

# SOFTWARE PÚBLICO, LIBRE, ABIERTO, GRATUITO, PRIVATIVO Y COMERCIAL PARA LA INNOVACIÓN Y MEJORA DE LA CALIDAD EN DOCENCIA, INVESTIGACIÓN, GESTIÓN Y ASISTENCIA

Gabriel Dorado<sup>1</sup>; Juan Antonio Caballero<sup>2</sup>; Jerónimo Ballesteros<sup>3</sup>; Francisco Esteban-Risueño<sup>2</sup>; Antonio García-Ortega<sup>4</sup>; Pilar Hernández<sup>5</sup>; Pilar Lora<sup>6</sup>; María del Pilar Dorado<sup>7</sup>

UNIVERSIDAD DE CÓRDOBA

<sup>1</sup>Profesor Titular de Universidad

<sup>2</sup>Servicio Informática

<sup>3</sup>Departamento Física

<sup>4</sup>Departamento Expresión Gráfica Arquitectónica

<sup>5</sup>Departamento Agronomía y Mejora Genética Vegetal

<sup>6</sup>Departamento de Enfermería

<sup>7</sup>Departamento Química Física y Termodinámica Aplicada

“La revolución de las nuevas tecnologías en la sociedad de la información y el conocimiento está siendo posible gracias al software en sus diversas formas (público, libre, abierto, gratuito, privativo y comercial)”

ESTUDIOS DE CALIDAD E INNOVACIÓN DE LA UNIVERSIDAD DE CÓRDOBA-AÑO 4º

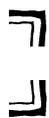
---

---

RES NOVAE IV  
CORDUBENSES



ÍNDICE



**1. INTRODUCCIÓN**

**2. TIPOS DE SOFTWARE**

**3. VENTAJAS E INCONVENIENTES DEL SOFTWARE ABIERTO**

**4. VENTAJAS E INCONVENIENTES DEL SOFTWARE GRATUITO**

**5. VENTAJAS E INCONVENIENTES DEL SOFTWARE PRIVATIVO**

**6. REFLEXIONES SOBRE EL SOFTWARE ABIERTO, GRATUITO Y PRIVATIVO**

**7. CONCLUSIONES**

**8. AGRADECIMIENTOS**

**9. REFERENCIAS**

---





TEXTO



**SOFTWARE PÚBLICO, LIBRE, ABIERTO, GRATUITO, PRIVATIVO Y COMERCIAL PARA LA INNOVACIÓN Y MEJORA DE LA CALIDAD EN DOCENCIA, INVESTIGACIÓN, GESTIÓN Y ASISTENCIA**

Gabriel Dorado ([bb1dopeg@uco.es](mailto:bb1dopeg@uco.es))

Juan Antonio Caballero

Jerónimo Ballesteros

Francisco Esteban-Risueño

Antonio García-Ortega

Pilar Hernández

Pilar Lora

María del Pilar Dorado

**RESUMEN**

El software público libre es el fundamento del movimiento de software de fuente abierta (OSS). Éste puede facilitar el uso de estándares abiertos, con accesibilidad independiente de plataforma informática. El bajo coste o gratuidad de muchos productos OSS permite su implantación cuando los recursos económicos son escasos (países en desarrollo, escuelas, etc.). Además, su código se puede modificar, adaptar y mejorar por cualquiera, ya que es público. No obstante, el OSS también puede presentar inconvenientes. Así, su desarrollo puede no ser tan ordenado y jerárquico como el del software privativo y puede no tener una financiación apropiada. De este modo, puede verse comprometida la facilidad de uso y coherencia del producto final. Ello puede incrementar los costes de transición, entrenamiento y soporte técnico. Por otra parte, conviene recordar que existe software libre comercial y software privativo gratuito. En cualquier caso, la excelencia del software puede ser independiente de dichas variables. Por tanto, la mejor estrategia es la coexistencia y libre competencia de todo tipo de software. Así se puede conseguir una mejora de calidad y una potenciación de la innovación, que beneficie a la docencia, investigación, gestión y asistencia universitarias en particular, así como a la sociedad en general

**PALABRAS CLAVE :** Unix, Darwin, Mac OS X, Linux, HispaLinux, GuadaLinux, Windows, forja.

**ABSTRACT**

The public free software (now renamed as open software) is the basis of the open source software (OSS) project. This initiative promotes the use of open standards and platform-independent accessibility. Many OSS products are free or low cost, which allows their deployment even in the absence of large economic resources (developing countries, schools, etc.). Besides, their source code can be modified, adapted and optimized by anyone, since it is public. Nevertheless, the OSS may also have drawbacks. Thus, its development may not be as orderly and hierarchical as the privative software is, and may not have an appropriate funding. This may jeopardize the ease of use and coherence of the final product, increasing the transition, training and technical support costs. Besides, it must be taken into account that there is also commercial open software, as well as free privative software. In any case, the excellence of the software may be independent of such variables. Thus, the best strategy is the coexistence and free competition of all kinds of software. This will allow a quality improvement and an innovation enhancement, which will benefit the education, research, management and assistance at the university in particular and at the whole society in general.



