada anteriormente y de una revisión bibliográfica, su característica es observar y analizar un proceso y sus distintos momentos, por lo que es probable que en su desarrollo se encuentre la forma de una sucesión de hechos que conforman, en parte, la descripción de la historia del movimiento del *software* libre dentro de la historia más general de la producción de *software*.

### **Los orígenes del movimiento**

En 1974, Ed Robert, titular de una firma de Albuquerque, Estados Unidos, que producía relojes digitales y calculadoras electrónicas de bolsillo, conocida como MITS, por sus siglas en inglés (Model Instrumentation Telemetry Systems), tuvo la idea de montar alrededor de un chip 8080 (que se empleaba para hacer funcionar semáforos y ascensores) fabricado por Intel, compañía de Santa Clara, California, un sistema de conexión, con una memoria y algunas puertas de ingreso y salida de datos, de manera que pudiera funcionar todo como una pequeña computadora. Era del tamaño de una radio de transis-

---

---

tores. La llamó Altair, nombre de una estrella. El movimiento lo difundió con gran velocidad en toda la comunidad de Berkeley y Palo Alto. Se presentó en la portada de un número de *Popular Electronics*, periódico dirigido a los aficionados a la electrónica, cuyo director era Lee Salomon en enero de 1975, otro talento tecnológico del movimiento.

[...] Era el sueño de cualquiera que tuviese el *hobby* de la computadora, y a un precio muy accesible, el Altair ofrecía las mismas potencialidades de otros elaboradores que podían costar de diez a cien veces más. Pero ¿cuál era su secreto? El Altair funcionaba gracias a un microprocesador, juntaba todos los elementos esenciales de una computadora en una placa de silicio de un tamaño no más grande que una uña. Esta placa constituía el corazón del Altair [...] (Zachary, 1994).

Con el Altair la computadora se transformaba en un bien de consumo masivo. Al presentar Altair, los redactores de la revista *Popular Electronic*, habían sentenciado: “Llegó la era de la computadora para cada casa”.

Formaban parte de este movimiento un grupo de profesores y estudiantes universitarios, principalmente de la universidad de Berkeley: Lee Felsestein, Bob March, Fred Moore, Bob Albrecht, entre otros. Entre ellos se encontraba Bill Gates, que venía de la universidad de Harvard, todos excelentes matemáticos e informáticos. Bill Gates y Paul Allen (un compañero suyo de la escuela superior) se darían cuenta del gran defecto del Altair: no tenía *software*. La persona que lo adquiría debía construir su *software*, de no ser así, la máquina resultaba inútil. El Altair necesitaba un lenguaje para crear los programas.

Gates y Allen se pusieron inmediatamente en contacto con la empresa de Albuquerque que había construido el Altair, y le propusieron proveerle un lenguaje. Eligieron el Basic (originalmente desarrollado en los años sesenta para la mini computadora producida por la Digital). La autoría del Basic (Beginners All-purpose Symbolic Instruction Code), pertenecía a dos profesores de la universidad de Dartmouth, quienes nunca habían hecho valer sus derechos de propiedad sobre el programa y consentían a quien quisiera usarlo y modi-

---

---

ficarlo gratuitamente, era el lenguaje ideal para programas cortos y era el más fácil de usar. Bastaron seis semanas para que Gates y Allen realizaran una versión del Basic para el Altair, la privatizaran y comercializaran transformando así al *software* en una actividad privada altamente rentable. Este hecho constituye la base del nacimiento de una de las sociedades que se constituirá en la más famosa y conocida dentro de la industria del *software*: Microsoft, la encargada de comercializar dicho programa.

Los apasionados de la época, que formaban parte del movimiento, decidieron copiarlo y difundirlo de manera gratuita, sosteniendo que el *software* era un bien público. Consideraban que el *software* vendido por Gates debería ser de libre circulación como los demás, como siempre había sido hasta el momento. Gates, en violento desacuerdo “con la idea de que el *software* fuese el equivalente electrónico del aire o del agua”, denunció abiertamente a los que difundían el sistema operativo construido a través del lenguaje Basic como piratas. Escribió una *Carta abierta sobre la piratería*, en la que los acusaba de robo, y explicaba que sin una adecuada remuneración, nadie podría producir más *software* profesional. Los aficionados o *hobistas* del *software*, le respondieron que el Basic que él había empleado para su propio sistema operativo había nacido en forma gratuita, en el ámbito de instituciones financiadas con fondos públicos, hecho por informáticos que no pretendían ninguna remuneración, y que además había sido continuamente mejorado por centenares de programadores anónimos. Pero a Gates, poco le interesaba los argumentos de esta “ética comunitaria”.

