



Software libre

en la educación de profesionales del diseño

Ana María Valero Quezada*
César Camacho Pérez*

* Universidad Autónoma de Ciudad Juárez. Departamento de Ciencias Administrativas. Maestra en Administración de Empresas.

* Universidad Autónoma de Ciudad Juárez. Departamento de Diseño. Maestro en Tecnología Educativa.

Resumen

El presente análisis muestra las ventajas que existen en el uso de *software* libre. Su importancia radica principalmente en asegurar una formación en competencias accesible para todos los estudiantes. Con programas que no establecen límites en su creatividad —los cuales les permiten estudiarlos, modificarlos, distribuirlos y aprovecharlos— será posible que los alumnos generen modelos de apropiación social tecnológica. La combinación de *software* libre y *software* privativo abre el abanico de posibilidades en un programa educativo en el cual la creatividad es una característica irrefutable, por lo cual entre mayores sean las herramientas que el alumno tenga a su disposición, mayor será la preparación para los ambientes de trabajo en la actualidad. Además de simplificar el aprendizaje, el *software* libre ahorra recursos a la comunidad universitaria e innegablemente estrecha la relación entre los procesos de innovación y la calidad educativa.

Palabras clave

SOFTWARE LIBRE, SOFTWARE PRIVATIVO, INNOVACIÓN, DISEÑO GRÁFICO, FORMACIÓN EN COMPETENCIAS

Abstract

THIS ANALYSIS SHOWS THE ADVANTAGES IN USING FREE SOFTWARE. Its importance lies primary in securing a skills

training accessible to all students. With programs that do not set limits in their creativity, the students are able to study, modify, distribute and use them with the possibility of generate models of social appropriation of technology. The combination of free software and proprietary software opens the range of possibilities in an educational program in which creativity is a compelling feature, so that between greater the tools that students have a higher availability for the present-day working environments. In addition to simplifying learning, free software saves resources for the university community and undeniably closes relationship between innovation processes and educational quality.

Keywords

FREE SOFTWARE, PROPRIETARY SOFTWARE, INNOVATION, GRAPHIC DESIGN, SKILLS TRAINING

Introducción

¿EL USO DE SOFTWARE LIBRE DARÁ UN VALOR AGREGADO A LA formación de profesionales del diseño? El *software* libre (SL) contribuye a la formación integral y personalizada de los profesionales; además, se puede adaptar a los requerimientos de la gestión docente y de los programas educativos (Avello, s.f.). Cobo señala que una cultura orientada a compartir el conocimiento, a abrir canales de intercambio, es una cultura que tiene más posibilidades de crear, de innovar y de crecer. Esa es justamente la

filosofía del SL (2014). Feltrero añade que la sociedad reclama el control del desarrollo tecnológico, para poder crear y recrear las tecnologías básicas, según las necesidades sociales e individuales (2015). Las universidades como incubadoras de nuevos profesionales, deben pensar en egresados conscientes en eficientar los recursos desde estándares legales. El SL es *software* legal y a través de un proceso de implantación y capacitación puede ayudar a sustituir el uso de *software* pirata e ilegal (Galván, 2016). El SL aun y cuando no es un *software* gratuito, a la larga resulta más eficiente en costo que el *software* privativo. En la Ciudad de México, durante el foro “Software Libre en México, reflexiones y oportunidades”, llevado a cabo el 2 de junio de 2011, en el auditorio Sebastián Lerdo de Tejada del Senado de la República, se estableció la importancia de potenciar y no limitar las habilidades de los educandos y de aprovechar la existencia de proyectos de SL maduros que incorporan conocimiento de punta; también se propuso que las universidades deberían incorporarse a dichos proyectos educativos para estudiarlos, modificarlos, distribuirlos y aprovecharlos en sus actividades de vinculación, generando modelos de apropiación social tecnológica.

Desarrollo

EL *SOFTWARE* LIBRE NACIÓ DE LA INICIATIVA DE RICHARD STALLMAN por crear un sistema operativo que tuviera las cuatro libertades que él consideró fundamentales: libertad de uso, libertad de estudio, libertad de redistribución, libertad de mejora y publicación de dichas mejoras.