**KEYWORDS :** Unix, Darwin, Mac OS X, Linux, HispaLinux, GuadaLinux, Windows, forge.

## 1. INTRODUCCIÓN

Vivimos en la denominada Sociedad de la Información, pero puede darse la paradoja de que no exista un acceso libre a dicho conocimiento (Dorado et al, 2005). Por un lado, pueden plantearse problemas de accesibilidad derivados de minusvalías. Por otro, debe tenerse en cuenta el acceso a la información independientemente de la plataforma informática usada. Además, existen herramientas como el software, que pueden ser necesarias para acceder a la información y al conocimiento mediante una enseñanza de calidad. No obstante, dichas herramientas pueden no ser de libre acceso y gratuitas. Puede crearse así una brecha digital, que suele afectar a los más desfavorecidos. ¿Representa el software gratuito y libre de código abierto (Feller et al, 2005) una solución a estos problemas?

La idea del software libre de código abierto es cuando menos atractiva (Golden, 2004). Para una minoría, el software público libre o GPL (del inglés, “General Public License”) representa un movimiento ideológico y sociológico que cambiará el mundo. Para la gran mayoría, es una forma de conseguir programas sin tener que pagarlos. Pero el software libre es un concepto complejo que precisa un tratamiento en detalle.

Por una parte conviene aclarar que software libre u OSS (“Open Source Software”) no tiene que ser necesariamente sinónimo de software gratuito (“freeware”). Éste es quizá el error más extendido en relación al concepto de OSS. Así, la licencia GPL no prohíbe la venta del software creado; sólo exige que el código fuente (“source code”) sea público y esté disponible de forma gratuita. Como el software libre de código abierto no tiene nada que ver con el precio, un precio bajo no indica que el programa sea más “libre” o esté más cerca de serlo. Redistribuir software libre de código abierto y cobrar por ello es por tanto una actividad comercial como cualquier otra (Stallman, 2005) <<http://www.gnu.org/philosophy/selling.html>>. De modo que existe software libre gratuito, pero también existe otro que se vende (generalmente, con algún valor añadido). Por este motivo quizá sea mejor llamarlo software abierto (“open source”), porque la palabra libre sugiere gratuidad.

Por otra parte, no todo el software privativo comercial (propietario) se vende; también existe software privativo –y por tanto no abierto– que es gratuito. Un ejemplo de ello es la inmensa mayoría de miniaplicaciones (“Widgets”) <<http://www.apple.com/es/downloads/dashboard>> del Tablero de Instrumentos (“Dashboard”) <<http://www.apple.com/es/macosx/features/dashboard>> de Mac OS X (Pogue, 2005) <<http://www.apple.com/es/macosx>>. Asimismo, aplicaciones como iTunes <<http://www.apple.com/es/itunes>>, QuickTime <<http://www.apple.com/es/quicktime/mac.html>> o iMovie <<http://www.apple.com/ilife/imovie>>.

Otro ejemplo son los navegadores Web, que pueden ser gratuitos de código abierto, así como privativos.

Entre ellos se encuentran Safari <<http://www.apple.com/es/macosex/features/safari>>, Firefox <<http://www.mozilla.org/products/firefox>>, Mozilla <<http://www.mozilla.org/products/mozilla1.x>>, Netscape <<http://browser.netscape.com/ns8/download/archive72x.jsp>>, Opera <<http://www.opera.com/products/desktop>> o Internet Explorer <<http://www.microsoft.com/mac/products/internetexplorer/internetexplorer.aspx?pid=internetexplorer>>. Por tanto, existen prácticamente todas las combinaciones que se puedan considerar.

En este trabajo se comentan y discuten las diferentes posibilidades y variantes del software para conseguir una mejora de calidad y una potenciación de la innovación, que beneficie a la docencia, investigación, gestión y asistencia universitarias en particular, así como a la sociedad en general.