Ese fue el inicio para que comenzaran a difundirse miles de programadores por el mundo, desarrollados en verdaderas fábricas de *software*. Junto a miles de hombres dedicados a la mercadotecnia, capaces de difundir el nuevo descubrimiento. El *software* pagado parecía ser una elección obligada para todos.

Sin embargo, durante todos esos años el movimiento continuó cultivando, pública y polémicamente, la idea de *software* como bien común, de libre y gratuita circulación. El movimiento de *software* libre empezó mucho antes de que Microsoft se convirtiera casi en un sinónimo de *software*. Richard Stallman, un legendario programador del Massachussets Institute of Technology (MIT), fue fundador de una organización: la Free Software Foundations (1984), Stallman sos-

---

---

tenía que mantener en secreto el código fuente era “un pecado y un crimen contra la humanidad”. Escribió un manifiesto en el que establecía algunos preceptos y recalca con mucho énfasis que los programadores siempre lo habían compartido todo en las décadas de los años cincuenta, sesenta y setenta. Para Stallman,<sup>3</sup> el mundo de los ordenadores debía volver a los viejos tiempos en que el código era libre y se podía compartir. El manifiesto de Stallman era una reacción a la tendencia por parte de las empresas privadas de mantener el código secreto, ya que consideraban que sus clientes estaban igualmente contentos sin saberlo. *Microsoft* estaba en pleno desarrollo y comenzaba a tomar el monopolio del mundo del *software*.

Un programador que comparte el código fuente permite aprender a los demás, y esos otros pueden contribuir aportando ideas a su vez al conjunto. Los códigos fuente cerrados dejan frustrados a los usuarios, porque éstos no pueden aprender nada del *software*, ni corregir errores (Wayner, Peter, 2001:25).

Tal vez para entender mejor la importancia de compartir el código fuente, es decir tener acceso al mismo y poder modificarlo, es necesario explicar qué se entiende por *software*, cómo funcionan y qué es el código fuente. Es difícil adoptar una definición de *software*, en parte, por el carácter intangible de los productos y, en parte, por los cambios tecnológicos. Esto hace muchas veces que se haga difícil de precisar si es un producto o un servicio. Independientemente de estas consideraciones que excede los objetivos de este artículo, podemos definir al *software* como conocimiento e información codificado, y en su definición más corriente propuesta por la OECD (1985), y similar a la empleada por la ISO (International Standardisation Organisation) y la WIPO (World Intellectual Property Organisation) por *software* se entiende la “producción de un conjunto estructurado de instrucciones, procedimientos, programas, reglas y documentación contenida en distintos tipos de soporte físico (cinta, discos, circuitos eléctricos, etc.) con el objetivo de hacer posible el uso de equipos de procesamiento electrónico de datos”. Básicamente, el

<sup>3</sup> Stallman tiene en la actualidad su página web: <[www.stallman.org](http://www.stallman.org)>.

---

---

*software* es un plan de funcionamiento para un tipo especial de máquina, una máquina “virtual” o “abstracta”. Una vez escrito mediante un lenguaje de programación, el *software* puede funcionar en computadoras, que temporalmente se convierten en esa máquina para la que el programa sirve de plan. Sin este conjunto de instrucciones programadas, las computadoras serían objetos inertes, sin capacidad de poder mostrar algo en la pantalla.

Ahora bien, las computadoras sólo procesan un lenguaje llamado binario (llamado técnicamente código-máquina), el cual no sirve para que las personas puedan comunicarse. Por otra parte es necesario aclarar que los programadores ya no usan este código binario, utilizado a los inicios de desarrollo del *software*, ahora escriben las instrucciones que ha de ejecutar la computadora mediante lenguajes llamados formales, de “alto nivel”, con rígidas reglas sintácticas que lo asemejan a los lenguajes lógicos-formales. Este sistema le facilita a los programadores la tarea de escribir programas, pero para que sean compresibles a las computadoras, deben ser convertidos antes a código-máquina. Esta conversión se realiza con programas especiales, llamados “compiladores”. A lo que escribe el programador se le denomina “código fuente”. El resultado de la conversión “compilación” en lenguaje-máquina, se lo denomina “código objeto”, “binarios” o “ficheros ejecutables”. La distinción entre fuentes y binarios es clave para entender el empeño de los partidarios del *software* libre por disponer de las fuentes, tener acceso y poder modificarlas ([www.sindominio.net](http://www.sindominio.net)).