En 1983, Stallman publica en net.unix-wizards:

Iniciando este día de Acción de Gracias, voy a escribir un sistema de software completo compatible con Unix llamado GNU (siglas de Gnu no es Unix), y lo distribuiré libre a quienes quieran usarlo. Considero que la regla de oro exige que si yo quiero un programa debo compartirlo con otras personas que también lo quieren. No puedo, conscientemente firmar un acuerdo de confidencialidad o un acuerdo

de licencia de software. Para que yo pueda seguir utilizando las computadoras sin violar mis principios, he decidido reunir suficiente Software Libre, de manera de poder continuar sin utilizar algún software que no sea libre (...) estoy buscando personas para quienes ayudar a la humanidad sea tan importante como el dinero.

En 1984, Stallman abandonó su puesto como profesor en el MIT (Massachusetts Institute of Technology) y un año más tarde fundó la Free Software Foundation (FSF) e inició el proyecto denominado GNU, el cual hoy en día tiene una difusión universal. En los años posteriores miles de seguidores han contribuido para producir *software* de calidad de libre distribución. El caso más conocido es Linux. Hoy en día es posible descargar *software* libre para diversas aplicaciones. Como el código de acceso es abierto, los programadores alrededor del mundo pueden detectar errores y proponer mejoras, luego publicar nuevamente el *software* con estas mejoras en un proceso de mejora continua (Mongue, 2015).

TABLA 1. Las libertades del *software* libre.

CERO	Libertad para utilizar el programa, sea cual sea el propósito.
UNO	Libertad para estudiar cómo funciona un programa y capacidad para adaptarlo a las propias necesidades. El acceso al código fuente es una condición <i>sine qua non</i>
DOS	Libertad para redistribuir copias, con lo cual puedes ayudar a otras personas
TRES	Libertad para mejorar un programa y hacer públicas esas mejoras a los demás para que toda la comunidad se beneficie. El acceso al código fuente es una condición <i>sine qua non</i> .

Fuente: Elaboración propia.

Ventajas del Software Libre

EL SL TIENE VENTAJAS SOBRE EL *SOFTWARE* PRIVATIVO O COMERCIAL, pues simplifica el proceso de aprendizaje. Por otro lado, ahorra recursos a la comunidad universitaria y finalmente forma una visión innovadora en los estudiantes.

En primer lugar, simplifica el proceso de aprendizaje. Las ventajas del SL están definidas por sus libertades (ver Tabla 1), las cuales permiten el uso,

modificación, distribución y mejora del programa, lo cual para el *software* privativo es ilegal, debido a que los programas de computadora son una obra intelectual, al igual que un libro o una canción y su reproducción y distribución sin permiso del autor está prohibida. González y Romero (2015) escriben sobre las ventajas del uso del *software* libre en educación, ampliando sobre dichas libertades. Afirman que el hecho de que el SL se pueda copiar y distribuir permite que las universidades utilicen sus recursos en formación de profesores o inversión en ordenadores, así que el SL puede ayudar a los países en vías de desarrollo a dotar de infraestructura tecnológica a sus escuelas y a paliar la 'brecha digital' con los países desarrollados". Indican que el SL permite conocer el funcionamiento y el código fuente para su modificación, con lo que ayuda a los estudiantes a enfrentar desafíos que les facilitan enormemente el aprendizaje, pues los estudiantes tendrán buenos modelos y ejemplos para modificar, lo cual no sucede con el *software* privativo, ya que este tiene códigos privados y tratar de saber cómo funcionan o tratar de modificarlos está prohibido por la ley, contribuyendo con ello a la ignorancia tecnológica. Señalan que el SL ayuda a los estudiantes en su proceso de crear valores de cooperación y solidaridad, lo cual, en informática, significa "compartir software"; utilizando SL podrán, sin incurrir en ninguna falta, hacer copias a todos los compañeros de clase o llevar a casa el que se usa en la escuela.

Formar a los estudiantes con *software* libre es enseñarles que el conocimiento se construye y que es fruto de la libertad.

La integración de las nuevas concepciones en educación, los nuevos materiales a disposición, del también nuevo proceso de enseñanza-aprendizaje y, sobre todo, las diferentes necesidades de formación de los nuevos alumnos, han forzado la creación y desarrollo de nuevos métodos para conseguir un proceso de formación eficaz (Moreno, 2009).