## 2. TIPOS DE SOFTWARE

El software privativo es cerrado y, como se ha indicado, puede ser comercial o gratuito. En el caso del OSS existen más posibilidades. Aunque las clasificaciones suelen ser artificiales e incorrectas hasta cierto punto, también sirven para entender el mundo en que vivimos. Así, el OSS se puede clasificar en varios apartados, según se indica a continuación:

- OSS jovial. Es aquel realizado por diversión en el tiempo libre del programador. No obstante, la capacidad de programación puede variar mucho de unas personas a otras, y en cualquier caso, estos programadores deberán tener también un trabajo remunerado para poder vivir dignamente.
- OSS forofo. Se trata de una minoría de programadores que producen este tipo de software porque lo asumen como un ideal religioso, social o de identidad. O bien porque tratan de luchar así contra la tiranía de empresas monopolio como Microsoft. A veces estos programadores producen software muy bueno, dada su genialidad, dedicación y devoción a la causa. La mayoría de estos programadores geniales acaban siendo contratados por empresas. De hecho, ésta es también una estrategia para conseguir un buen trabajo de programador en el mercado informático. Un ejemplo de OSS forofo es Asterisk <<http://www.asterisk.org>>, que permiten contar con centrales telefónicas de software abierto, según se indica más adelante.
- OSS educativo. Se trata de programas escritos por profesores y estudiantes como tarea docente, para enseñar y aprender a programar. Aunque las habilidades de los implicados pueden ser intermedias, se pueden generar resultados muy positivos. De hecho, el propio kernel de Linux (Raymond, 2001; Weber,



2004; Hill et al, 2005; Negus, 2005) <<http://www.linux.org>> fue creado originalmente de este modo por Linus Torvalds. Se han publicado diversos catálogos de software de código abierto para su uso en educación, como Seul/Edu (“Educational Application Index”) <<http://richtech.ca/seul/index.php>>, que está clasificado por categorías. Otro catálogo de software educativo libre se puede consultar en Freeduc (“Organization for Free Software in Education and Teaching”) <<http://www.ofset.org/freeduc>>.

- OSS empresarial. Aunque parezca paradójico, se puede ganar dinero produciendo software abierto, libre y gratuito. Así, el software de Servidor Web Apache <<http://www.apache.org>> es producido por profesionales que trabajan en empresas que luego usan dicho software de forma gratuita. No obstante, para que estos proyectos tengan éxito es necesario que las empresas interesadas los financien con el tiempo y esfuerzo de sus empleados (que dejarán de realizar otras tareas). Otro ejemplo de software abierto y gratuito que genera beneficios son los controladores (“drivers”) de diversos periféricos (cámaras fotográficas digitales, etc.), ya que ello favorece la compra de dichos periféricos; aparte de todas las evaluaciones (control de calidad) que llevan a cabo innumerables usuarios, de forma altruista. De forma similar, el gestor de contenidos Zope <<http://www.zope.org>> es gratuito, pero resulta rentable económicamente de forma indirecta. Así, la empresa que produce y distribuye dicho software genera beneficios al vender servicios asociados al mismo, como son las instalaciones a medida, soporte, formación, optimización, etc.

También son interesantes proyectos como Morfeo (Comunidad de Software Libre de Telefónica), que tiene como objetivo proporcionar un marco abierto de colaboración entre instituciones de la Administración, grandes empresas, PYMEs, universidades y Telefónica I+D para llevar a cabo proyectos conjuntos de investigación, desarrollo e innovación (I+D+i) relacionados con el desarrollo de tecnologías OSS (Hierro, 2005).

Por otro lado, la informática moderna es la informática con interfaz gráfica intuitiva y fácil de usar por cualquier usuario. Es lo que se denomina el escritorio del ordenador (“desktop”) y cuyo líder y ejemplo emblemático es Mac OS X (Pogue, 2005) <<http://www.apple.com/macosx>>, que es un software privativo comercial. En el caso del OSS, caben destacar los desarrollos de la empresa Canonical para dotar de un escritorio similar a Linux <<http://www.canonical.com>>.