La base principal del movimiento de *software* libre es la de compartir este código fuente del *software*, y hacerlo girar y distribuir por Internet. El *software* de código abierto flota por Internet controlado por un cierto número de licencias con nombres como GNU (General Public License-GPL);<sup>4</sup> la manera en la que funciona es que la licencia GPL GNU permite a los usuarios modificar el programa y distribuir sus propias versiones. Y sobre todo, obliga a compartir el código fuente, más que controlar o privatizar el mismo.

Una de las primeras distribuciones de código fuente de sistema operativo fue creado también en Berkley, siguiendo una gran tradición de búsqueda y desarrollo de sistemas operativos. Su nombre

<sup>4</sup> Versión original en inglés, única con valor legal: <[www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html)>

---

---

fue BSD (Berkley Software Distribution). “Este paquete creció, se transformó y fue absorbiendo muchas otras contribuciones a lo largo de los años” (*idem*).

De manera paralela, existía ya otro sistema operativo como el de Microsoft, que se usaba sólo para grandes computadoras, conocido como UNIX, diseñado y construido en los laboratorios Bell de la casa telefónica AT&T, a lo largo de la década de los setenta. Como la empresa inicialmente se dedicaba a los teléfonos, UNIX se concedía gratuitamente a todos, especialmente a los centros de investigación universitarios. Posteriormente AT&T comenzaría a interesarse en privatizar el código, lo que significó épocas de grandes conflictos con la universidad de Berkley, principalmente con el grupo del departamento de Informática, con quienes AT&T compartía la versión.

Stallman siempre había soñado con sustituir el UNIX por algo que fuera igual de bueno, acompañado además del código, pero era una larga tarea. Esa fue la razón por la que inició el proyecto GNU. Lentamente pero de forma segura, el proyecto GNU iba ensamblando las partes para hacer que funcionase. Había centenares de pequeñas utilidades y herramientas mayores donadas al proyecto GNU, y todos esos pequeños fragmentos se iban añadiendo (*idem:63*).

Stallman llamó su trabajo GNU porque en realidad es un acrónimo recursivo que significa “*Gnu is Not Unix*”, o GNU no es UNIX. El proyecto estaba destinado a generar un sistema operativo que hiciese todo lo que hacía UNIX, pero que no costase nada.

### **El mundo de Linux**

En otro lugar del mundo, un finlandés llamado Linus Torvalds, quien nada sabía sobre los pleitos de AT&T con la universidad, ni de la historia del BSD, comenzaba su búsqueda sobre cómo crear un sistema operativo. No podía permitirse comprar una máquina con sistema UNIX, por lo que había decidido crear su propia versión. Sólo se podía permitir una PC básica con un procesador 386. La General Public License (GNU) “había plantado su semilla en el cerebro de Torvalds, y éste pensaba que cobrar por el *software* era una limitación”.

---

---

Contemporáneamente, el centro de investigación de Computer Systems Research Group de Berkley, en junio de 1991, difundió una versión completamente libre de cargas de UNIX BSD, conocida como Network Release 2, cuyas versiones podían instalarse a la 386. Las actuales versiones FreeBSD y NetBSD, son el resultado de este mismo proyecto. Linus Torvald nunca se enteró de este proyecto: “nunca habría empezado el Linux si hubiese sabido que podía descargar un sistema operativo más completo de Berkley”, declaró en una oportunidad.

