

Software Libre y sus posibilidades en la educación

JULIO CABERO ALMENARA
M.C. LLORENTE CEJUDO
(Universidad de Sevilla)

Los inicios del software libre debemos buscarlo en la década de los sesenta con los comienzos de la programación informática, y el intercambio de información entre los programadores del código con el cual se estaba trabajando, pero a comienzo de los años ochenta ese modelo entra en crisis, surgiendo un modelo privatizador y mercantilista, de venta e investigación de la producción informática producida, hecho que lleva a la cristalización del movimiento del software libre y en el cual podemos distinguir una serie de acontecimientos claves, como nos señala Más i Hernández (2005, 177-178):

- La creación del proyecto GNU is Not Unix (GNU), liderado por Richard Stallman.
- La constitución de la Free Software Foundation (FSF).
- La publicación de la primera versión de la GNU Public License (GPL).
- El desarrollo de BSD UNIX por parte de la Universidad de California en Berkeley.
- Y la libre circulación e intercambio del software a través de Internet.

El movimiento ha alcanzado una gran extensión, en la década de los noventa y los comienzos del nuevo siglo, con el respaldo de diferentes países como Venezuela y Brasil, o regiones de España como Extremadura y Andalucía. Este impulso ha sido potenciado por la presencia de la red, con sus posibilidades de comunicación, sincrónica y asincrónica, y la potenciación del trabajo colaborativo.

Un software lo podemos considerar como libre cuando los usuarios pueden ejecutarlo, copiarlo, distribuirlo, estudiarlo, cambiarlo y mejorarlo. O dicho en otros términos cuando cumple con cuatro libertades:

Libertad 0: la libertad para ejecutar el programa sea cual sea nuestro propósito.

Libertad 1: la libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades —el acceso al código fuente es condición indispensable para esto.

Libertad 2: la libertad para redistribuir copias y ayudar así a tu vecino.

Libertad 3: la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad —el acceso al código fuente es condición indispensable para esto. (Richard Stallman, 2004, 45). Respecto al software propietario tiene las diferencias que presentamos en el cuadro nº 1.

ASPECTO A CONSIDERAR	SOFTWARE PROPIETARIO	SOFTWARE LIBRE (GPL)
Acceso al código fuente	Prohibido por licencia	Si, Garantizado
Corrección de errores en el programa por el cliente/ usuario	No	Si
Duplicación del software	Prohibido	Posible y Recomendada
Libertad de Competencia para el mantenimiento	No, depende del fabricante	Si imposible limitarla
Posibilidad de examinar el código del producto	Prohibido, salvo permiso fabricante.	Si
Venta de segunda mano	Prohibido	N/A
Respeto a estándares globales	En función del fabricante	En la mayoría de los casos
Adaptaciones al cliente (P.e. Idioma)	En función del fabricante	Disponible
Virus, gusanos	Frecuentes	Muy Infrecuentes

Cuadro nº 1. Diferencias entre el software libre y propietario (Abella y otros, 2003, 31).



REDUCCIÓN DE COSTES ECONÓMICOS

No debe confundirse con que el software sea gratuito, pero si conlleva un ahorro de programas y de licencias. En su producción su coste es menor ya que se reducen costes iniciales, o se trabaja sobre versiones realizadas por otras personas, en consecuencia ya existe un trabajo previo, sólo deben pagarse los servicios de su implantación y adaptación, los cuales son siempre menores, pues las empresas que trabajan con software libre se encuentran en competencia real, y no determinada por las casas comerciales.



DISTRIBUIR COPIAS LIBREMENTE

Lo cual presenta, además de la ventaja de la reducción del coste económico, en el caso de la enseñanza, la añadida de que los alumnos pueden tener en sus domicilios los mismos programas que sus profesores, facilitando con ello el aprendizaje y evitando posibles discriminaciones entre los alumnos, que no puedan acceder a los programas de los profesores por motivos económicos.



INTERCAMBIO DE DOCUMENTOS

Favorece el intercambio de documentos electrónicos entre las diferentes administraciones e instituciones.



DESARROLLO TECNOLÓGICO

Facilita el desarrollo tecnológico del software ya que existe un gran número de personas dispuestas a potenciar la evolución de los programas, independientemente del mercado y los intereses de las empresas.



MENOS CAPACIDAD HARDWARE



FAVORECE LA INVESTIGACIÓN EN LOS ENTORNOS EDUCATIVOS

Favorece la investigación en los entornos educativos, ya que al estar el código fuente abierto, los alumnos podrán aprenderlo con más facilidad, y conocer las decisiones adoptadas por otros programadores para resolver los problemas que se les hayan ido presentando. Relacionado con lo anterior podríamos hablar de la innovación tecnológica que supone, ya que al compartir la información es posible un mayor avance de todo el desarrollo tecnológico.



REDUCCIÓN DE COSTES DE FORMACIÓN

Se disminuye el coste de la formación para los usuarios, ya que en el propietario, por lo general éste viene dado por la empresa creadora del software quien da licencias para autorizar la formación; en el libre por el contrario puede haber mayor competencia y desarrollo.



COLABORACIÓN ENTRE PERSONAS Y SOCIEDADES

Desarrolla una filosofía de colaboración y cooperación entre las personas y entre las sociedades, ya que en si mismo se potencia el compartir información y el construir de forma colaborativa.



DURACIÓN

Las acciones que se realizan pueden ser más duraderas, puesto que los propietarios del software propietario, valga la redundancia, pueden tomar la decisión de no continuar con la producción del software y su actualización debido a motivos económicos.