- OSS estatal. Se trata de software abierto financiado con el dinero público de los contribuyentes. Ésta es una iniciativa que cada vez está teniendo más auge. Así, en España existen distribuciones (“distributions”) de Linux <<http://www.hispalinux.es>> como, por ejemplo, las siguientes:

- ◆ LinEx <<http://www.linex.org>>. Está basada en Debian (Hill et al, 2005) y patrocinada por la Junta de Extremadura, habiendo tenido una gran repercusión en los medios de comunicación internacionales tras su publicación en la primavera de 2002.
- ◆ GuadaLinex <<http://www.guadalinux.org>>. Es una iniciativa de la Junta de Andalucía, inicialmente derivada de Linex y actualmente basada en Ubuntu <<http://www.ubuntu.com>>. GuadaLinex se publicó en el verano del 2003, estando su desarrollo abierto a cualquiera desde el verano de 2005, mediante la creación de una forja (G-Forja) <<http://forja.guadalinux.org>>. La forja es una web de desarrollo de software abierto en la que pueden aglutinarse personas, grupos, empresas y organismos públicos (como las universidades).
- ◆ Molinux <<http://www.molinux.info>>. Está patrocinada por la Junta de Castilla-La Mancha. La última versión (Sancho) está basada en “Ubuntu Breezy Badger” <<http://www.ubuntu.com>>.
- ◆ MAX (MAdrid\_LinuX) <[http://www.educa.madrid.org/web/madrid\\_linux/index.html](http://www.educa.madrid.org/web/madrid_linux/index.html)>. Está patrocinada por la Comunidad de Madrid.

Existen otros proyectos estatales de países como China, India, Rusia, Polonia y Méjico. Todo esto es muy bueno, porque permite que regiones o países menos desarrollados puedan realizar contribuciones al desarrollo tecnológico, en vez de dedicarse a copiar sin permiso un software que posiblemente no podrían comprar. Por su parte, el gobierno de los Estados Unidos de América (EUA) está financiando el desarrollo de OSS realizado en universidades (Stanford) y empresas (Symantec y Coverity) (Muncaster, 2006). Todo ello sugiere que el OSS tiene un gran futuro por delante.

A continuación se discuten algunos detalles sobre las ventajas e inconvenientes de diferentes tipos de software: abierto, gratuito y privativo.

### 3. VENTAJAS E INCONVENIENTES DEL SOFTWARE ABIERTO

Un ejemplo emblemático de OSS es Linux, que puede adquirirse de forma gratuita o pagando un cierto precio por algún “sabor” (“flavor”) o distribución de mismo que le da un valor añadido, según se indica más abajo.



---

A priori, el sistema de OSS tiene grandes ventajas, ya que otros pueden analizarlo, modificarlo, adaptarlo,

actualizarlo y mejorarlo. De este modo, todos se benefician porque el código fuente es gratuito y de libre disposición (abierto). Esta alternativa de software puede ser fantástica en la enseñanza, ya que los estudiantes pueden analizar y modificar el código fuente, comprobar cómo funciona y adaptarlo a las necesidades concretas del curso. En teoría, una aplicación basada en OSS nunca “muere”, ya que al estar el código fuente disponible, cualquiera puede seguir desarrollándola, incluso cuando los autores iniciales no lo hagan. Al poder tener muchos programadores o desarrolladores (“developers”) trabajando en el código fuente, éste puede mejorar notablemente (Raymond, 2001; Weber, 2004; Hill et al, 2005; Negus, 2005). Así, si cualquiera detecta un problema o error en dicho código, puede arreglarlo sin tener que esperar a que lo haga el autor original. Por otro lado, el OSS permite controlar y evitar la existencia de “puertas traseras” o código inesperado (que atente contra la privacidad), así como el uso malicioso de un programa o sistema operativo dado.

Pero el sistema de OSS también puede tener inconvenientes. El factor más importante es el económico (Welch, 2003). Como se ha indicado, software libre (abierto) no implica necesariamente que se trate de software gratuito. Además, puede existir un coste de entrenamiento, transición y soporte técnico. De hecho, programar es un trabajo arduo y hacerlo de forma genial y exquisita está reservado a unos pocos. No es por tanto realista esperar que todo el software que se necesite o que esté disponible en el mercado sea de código abierto o que sea producido por alguien gratuitamente y “por amor al arte”. Programar es un trabajo, como lo es el desarrollado por escritores, músicos, ingenieros o médicos, por citar algunos ejemplos.

El código abierto puede significar mayor seguridad, como se ha indicado, pero también puede haber una mayor vulnerabilidad, en el caso de que programadores maliciosos aprovechen posibles errores de seguridad del código antes de que sean corregidos. Por otro lado, en general, no existe un liderazgo claro en el desarrollo del OSS, lo cual puede ser contraproducente.

Por todo ello, puede producirse la paradoja de que la contratación del desarrollo de OSS para fines concretos puede ser más caro que el desarrollo de software privativo comercial. Ello se puede deber a que la empresa encargada de dicho desarrollo no pueda vender su producto a terceros y además no pueda garantizar necesariamente el desarrollo de actualizaciones o dar soporte técnico. Así pues, al no obtener otros beneficios de terceras partes, el costo total del desarrollo del OSS se puede incrementar para la parte contratante (como pueda ser la Administración). No debe olvidarse que, en general, al final alguien debe pagar dicho desarrollo.