Hasta aquel momento, UNIX era uno de los mayores sistemas operativos del mundo. Los ordenadores de IBM, Sun, Apple, y la mayor parte de los fabricantes de máquinas de gran potencia lo utilizaban. Todos estos fabricantes tuvieron la necesidad de unificar el mercado estandarizando UNIX y los llamaron POSIX. Torvalds seguía trabajando con su proyecto, hasta que logró crear un sistema que funcionaba con un compilador, tomando cosas prestadas del proyecto GNU. En 1991 Linus Torvald creó el núcleo o *kernel* de Linux.<sup>5</sup>

A principios de 1992, Linux ya no era simplemente un *hobby* de un estudiante finlandés. Varios programadores influyentes se habían interesado por el código que era libre y relativamente manejable. Ejecutaba gran parte del código GNU y permitía experimentar con algunas herramientas excelentes de una forma neta y barata. Cada vez había más gente que descargaba el sistema, y una parte significativa empezó a comunicar fallos y sugerencias a Torvalds. Éste les iba comprometiendo a su vez y el proyecto creció como una bola de nieve (Wayner: 96).

Torvalds decidió poner en práctica la idea de recurrir a la inmensa comunidad de apasionados, programadores y *hackers*,<sup>6</sup> invitándolos a poner en red sus competencias y deseos de elaborar *software* y

<sup>5</sup> El núcleo en un ordenador es el responsable de mantener la fluidez de datos entre los discos duros, la memoria, las impresoras, la pantalla de video y todas las partes que se hallan unidas al mismo.

<sup>6</sup> *Hacker*: término de difícil traducción, mitad genio y mitad pirata, entre virtuoso de la computadora y pirata tecnológico. Steven Levy, les dedicó su libro *Hackers. Heroes of the Computer Revolutions* (1984), donde explica que el término deriva originalmente de la expresión *hack*, que se usaba para indicar bromas refinadas que los estudiantes se hacían periódicamente en la época del surgimiento de las computadoras, después se fue extendiendo a toda la comunidad científica que demostraba innovación, virtuosismo y estilo técnico.

---

---

compartirlo. El modelo de trabajo cooperativo, voluntario y difuso fue conquistando espacios cada vez con mayor popularidad y crédito. Un proyecto de tal magnitud no hubiese podido ser llevado a cabo por una sola persona. Pero en ningún momento Linus Torvald se atribuyó la paternidad ni poder alguno sobre Linux, “el único control efectivo que he mantenido siempre sobre Linux es conocerlo mejor que nadie” (*idem*:97).

La novedad y originalidad de Linux consiste en que Torvalds decidió colocar el código fuente de su proyecto bajo la licencia GNU (General Public Licence) de Richard Stallman, conocida como GPL. La licencia establecía que todo el código fuente que Torvalds o cualquier otra persona escribiera, debía ser accesible con entera libertad y compartido por todos los usuarios. El código fuente se convirtió así en la clave del *software* libre. Nadie puede tomar el control, ni ponerle impedimentos, vallas o ligaduras.

Todo el trabajo de Stallman y sus discípulos podía ser fácilmente conectado para que trabajase con el núcleo del sistema operativo, al que Torvalds ya llamaba Linux. Ése es el poder que tiene el código fuente para ser distribuido libremente. Cualquiera puede conectarse, de modo que algunos, inevitablemente lo hacen (*idem*:99).

El movimiento de *software* libre con Richard Stallman, Linus Torvalds y muchos programadores anónimos y menos conocidos, lograron hacer posible para todo el mundo la idea de usar un *software* sin pagar licencia a nadie.

### **¿Libre, gratis, abierto?**

Hay una serie de palabras, cuyas acepciones, al ser aplicadas en el contexto del movimiento del *software* libre, se prestan a confusión. La palabra “libre” (*free*), por ejemplo, no significa sólo “gratis” para el movimiento, sino más bien describe una forma de vida.

Para muchos de ellos, la revolución del *software* libre fue concebida en libertad y dedicada a determinados principios, como el hecho de que todos los hombres y mujeres tienen derechos

---

---

inalienables a intercambiar, modificar y hacer lo que quieran con su *software* en persecución de la felicidad (*idem*:121).