APROVECHAMIENTO POR OTRAS PERSONAS

Los programas producidos de forma propia o por encargo, siempre serán aprovechables para otras personas, especialmente si los programas se encuentran con la licencia GNU GENERAL PUBLIC LICENSE (GPL) <http://www.gnu.org/copyleft/gpl.html>.



LIBRE CIRCULACIÓN DE IDEAS

Se sigue un modelo muy parecido al mundo científico y académico, donde los hallazgos y los resultados de las investigaciones son dados a conocer y utilizados en consecuencia por otras personas para seguir avanzando en el conocimiento científico. Favorece en consecuencia, la libre circulación de ideas.



INDEPENDENCIA DE LOS PROVEEDORES

Independencia de los proveedores, ello permite que pueda seguir estando vigente independientemente de las versiones iniciales, o que los programas se puedan adaptar a las necesidades o gustos (idiomas, color, nivel de accesibilidad,...) de los creadores o proveedores.



COMUNIDADES DE DESARROLLO

Y por último, indicar que suelen existir verdaderas comunidades de desarrollo, impulso y ayuda a su utilización para los inexpertos, los cuales, no debemos olvidarlos, pueden ser exclusivamente usuarios y no productores, pero se pueden convertir en una masa crítica para la validación de forma desinteresada de los productos elaborados por los creadores, y puede conllevar un gran potencial para la generación de nuevo y mejor software.

Independientemente de las ventajas, su incorporación presenta una serie de inconvenientes, como por ejemplo:

■ Todavía no existe suficiente confianza en que un software de este tipo pueda aportar soluciones a los problemas de las instituciones y a la administración.

■ Desde los sectores industriales del software propietario se han establecido una serie de imágenes, que han calado en el imaginario colectivo que se maneja sobre este software, como son: es cosa de aficionados voluntarios en su tiempo libre, no hay negocio, no puede ser bueno ...

■ En cierta medida como consecuencia de los dos aspectos indicados anteriormente, los responsables de las instituciones no son capaces de adoptar medidas, que no siempre se les presentan como opciones coherentes.

■ La comodidad de muchos técnicos informáticos responsables en instituciones que prefieren pagar, y no hacer un trabajo y un esfuerzo creativo.

■ Tampoco podemos olvidarnos, que el marketing de las grandes empresas y distribuidores de software han realizado, y la cultura que con ella se ha imperado.

■ Su presencia no ha alcanzado un desarrollo en todas las tecnologías.

■ Falta de manuales para la formación de los usuarios en este tipo de software.

■ Y por último creemos que falta información sobre las posibilidades y diversidad de funciones que puede cubrir este tipo de software.

Aunque la presencia del software libre en las instituciones educativas va ganado progresivamente terreno, la realidad es que su presencia no ha sido tan mayoritaria como ha ocurrido en otros sectores, y ello es debido a una serie de razones como las siguientes:

- a) La formación del profesorado, b) La creencia de que los materiales educativos que los profesores tienen no pueden ser utilizados con software de carácter libre, c) La existencia de un grupo de “talibanes” de este software, que está llevando el discurso a los aspectos meramente éticos y políticos de utilización del software olvidándose del componente educativo, d) La jerga que muchas veces utilizan algunos colectivos de este tipo, que se consideran abiertos, pero utilizan un lenguaje encriptado que muchas veces desconcierta a los profesores, e) La creencia de que su mera utilización ya establece principios de calidad educativa f) La falta de accesibilidad inicial de los programas en software libre.

Por lo que respecta a sus posibilidades educativas, e independientemente de las posibilidades que el software de forma general incorpora a la educación, podemos señalar las siguientes:

- 1) La cultura de colaboración que introduce en la escuela. Y cultura de colaboración a diferentes niveles y con diferentes actores. Cultura de colaboración entre programadores y usuarios (profesores/estudiantes), cultura de colaboración entre profesores, y cultura de colaboración entre estudiantes.

2) La cultura de colaboración que introduce entre los profesores que utilizan este tipo de software. Se han creado verdaderas comunidades virtuales para el intercambio de programas, información, objetos de aprendizaje y prácticas educativas.

3) Favorece el desarrollo de una cultura tecnológica entre los estudiantes, que no se ven como meros consumidores de software informáticos, sino que se adentran en su funcionamiento, con la posibilidad de realizar cambios, lo que favorece el desarrollo de una alfabetización informática de los estudiantes.

4) La diversidad de programas a los cuales podemos acceder en la actualidad.

5) La distribución de copias entre los estudiantes para que puedan trabajar también en sus casas.

No nos gustaría finalizar sin hacer una llamada de atención, respecto a que el uso del software, independientemente del tipo que sea, sin su incorporación a una estructura educativa, y con la aplicación de unas estrategias didácticas concretas por parte del profesorado no garantiza una acción educativa de calidad.

BIBLIOGRAFÍA

■ **ABELLA, A y otros (2003):** Libro Blanco del Software Libre en España 2003, Consejería de Educación, Ciencia y Tecnología de la Junta de Extremadura Asociación de Usuarios Españoles de GNU/LINUX, pdf.

■ **ADELL, J. y BERNABÉ, Y. (2007):** Software libre en educación, en CABERO, J. (coord): Tecnología educativa, Madrid, McGraw-Hill, 173-195.

■ **CABERO, J. y LLORENTE, M.C. (2007):** Experiencias educativas mediante la aplicación de software libre, en CABERO, J. (coord): Nuevas tecnologías aplicadas a la educación, Madrid, McGraw-Hill, 309-323.

■ **MAS i HERNANDEZ, J. (2005):** Software libre: técnicamente viable, económicamente sostenible y socialmente justo, Barcelona, Infonomía.

■ **STALMMAN, R. (2004):** Software libre para una sociedad libre, Madrid, Traficante de sueños.