#### 4. VENTAJAS E INCONVENIENTES DEL SOFTWARE GRATUITO

No cabe duda que la mayor ventaja del software gratuito es precisamente que no hay que pagar por su uso. Pero, como se ha indicado, el software gratuito no es necesariamente software abierto, sino que también puede ser privativo.

Uno de los ejemplos más populares de código fuente abierto y gratuito es el sistema operativo Linux, por lo que está libremente disponible para su descarga y uso. Linux es un sistema operativo similar a Unix (Lepage e Iarrera, 2000) <<http://www.unix.org>>, aunque desarrollado de forma independiente mediante software abierto. Linux fue creado originalmente por Linus Torvalds con la ayuda de programadores de todo el mundo. Es desarrollado de acuerdo con la licencia pública general del GNU (acrónimo repetitivo de “GNU's Not Unix”) <<http://www.gnu.org>> (Hill et al, 2005). Es decir, estrictamente hablando, Linux no deriva del tronco Unix, pero es como Unix (“Unix-like”) <<http://en.wikipedia.org/wiki/Linux>>.

Por tanto, esta variante o “sabor” de Unix –o “como Unix”– llamado Linux puede considerarse como “Patrimonio de la Humanidad”. No obstante, es preciso realizar unas aclaraciones. Cuando se habla de gratuidad del software abierto como Linux, ésta se refiere al código base o fuente, que es necesario compilar. Además, no se debe olvidar la interfaz gráfica que usan actualmente la mayoría de los usuarios de informática. Dicho con otras palabras, la informática de comandos es ya historia desde el punto de vista del usuario general no informático. Los usuarios de ordenadores quieren y piden sistemas operativos y aplicaciones (programas) con una interfaz gráfica de usuario o GUI (“Graphical User Interface”) agradable, y que sea sobre todo fácil de usar y por tanto intuitiva. Por ello han surgido numerosas distribuciones de Linux que no son gratuitas y que se diferencian entre sí principalmente por su interfaz gráfica de usuario. Entre ellas se encuentran los siguientes “sabores” de Linux: LinuxPPC, Yellow Dog Linux, Debian GNU/Linux, Mandrake Linux, Familiar Linux, YOPY/Linupy, SuSE Linux, Fedora Core, Red Hat Linux, etc.

Por tanto, lo que aparentemente es un software gratuito, puede no serlo totalmente. Por otro lado, la principal desventaja del software gratuito puede ser la ausencia de desarrollo. Así, hay pocas aplicaciones de productividad (como las de Adobe) para Linux (Hill et al, 2005; Negus, 2005), mientras que dichas aplicaciones existen para otros sistemas operativos privativos comerciales como Mac OS X (Pogue, 2005) y Windows (Russel y Crawford, 2005). Por razones obvias, el software gratuito no suele tener, en general, la calidad del comercial (sea éste abierto o privativo).



## 5. VENTAJAS E INCONVENIENTES DEL SOFTWARE PRIVATIVO

El software privativo no suele ser gratuito sino comercial y –como su nombre indica– es cerrado (no abierto). Todas esas características pueden ser inconvenientes de este tipo de programas; sobre todo si el precio es elevado. No obstante, como se ha comentado, existe software privativo gratuito; es el llamado software de prueba (“tryware”) y las versiones de demostración (“demo”), limitadas de alguna forma (uso restringido de funciones) o totalmente funcionales durante un periodo de tiempo limitado. El objetivo de estos últimos es que se pueda probar la aplicación antes de decidirse a comprarla.

Probablemente la mayor ventaja del software privativo es que en general es de calidad, aunque también existen casos de software privativo comercial mediocre e incluso pésimo, que a veces se mantiene por la inercia del mercado y la posición dominante y monopolista de algunas empresas. Esta situación sorprendente de prevalencia de un software mediocre también puede producirse con el software abierto y gratuito. Basta comparar la funcionalidad de Mac OS X, Linux y Windows para sacar las conclusiones apropiadas al respecto. Igual podría decirse cuando se compara la calidad de iLife (iPhoto, iMovie HD, iDVD, iTunes y GarageBand) <<http://www.apple.com/ilife>> con los productos competidores gratuitos, siendo evidente la elevada calidad y facilidad de uso de los primeros.