Según la filosofía de Richard Stallman existen muchos motivos para incorporar el *software* libre

en la educación, pero el principal es considerar al estudiante como un ser integral, a quien se debe enseñar la importancia de la libertad, guiándolo en el uso de la misma, haciéndole comprender que debe estudiar hasta donde su curiosidad lo lleve, que debe profundizar en su conocimiento y que esto implica una responsabilidad, el *Software Libre* es esta responsabilidad, la cual es inherente a todos sus realizadores y que el estudiante encara al momento de usar las libertades del SL.

Los avances tecnológicos y la incorporación de las tecnologías en casi todos los campos han empujado a las universidades a encontrar formas de integrar estas tecnologías en la forma de transferir el conocimiento, aumentando la complejidad del panorama educativo. Los nuevos profesionistas requieren de conocimientos y habilidades específicos que sean inmediatamente aplicables a su potencial área profesional. Programas como Blender, Inskape, Gimp, Open Office Draw, pueden ayudar al estudiante en el diseño 3D, creación de videos, edición de fotografía, edición de videos y mucho más. Todos estos programas pueden instalarse una vez que se ha establecido Linux Ubuntu como sistema operativo con un simple clic en instalar o desinstalar a una velocidad bastante aceptable.

Los profesionistas que mayor posibilidad tendrán de insertarse en las áreas laborales deberán tener un amplio conocimiento en las Tecnologías de la Información, pero sobre todo contar con las habilidades que les permitan ofrecer a sus empleadores un valor agregado en su currículo. Recientemente, el sistema universitario español, en la Conferencia de Rectores de Universidades Españolas, ligó el futuro de la educación al SL: "contribuye a hacer una Universidad más eficiente desde el punto de vista económico, fomenta el desarrollo interno y la colaboración con otras universidades e instituciones" (CENATIC, 2014).

En segundo lugar, ¿cuánto invierten las universidades y su comunidad en licencias de programas privativos o propietarios necesarios para cumplir con los contenidos de las asignaturas?, ¿cuánto invierte la universidad en licencias de programas privativos o propietarios necesarios

para la gestión administrativa? ¿Cuánto tienen que invertir los alumnos en *software* para poder estudiar o realizar sus tareas fuera de las instalaciones de la universidad? El SL ahorra recursos a la comunidad universitaria, compuesta por alumnos, docentes y trabajadores. El SL permite la accesibilidad a todos los miembros de la mencionada comunidad, fomentando el uso de un sistema de desarrollo más justo social y económicamente. El SL aun y cuando no es un *software* gratuito, a la larga resulta más eficiente en costo que el privativo. Por ponerlo de otro modo, en una licenciatura se requieren, al menos, algunas licencias para *software* privativo o propietario que permita a los estudiantes alcanzar los objetivos de las asignaturas como adobe, flash, coreldraw, photoshop y otros. Siguiendo con este ejemplo y suponiendo que la universidad tenga todas estas licencias disponibles en sus instalaciones para el uso de los alumnos, ¿pueden estos llevar el *software* a casa para realizar sus tareas y estudiarlo al grado de poder hacer sugerencias de mejoras? ¿Pueden los maestros interesar a los alumnos en *software* que pueden usar, pero con la precaución de no causar un problema técnico que no se pueda solucionar? ¿Puede este *software* privativo contribuir a la innovación? La respuesta es no, a menos que los alumnos, con permiso o no de los docentes, realicen copias piratas para su práctica diaria. Las universidades deben pensar en egresados conscientes en eficientar los recursos desde estándares legales. El SL es legal y a través de un proceso de implementación y capacitación puede ayudar a sustituir el uso de *software* pirata e ilegal (Galván, 2016).

El bajo costo de los productos libres permite a las universidades emplear sus recursos en rubros distintos. Las licencias de SL permiten la instalación del *software* tantas veces y en tantas máquinas como el usuario lo desee. El acceso al código fuente permite el desarrollo de nuevos productos sin la necesidad de desarrollar todo el proceso partiendo de cero. Sin el secretismo tecnológico no existe freno al desarrollo. El SL no se vuelve obsoleto, cada seis meses se pueden descargar directamente las actualizaciones en internet, por

lo cual se descartan también los costos de actualización a los que estamos obligados cada vez que sale al mercado una nueva versión del *software* privativo o propietario que se ha adquirido, pues el propietario de este diseña una nueva versión que invalida la anterior y de ninguna manera genera un soporte de ayuda para eliminar las fallas de la versión anterior; la idea es generar una necesidad nueva en el producto que lleve a los consumidores a comprar la nueva versión.