Muchos prefieren usar *software* abierto para describir el proceso de intercambio. El movimiento no acentúa la importancia del valor económico “gratis”, diferenciándose, de esta manera, de las empresas que ofrecen *software* gratis como mecanismo de conquista del mercado. El *freeware*, se distribuye por Internet, y es producido por grandes empresas para afianzar su presencia en el mercado del *software*, por ejemplo, el caso de los *browsers* (Netscape e Internet Explorer) o de correo electrónico (Eudora). También existen *freewares* producidos por programadores independientes y otros por unidades académicas. “Gratis” se transforma así, en un dispositivo de *marketing* para poder entrar en un mercado mediante el obsequio de un producto y poder difundir una marca. O, en otros casos, sirve para poder difundir los programas *shareware*,<sup>7</sup> para sus propios *Curriculum vitae*. En cambio, para el sector académico, son generalmente trabajos pioneros en un área para después entregar o vender las iniciativas al sector privado (Fritz, Alejandro, 1998). En cambio, “libre” para el movimiento significa fundamentalmente la difusión del código fuente (modificable), que es absolutamente otra cosa. Richard Stallman definió muy bien la palabra libre y sus matices, codificándola en cuatro principios básicos:

1. Libertad de ejecutar el programa para cualquier finalidad;
2. Libertad de estudiar cómo funciona el programa y adaptarlo a las propias necesidades;
3. Libertad de redistribuir copias para ayudar al vecino;
4. Libertad de mejorar el programa y publicar las propias mejoras, para que se beneficie de ellas toda la comunidad.

Otras de las ideas que pertenecen a Stallman son que él podía controlar estrictamente el uso del *software* GNU con la licencia legal GNU General Public License o GPL, la cual nunca fue una *copyright*, sino que se trata más bien de una antilicencia, definida como *copyleft*. El GPL es un documento legal que no coloca al *software* bajo nin-

<sup>7</sup> *Software* de uso compartido, es entregado a modo de prueba por los desarrolladores al público, si lo desean, para continuar su uso. Se puede adquirir pagando un único derecho de uso. *Shareware* no es de dominio público.

---

---

gún “dominio público”. Registra los derechos de *software* y luego otorga a los usuarios derechos para hacer innumerables copias, mientras no se dañen los derechos de otras personas a usar el *software*. GPL no obliga a publicar los cambios que uno hace al código fuente, sólo obliga a publicar las modificaciones si el mismo se distribuye.

Otro código fuente disponible es el BSD (Berkeley Software Distribution). Se puede entrar a él con costos muy bajos, es usado principalmente por empresas y éstas sólo pueden cobrar por el valor de lo que añaden al código.

El debate entre la libertad estilo BSD y estilo GNU, es uno de los más enconados en el mundo de la programación libre, y además está condenado a continuar durante mucho tiempo todavía, mientras los programadores se apunten a uno u otro bando y vayan experimentando (*idem:148*).

Otra gente del movimiento, en 1998, trató de unificar las dos posturas creando un nuevo término: “código abierto” ([www.open-source.org](http://www.open-source.org)). Sin embargo, en 1999 esta iniciativa aún no había conseguido registrar el término “código abierto”, como marca registrada porque era una expresión demasiado común. Para Stallman, “abierto” no captaba la esencia de “libertad”. Para muchos del movimiento, se llega entonces a una especie de términos de compromisos, usar las palabras “código libre” y “*software* abierto” es casi sinónimo. Otros prefieren no detenerse en el significado de “libre”, y se refieren al *software* protegido por la GPL como “GNU libre”.

Como hemos podido observar, detrás de cada palabra, existe un contenido y una acción, vinculadas a corrientes filosóficas de pensamiento dentro del mundo de este movimiento. La esencia del movimiento, sin embargo, se manifiesta en su libertad, sin importar cuánta gratuidad haya en el valor atribuido.

### **El modelo Bazar**

Eric S. Raymond en su clásico artículo “La catedral y el bazar” (1997) explicó la diferencia entre el *software* propietario y el *software* libre. Para Raymond el mundo del *software* libre es como un bazar, con muchos comerciantes diferentes que ofrecen sus mercancías. El mo-

---

---

delo de *software* propietario corresponde al “modelo catedral”, rígidamente estructurado y jerarquizado. Estructurado como los sindicatos religiosos que construyeron las catedrales medievales. Las catedrales estaban sometidas a la dirección de jerarquías sacerdotales, que aprovechaban la riqueza de la ciudad para construir el proyecto de un solo arquitecto. A juicio de Raymond, el modelo bazar es mucho más eficaz y produce un *software* de mayor calidad con menor gasto de recursos.