## 6. REFLEXIONES SOBRE EL SOFTWARE ABIERTO, GRATUITO Y PRIVATIVO

Hay algo fantásticamente altruista y bueno en los proyectos de OSS. Pero también hay algo que puede no encajar: la economía (Welch, 2003). Está claro que hoy por hoy los seres humanos necesitan ganar dinero para mantenerse ellos y sus familias. El software abierto y gratuito es una gran idea y tiene su lugar. Pero el software privativo y comercial también es una gran idea que debe tener su lugar, como lo tienen las obras maestras del arte, la literatura, la tecnología y la ciencia. Es bueno que algunas cosas sean gratis, pero todo no puede ser gratis. Al menos, por ahora, no sólo no puede ser así, sino que sería negativo y contraproducente para la creatividad, el desarrollo y la innovación.



Es también necesario que las empresas ganen dinero para que puedan invertir parte del mismo en I+D+i. De hecho, tanto la comunidad de OSS como las empresas de software privativo pueden beneficiarse mutuamente y complementarse. Todo ello puede crear también un ambiente legítimo de competitividad creativa y productiva. La idea de la complementariedad entre los distintos tipos de software para generar innovación y potenciar la

creatividad no es una utopía. Como ejemplo de la interacción entre OSS y software privativo comercial se puede considerar a Darwin <<http://www.apple.com/es/macosx/features/unix>>, que es de código abierto <<http://developer.apple.com/darwin>> y el fundamento Unix (Lepage e Iarrera, 2000) <<http://www.apple.com/es/macosx/features/unix>> basado en FreeBSD <<http://www.freebsd.org>> y Mach <[http://www.cs.cmu.edu/afs/cs/project/mach/public/www/sources/sources\\_top.html](http://www.cs.cmu.edu/afs/cs/project/mach/public/www/sources/sources_top.html)> sobre el que se asienta la interfaz gráfica más avanzada e intuitiva que existe actualmente: Aqua <<http://www.apple.com/es/macosx/overview/aquauserinterface.html>> de Mac OS X <<http://www.apple.com/es/macosx>>, que son desarrollos privativos comerciales.

Por tanto, el software gratuito es una excelente idea como una opción complementaria más en el conjunto del software. El OSS perdería su sentido si se considera que su objetivo es sólo y exclusivamente producir software gratuito para no tener que pagar por algo que cuesta esfuerzo hacer (programar no es fácil). A menos que estemos pensando que el Estado se encargue de ello (es decir, que se financie con los impuestos de todos). En este sentido, la historia muestra que, en general, ésta no es precisamente la mejor opción para la creatividad, innovación y desarrollo. El viejo dilema entre sociedades comunistas y capitalistas así lo ha demostrado. Por suerte o por desgracia, las utopías no siempre casan con la naturaleza humana. Por eso el mundo es como es, aunque no cabe duda que podemos intentar mejorarlo. La frase “el dinero hace girar al mundo” tiene una relevancia nada desdeñable.

Precisamente por todo ello existe el derecho a la propiedad, tanto privada como pública, incluyendo marcas registradas y patentes. De este modo se pueden proteger los derechos de autor y se incentiva así el trabajo de calidad y la innovación. Está sobradamente demostrado que cuando existe un incentivo económico es cuando se consiguen productos de calidad de forma más rápida y eficiente. Esto no quiere decir que todo tenga que estar protegido y patentado, pero sí que estas protecciones también tienen su lugar en nuestra sociedad, como lo debe tener el sistema OSS y lo que puede implicar (gratuidad, software libre y abierto, etc.). Por tanto, en el equilibrio puede estar la virtud: coexistencia de distintos tipos de software (abierto, gratuito, privativo y comercial). Se deben potenciar todos y que la competencia y el mercado hagan su tarea de seleccionar lo mejor, dentro de unas reglas de coexistencia.

Quizá la prueba más clara e irrefutable de que una cosa son los deseos y otra la realidad es que a pesar de existir software y en particular sistemas operativos de tipo OSS, no han podido eliminar a los competidores privativos comerciales. Es cierto que en general existe una inercia al cambio, pero siendo los primeros gratis, esta situación es reveladora. Por otro lado, curiosamente, la piratería juega un papel en contra del software gratuito. Así, este último no sólo compite contra el privativo comercial legal en calidad/precio, sino también contra el software comercial pirata, donde la competencia es exclusivamente calidad/calidad (si excluimos las consi-



deraciones legales).