Por otro lado, cuando en una universidad se utilizan diferentes programas, lo más seguro es que los formatos no sean compatibles, o cierto *software* no pueda ser instalado en ciertas computadoras, esto también se elimina con el SL. Linux Ubuntu es un sistema operativo mantenido por una comunidad de desarrolladores orientado al usuario promedio, que está compuesto por múltiple *software* de licencia libre o código abierto. Su patrocinador es Canonical, que se financia a través del soporte técnico y de servicios vinculados con el sistema operativo, y se beneficia de la comunidad de desarrolladores, quienes siempre están mejorando los componentes del sistema operativo. Es un sistema operativo hecho para humanos, como su eslogan lo señala: "Linux for human beings"

En tercer lugar, innovación es la "creación o modificación de un producto y su introducción en el mercado" (RAE, 2017). El SL permite a los profesionistas la práctica de modificar y publicar mejoras.

Las habilidades de innovación que los estudiantes adquieren durante su preparación separan a los preparados para los ambientes de vida del siglo XXI, de aquellos que no lo están (Consortio de Habilidades Indispensables para el Siglo XXI, 2016).

Además:

"El poder acceder al código fuente es un estímulo permanente para la apropiación de las nuevas tecnologías y la innovación (...) el SL se puede adaptar a las necesidades locales y cualquier persona con conocimientos necesarios puede participar de la construcción, adaptación y uso de programas de computadoras.

TABLA 2. Programas de código abierto para el diseño.

SOFTWARE LIBRE	TAREAS QUE EJECUTA	LINK PARA DESCARGAR	SOFTWARE PRIVATIVO
Blender	Edición de gráficos 3D	http://www.blender.org/	3DStudio, Max
Cinelerra	Edición no lineal de video	http://cinelerra.org/	Adobe Premiere Media StudioPro
CinePaint	Posproducción de video, retoque de fotogramas en secuencia (películas)	http://www.cinepaint.org/	Adobe After Effects y Final CutPro
Fontforge	Ceación y edición de tipografías digitales	http://fontforge.sourceforge.net/	Macromedia Fontographer
FontMatrix	Gestor de fuentes	http://fontmatrix.net/	Font Expert
GIMP	Edición potentes de imágenes y retoque fotográfico	http://www.gimp.org.es/	Photoshop
Inkscape	Edición de gráficos vectoriales	http://www.inkscape.org/?lang=es	Adobe Ilustrador
Kinodv	Edición no lineal de video	http://www.kinodv.org	Adobe Premiere
Komposer	Diseño Web	http://kompozer.net/	Dreamweaver
Ktoon	Animación 2D	http://ktoon-es.tonka.com/ http://taringa.net/	Adobe Flash
Open Office-Draw	Decorar texto	http://www.openoffice.org/product/draw.html	WordArt
Processing	Producción de proyectos multimedia e interactivos de diseño digital	http://processing.org/download/	Basado en JAVA
Qcad	Dibujo 2D	http://www.ribbonsoft.com/qcad.html	Autocad
Transcode	Convertidor de video	http://www.transcoding.org/cgi-bin/transcode	VirtualDub
Scribus	Maquetación y publicación	http://www.scribus.net/	Adobe In Design o Adobe PageMaker
StopMotion	Animación	http://developer.skolelinux.no/info/studentgrupper/2005-hig-stopmotion/index.php?side=3	
SynfigStudio	Animación 2D	http://www.synfig.org/	Adobe Flash

Fuente: elaboración propia.

Esta es una forma real y tangible de acercar las nuevas tecnologías a nuestra gente, acercando las posibilidades, en lugar de mostrar el software como la obra de algún iluminado que seguramente vive en un país desarrollado y necesita una enorme infraestructura para su trabajo (Da Rosa & Heinz, 2016)

Es innegable la estrecha relación entre los procesos de innovación y la calidad educativa, y de manera específica, en el caso de las universidades, con el uso de SL le estamos mostrando a los estudiantes que queda mucho por hacer, que no todo está terminado y que ellos pueden ser parte de la construcción, modificación y difusión del software.

En SL hay un sin número de posibilidades para utilizar en las diferentes áreas del diseño para revisar algunas de ellas ver tabla 2.