Si bien es cierto que la metáfora de Raymond tuvo mucha difusión, y sus imágenes de la catedral y el bazar tenían tal plasticidad que muchos terminaron dividiendo el mundo en catedrales y bazares, en realidad las dos imágenes están bastante mezcladas, ya que por ejemplo el modelo catedral no sólo corresponde a la industria del *software* propietario, sino a algunos grandes desarrollos libres avalados por la FSF.

Algunos críticos de Raymond incluso consideraron el nombre catedral desafortunado para describir el fenómeno, pues la construcción de catedrales góticas se debía a los *compagnons*, que eran colectivos nómadas e itinerantes compuesto por albañiles, carpinteros, herreros, etcétera, que las construían aquí y allá, diseminando las obras, sin división entre trabajo manual e intelectual y con una planificación y construcción descentralizada y autónoma. Hubiese sido entonces más correcto denominar “modelo pirámide” o “modelo rascacielos” al modelo jerárquico y planificado que describe Raymond en su artículo ([www.sindominio.net](http://www.sindominio.net)).

Francois René Rideau en su trabajo “Sobre los artículos de Eric D. Raymond” (1998) explica también que todas las confusiones y parcialidades que aparecen en los artículo de Raymond son típicos de su elección de la *realpolitik*, como principio de actuación en su activismo en pro del *software* libre. Un ejemplo de esta elección es haber cambiado con efectos retroactivos en sus artículos y conferencias el término *software* libre por *open source*.

### **El proyecto Debian**

El proyecto Debian nació bajo los auspicios de la Free Software Foundation (FSF) en 1993, con el objetivo de juntar las piezas de GNU

---

---

y construir un sistema operativo libre completo. Con el tiempo se independizó de la FSF y siguió su propio camino. Debian continúa siendo, sin embargo, un importante sistema y sigue funcionando. Pernees Bruce director del proyecto Debian, no bien asumió la dirección, cuando creó una fundación sin ánimo de lucro: la Software in the Public Interest; se mantiene y desarrolla de manera distribuida mediante la cooperación desinteresada de más de trescientos *hackers* de todo el mundo, y dispone de una comunidad de miles de usuarios a través de más de cincuenta listas de correo públicas extraordinariamente activas. El grupo obtiene muchas donaciones (de gente y de empresas) de *hardware* y ancho de bando. También imprimen un gran número de CD-ROM con el *software*.

### **Retos y desafíos, a modo de conclusión**

Es muy común pensar en que el *software* libre es una utopía. ¿Cómo se puede vivir produciendo *software* libre?, ¿se arrepentirán los programadores en el futuro de darlo gratis?, ¿conseguirá *Linux* competir con *Microsoft*?, ¿cómo hacer para que los grandes productores de *software* no se apoderen de *Linux* y lo privaticen?, ¿serán cada vez más fáciles las herramientas para usar *Linux*, o quedará confinado al uso de programadores y fanáticos de la informática?

Estas y muchas preguntas más, preocupan a la comunidad del código libre. La necesidad de que cada vez haya más códigos modificables se transformará en el futuro en una lucha, no sólo de los programadores pertenecientes al movimiento, sino de todas las personas que al usar computadoras necesiten un lenguaje legible y comprensible para su uso. La comunidad *Linux* está cada vez más extendida por todo el mundo, existen desarrolladores voluntarios en muchos países, sin distinción de raza, credo, ni sexo. “Algunos son más libertarios, otros más comunitarios” (*idem*:191). Como señalamos en el apartado anterior, la mayor distinción puede darse entre grupos que prefieren la licencia GPL y los que usan la licencia estilo BSD para proteger su *software*.

Es cierto que muchos de los usuarios *Linux* pueden no necesitar reinscribir el código y limitarse a ser sólo usuarios pasivos. El código fuente puede convertirse sólo en una ambición para aquellos que

---

---

tienen necesidades y capacidades suficientes para leerlo, estudiarlo y mejorarlo. La mayoría de las personas, sin embargo, sólo quieren que el *software* funcione y no se preocupan demasiado sobre cómo lo hace, cómo se podría acceder al código fuente y cómo podrían adaptarlo a sus necesidades. De aquí que muchos piensen lo difícil que será la continuidad y crecimiento del *software* libre. Otros, sin embargo, continúan siendo optimistas y señalan que los motivos por los que surgió y se desarrolló el *software* libre continúan vigentes.