La explicación última de esta realidad es que también pueden existir problemas asociados al OSS. La historia del desarrollo de software indica que algunos tipos de programas se desarrollan mejor como OSS (los de bajo nivel e interfaz de comandos), pero otros lo hacen mejor como software comercial (los que requieren una interfaz de usuario más refinada e intuitiva). Por ello, los sistemas operativos Linux y BSD <<http://www.bsd.org>> se han desarrollado muy bien como OSS, pero justo lo contrario ha sucedido con los escritorios de ordenador (como la interfaz Aqua de Mac OS X) y con los juegos de ordenador.

Cualquiera que haya usado por ejemplo Linux (abierto y gratuito) y Mac OS X (privativo y comercial) entenderá la situación. Ambos no son comparables entre sí en términos de facilidad de instalación, configuración, administración, uso y funcionalidad. Desde luego no son comparables para un usuario medio, conocedor de la “informática de usuario”, pero sin conocimientos avanzados en “informática de código” (comandos, programación, etc.). Así, aunque en Mac OS X existe una interfaz de comandos Unix (Lepage e Iarrera, 2000) llamada Terminal (Pogue, 2005), la inmensa mayoría de los usuarios de dicho sistema operativo nunca la han necesitado y probablemente nunca sepan que existe, pues difícilmente precisarán manejarla. En Linux no se ha llegado –al menos por el momento– a tal nivel de facilidad de uso, aunque dicho objetivo es un reto en el mundo del OSS. Puede que algún día la interfaz gráfica de usuario de Linux sea tan intuitiva como la de Mac OS X, pero hasta entonces las diferencias son obvias y evidentes, con Windows (Russel y Crawford, 2005) a mitad de camino en facilidad de uso. En definitiva, el objetivo debe ser hacer software de alta calidad que sea intuitivamente fácil de usar. Actualmente existen diferencias abismales entre programas y entre sistemas operativos a este respecto. Todo ello no tiene que ser necesariamente negativo; al contrario: éste puede ser otro ejemplo de complementariedad, sana competencia y creatividad entre el software abierto, gratuito, privativo y comercial. Todo ello puede redundar en la creación de software de mayor calidad y por tanto en un beneficio del usuario final.

Existe también otro aspecto positivo del OSS que es limitar el posible precio abusivo que podrían tener algunos programas comerciales de no existir alternativas abiertas y gratuitas (o más baratas). En este sentido, el OSS actúa como un amortiguador. Además, puede permitir la innovación y el desarrollo futuro de otros OSS o de software comercial con más valor añadido.



Como se ha comentado, el principal problema del software abierto puede ser que, si no existe una fuente directa de financiación por la venta del mismo, se pueden resentir aspectos de control de calidad, interfaz intuitiva y desarrollo. Por eso en algunos casos el software abierto suele adornarse con alguna ventaja añadida, sien-

do entonces vendido a un determinado precio, que no suele ser elevado. Debe recordarse que, como dice el aforismo, “nada es gratis”. Alguien habrá tenido que pagar –aunque sea con su tiempo– el esfuerzo de desarrollar algo. Y tampoco debemos ser egoístas, pretendiendo que los programadores trabajen para nosotros y nos den siempre el software gratis (los programadores no comen “código”).

Por todo ello, algunos proyectos de OSS tienen problemas para llegar a la versión 1 como software abierto y gratuito. De modo que los que utilizan dichas versiones previas pueden actuar como equipo de pruebas gratuito (probadores beta), en vez de como auténticos usuarios finales. Se llega a dar incluso el caso de que sólo las versiones 0.x sean gratuitas u OSS, mientras que las versiones 1.x y posteriores sean privativas o comerciales. Como ejemplos se puede citar a PocketWeather (gratuito hasta la 1.0) <<http://www.sbsb.net>> y GateD (OSS hasta la 5.0) <<http://www.nextop.com>>.

No obstante, las nuevas estructuras de creación de software libre colaborativas pueden ser vehículos capaces de superar el antagonismo económico privativo-comercial versus abierto-gratuito. En España existen dos ejemplos de cada uno: G-Forja y Morfeo, comentados anteriormente. Por otro lado, iniciativas como Asterisk (comentada anteriormente) demuestran que el movimiento OSS está saliendo del ámbito puramente informático para alcanzar otros campos como el de las telecomunicaciones telefónicas tradicionales, en los que hasta ahora sólo había software privativo (generalmente incompatible entre los distintos fabricantes y con un precio muy elevado).

## 7. CONCLUSIONES

El corolario y conclusión de este trabajo es que la coexistencia de los distintos tipos de software (abierto, gratuito, privativo y comercial) es interesante y puede facilitar el desarrollo de nuevos productos de calidad. Ninguno de ellos representa una panacea en sí mismo, ni existe de forma independiente a las leyes que rigen la economía.