La *Guía Práctica sobre el Software Libre, su selección y aplicación local en América Latina y el*

Caribe, publicada por la UNESCO, contiene una larga lista de programas de SL con sus correspondientes links; además, los estudiantes pueden descargarla, editarla y compartirla con toda la comunidad, lo cual la hace una guía completa y actualizada sobre los programas de SL que existen. Para 2015 existían alrededor de cien mil proyectos de SL, en diferentes etapas, desde planificación hasta programas maduros, toda una variedad para cubrir realidades que requieren soluciones particulares. Algunas consideraciones son muy importantes como: 1. Conocer si es un proyecto vivo, 2. Conocer si es un proyecto aceptado, 3. Conocer si es un proyecto estable y maduro, 4. Conocer su funcionalidad y 5. Conocer su tendencia (Da Rosa, 2016).

Conclusión

PERTENECER A UNA COMUNIDAD DE SL PERMITE UTILIZAR EL modo cooperativo de construcción, sin tener que empezar de cero cada vez, ya que los desarrolladores de SL pueden acudir a colecciones de programas y bibliotecas adecuados a sus necesidades, funcionales y probadas. El *software* aumenta en valor mientras más lo compartimos, el efecto red hace a todo programa más útil y más valioso mientras más gente lo usa, pues permite intercambiar información y resultados con otros usuarios. Sin duda alguna no todo está dicho en materia de recursos tecnológicos para el diseño, por ello proponer modificaciones e incorporación de nuevas herramientas que supongan la innovación educativa en pro de elevar los índices de calidad educativa será siempre motivo de análisis.

APROVECHANDO LAS CARACTERÍSTICAS DE CÓDIGO ABIERTO Y adaptabilidad del Software Libre es posible dar un valor agregado a la formación integral de los estudiantes, ya que los profesionistas que mayor posibilidad tendrán de insertarse en las áreas laborales deberán tener un amplio conocimiento en las Tecnologías de la Información, pero sobre todo contar con las habilidades que les permitan ofrecer a sus empleadores un valor agregado en su currículo.

Bibliografía

Libros

- Feltre Oreja, R. (2015). *Software libre y la construcción ética de la sociedad del conocimiento*. Barcelona: Icaria.
- González Arrabal, E., & Romero Granados, S. (2015). *Educación temprana a las TIC: estrategias para educar en un uso responsable en educación infantil y primaria*. (S. G. Técnica, Ed.) Madrid, España: Ministerio de Educación y Ciencia.

Artículos en revistas

- Avello, R. (s.f.). *Software libre: algunas herramientas de excelencia para la educación a distancia*. Recuperado el 9 de junio de 2016, de <http://dim.pangea.org/docs/raidellavello2.pdf>
- Cobo Romani, J. C. (2014, abril). Conocimiento, creatividad y *software* libre. Una oportuni-

dad para la educación en la sociedad actual. *UOC Papers. Revista sobre la Sociedad del Conocimiento.*, 1-8.

- Galván, J. (2016, agosto). Ejercicio de creación de oferta y demanda de *software* libre. *Revista Linux*, págs. 16-21.
- Mongue, S. (2015, abril). ¿Es aplicable el modelo de producción del *software* libre a contenidos educativos? *RED. Revista de Educación a Distancia*. Murcia, España.

Fuentes en la red

- Cenatic. (2014, 13 de abril). Recuperado el 8 de junio de 2016, de Centro Nacional de Referencia de Aplicación de las Tecnologías de la Información y la Comunicación: <http://www.cenatic.es/cenatic-en-la-red/bienvenido-a-la-universidad-la-republica-independiente-del-software>
- Consortio de Habilidades Indispensables para el Siglo XXI. (2016, 1 de septiembre). *21st Century Students Outcomes*. Retrieved October 10, 2016, from www.21stcenturyskills.org
- Da Rosa, F. (2016). *Guía práctica sobre Software Libre, su selección y aplicación local en América Latina y el Caribe*. Retrieved 12 02, 2016. from <http://unesdoc.unesco.org/images/0015/001560/156096s.pdf>
- RAE (2017). Real Academia Española.

Tesis

- Moreno Clari, P. (2009). *Análisis del uso universitario de plataformas de gestión del aprendizaje. Estudio de caso en la Universitat de València*. Valencia.