Ahora bien, es necesario remarcar que las posibilidades para que el *software* libre se transforme en una verdadera alternativa al *software* propietario, pasan por la necesidad de ir resolviendo los problemas que plantea el reproducir algo accesible a la mayoría de los usuarios, quienes necesitan algo menos sofisticado y más práctico.

Otro elemento que es importante señalar, es el tema de los costos. Si bien el *software* libre no es totalmente gratuito, ya que las diferentes empresas que ofrecen soporte Linux cobran algo, los precios por sus servicios son inmensamente más bajos que los de productos comerciales brindados por empresas privadas tipo Microsoft. Los sistemas operativos libres son baratos de ejecutar y, por lo tanto, su distribución también es menos costosa. No sucede lo mismo con las empresas del mercado que, para llegar a pagar todos sus desarrollos, deben gastar fortunas en programadores y en investigadores. Por ello, el tema de los costos se transforma en un elemento fundamental como punto de partida para la competencia.

Otra cuestión interesante para considerar, es el mundo de los usuarios jóvenes. Son ellos lo que dominarán el futuro de las tecnologías. La personalidad de los jóvenes, ansiosos y “sin lealtades previas”, con interés por desentrañar el alma de lo que usan, los sitúa como usuarios preferenciales dentro del movimiento.

Todos estos elementos y muchos más superan, tal vez, el espíritu del movimiento; sin embargo, la idea de compartir el código fuente continuará como el arma más poderosa del movimiento para su sobrevivencia.

Recibido el 29 de agosto de 2002  
Aceptado el 13 de septiembre de 2002

---

## Bibliografía

- Cafassi, Emilio (1998), *Internet: políticas y comunicación*, Editorial Biblos, Buenos Aires.
- Castells, Manuel (1999), *La era de la información*, vols. I, II y III, Siglo XXI Editores, México.
- \_\_\_\_\_(2001), *La Galaxia Internet*, Brosmac, S.L, Madrid, España.
- Crovi, Delia y Girardo Cristina (2001), *La convergencia tecnológica en los escenarios laborales de la juventud*, Hersa Ediciones, UNAM, México.
- Finquilevich, Susana (2000), *¡Ciudadanos a la Red!*, Ediciones La Crujía, Argentina.
- Gambardella, Agata P. (1998), *Costruzione e appropiazione del sapere nei nuovi scenari tecnologici*, CUEN, Nápoles.
- Levy, Pierre (1999), *¿Qué es lo virtual?*, Paidós, Multimedia 10, Barcelona.
- Mochi Alemán, Prudencio (2001), *Las organizaciones de la sociedad civil como actor-red privilegiado y regulador en la globalización*, Cuadernos de Investigación, núm. 14, El Colegio Mexiquense.
- \_\_\_\_\_(2001), *Las organizaciones de la sociedad civil y las ciencias sociales*, Cuadernos de Investigación, núm. 16, El Colegio Mexiquense.
- \_\_\_\_\_(1999), “Netscape to Release New Browser Engine to Developers”, *New York Times* , 7 de diciembre.
- Wayner Peter (2001), *La ofensiva del software libre*, Editorial Granica, Buenos Aires.
- Whitenger , Dave (1999), “Words of a Maddog”, *Linux Today*, 19 de abril, en <<http://linuxtoday.com/stories/5118.html>>.
- Williams, Sam (1999), “Linus Has Left The Building”, *Upside*, 5 de mayo, en <<http://www.upside.com/Open-Season/>>.
- Williams, Riley, “Linux Kernel Vertsion History” en <<http://ps.cus.umist.ac.uk/~rhw/Kernel.versions.html>>.
- Zachary, Pascal (1998), *Showstopper! The Breakneck Race to Create Windows NT and the Next Generation of Microsoft*, Nuovo Gruppo Grafico-Milano.
- Zawinske, Jamie, “Resignation and Postmortem” en <<http://www.jwz.org/gruntle/nomo.html>>.

---

---

*Electrónica*

<http://www.sindominio.net>  
<http://www.open-source.org>  
[www.samizdat.net/slut/textes/mult\\_coopprod.html](http://www.samizdat.net/slut/textes/mult_coopprod.html)  
[www.stallman.org](http://www.stallman.org)  
[www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html)