En el mundo del software, en el equilibrio está la virtud. De este modo puede potenciarse la creatividad y el desarrollo en la sociedad del conocimiento. Ello favorece la innovación en educación, investigación, gestión y asistencia en el entorno universitario. El beneficiario de esta convivencia y libre competencia será toda la sociedad en general y el usuario final que se sienta en frente de la pantalla de un ordenador, en particular. No hay nada como la competencia y la libertad de mercado para que la tecnología avance y los productos mejoren.



Así, por ejemplo, un programa de código abierto puede resultar conveniente cuando interese modificarlo y adaptarlo a unas necesidades determinadas, como sucede con el sistema operativo Linux. Esto es particularmente relevante en grandes empresas, instituciones o en la Administración regional o estatal. En otras situaciones lo ideal puede ser un programa gratuito, porque no existan recursos para adquirirlo, como puede suceder en una escuela o en países en desarrollo. Pero en otros casos lo ideal puede ser un programa o sistema operativo comercial, porque ofrezca un valor añadido extraordinario. Tal es el caso del sistema operativo Mac OS X con su interfaz Aqua, o de paquetes como Final Cut Studio (Final Cut Pro, Soundtrack Pro, Motion y DVD Studio Pro) <<http://www.apple.com/finalcutstudio>> para la edición de películas profesionales, cuya calidad lo ha erigido en la herramienta preferida en los estudios cinematográficos de Hollywood.

## 8. AGRADECIMIENTOS

Financiado por Grupo PAFPU 'FORMAPROFE' ('UCO N 031') y Grupo PAFPU 'FORMAPRO' ('UJA N 025') de Formación del Profesorado Universitario, Junta de Andalucía (Spain).

## 9. REFERENCIAS

DORADO G., CABALLERO J.A., BALLESTEROS J., MELÉNDEZ L., ESTEBAN-RISUEÑO F., MARTÍNEZ A., DE JUAN F., DE CASTRO C., GARCÍA-ORTEGA A., HERNÁNDEZ P., LORA P., DORADO M.P. (2005): Accesibilidad a Internet: requisitos para la calidad en docencia e investigación. *Res Novae Cordubenses (Estudios de Calidad e Innovación de la Universidad de Córdoba)* 3: 119-142.

FELLER J., FITZGERALD B., HISSAM S.A., LAKHANI K.R. (Eds) (2005): *Perspectives on Free and Open Source Software*. The MIT Press, Cambridge, MA, USA.

GOLDEN B. (2004): *Succeeding with Open Source*. Addison-Wesley Professional, Boston, MA, USA.

HIERRO J.J. (2005): Telefónica I+D y el software libre: experiencias y creación de la comunidad MORFEO <<http://www.mityc.es/NR/rdonlyres/C3668503-C36D-4A0F-AE88-9A3D08AE5D1A/0/SwLibre20050309TID.pdf>>.

HILL B.M., HARRIS D.B., VYAS J. (2005): *Debian GNU /Linux 3.1 Bible*. Wiley ed., Hoboken, NJ, USA.

LEPAGE Y., PAUL IARRERA P. (2000): *UNIX Bible*. 2nd ed. Wiley ed., Hoboken, NJ, USA.

MUNCASTER P. (2006): US pays to make open source safer. *IT Week* (16 January) <<http://www.vnunet.com/itweek/news/2148598/pays-open-source-safer>>.



NEGUS C. (2005): *Linux Bible*. 2nd ed. Wiley ed., Hoboken, NJ, USA.

105

POGUE D. (2005): *Mac OS X: The Missing Manual, Tiger Edition*. O'Reilly ed., Sebastopol, CA, USA.

RAYMOND E.S. (2001): *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly, CA, USA.

RUSSEL C., CRAWFORD S., Microsoft Windows Team (Eds) (2005): Microsoft Windows XP Professional Resource Kit. 3rd ed., Microsoft Press, Redmond, WA, USA.

STALLMAN R. (2005): Selling Free Software <<http://www.gnu.org/philosophy/selling.html>>.

WEBER S. (2004): The Success of Open Source. Harvard University Press, Cambridge, MA, USA.

WELCH A. (2003): Ambrosia Times: President's Letter: Open Source Software economics <<http://www.ambrosiasw.com/news/newsletter.php?show=6&start=0>> y <<http://www.ambrosiasw.com/forums/index.php?act=ST&f=26&t=34060>>.

