

Software libre y código abierto en educación

Zulma Cataldi y Fernando Salgueiro
Facultad de Ingeniería. Universidad de Buenos Aires
liema@fi.uba.ar

Resumen

A través de la presentación de las características y ventajas del software libre y de código abierto se busca que los docentes puedan identificar sus necesidades y encontrar productos alternativos aptos para sus programas educativos.

Si bien no quedan resueltas todas las necesidades con un solo paquete se puede recurrir a diferentes programas atendiendo a cada una de ellas; siempre a un costo menor y con una mayor vida útil de los recursos de hardware disponibles; entre otras posibilidades.

Palabras clave: Software libre, código abierto en educación

1. Introducción

En América Latina tanto en el sector privado como en el Estado se observa un incremento en la utilización de GNU¹/Linux, según la compañía de estudios en tecnología IDC², se espera que la venta de paquetes de software GNU/Linux para computadoras de escritorio y servidores a nivel mundial siga en aumento debido a las empresas que adoptan software libre para administrar sus negocios debido a razones tales como: reducción de costos, estabilidad, flexibilidad, estándares abiertos, calidad y seguridad, independencia de plataforma, escalabilidad, mayor rendimiento, multiplataforma, independencia del proveedor, personalización de los sistemas, entre otras.

Esta tendencia se fundamenta en motivos:

- *Económicos:* El objetivo es poder reducir los costos de las licencias de ambos tipos de aplicaciones, por ejemplo que el sistema operativo sobre el cual corre la aplicación específica sea software libre.
- *Legales:* Según informes de la ONG³ Software Legal, los índices de piratería en Argentina alcanzan el 65 por ciento, por lo que se deberá prever la forma de regularizar esta situación por lo que una alternativa a considerar es el software libre.
- *Estratégicos:* El software libre se basa en la utilización de estándares abiertos para su desarrollo, esto garantiza dos condiciones: primero la interoperabilidad entre sus aplicaciones y segundo y más importante la independencia en la elección de la aplicación.
- *Morales:* Copiar software es un delito aunque el objetivo sea para ayudar a alguien. Por otra parte, existe una concientización de los individuos y un rechazo a aportar

¹ GNU es un acrónimo recursivo para "GNU No es Unix"

² IDC sitio web www.idc.com

³ ONG Organización No Gubernamental

dinero a las grandes “megacorporaciones” que no ayudan a desarrollar la industria local.

2. Algo de historia del software

Durante los años sesenta el software se consideraba un servicio, no un producto en sí mismo, y se lo concebía como un agregado necesario para que los usuarios pudieran utilizar las computadoras. En esa época era usual que los usuarios y los programadores compartieran libremente sus aplicaciones. Pero, hacia fines de los años setenta las compañías comenzaron a aplicar algunas restricciones a los usuarios debido a la implementación de las licencias, es decir, a través de contratos o acuerdos de uso entre el dueño del *copyright* y el usuario de la aplicación, en el cual se establecían los límites de uso, modificación y distribución.

En 1985, Richard Stallman creó la *Fundación del Software Libre*⁴ (FSF, por sus siglas en inglés) y acuña el concepto de *copyleft* como posible traducción: izquierdos de autor y da una definición para el software libre. Según la definición de la FSF (Free Software Foundation, 2005) el software se considera libre si garantiza las cuatro libertades básicas de la Tabla 2:

- libertad 0: La libertad de usar el programa, con cualquier propósito
- libertad 1: La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades
- libertad 2: La libertad de distribuir copias, con lo que puedes ayudar a tu vecino
- libertad 3: La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

Tabla 2: Libertades básicas.

Cabe destacar que el cumplimiento de las libertades 1 y 3 implica el acceso a las fuentes del programa, o sea al código de fuente, lo que permite que cualquier programador con un editor de texto simple pueda entender su funcionamiento.

Considerando la ambigüedad del término *libre* (en inglés se considera free a todo aquello que es libre o gratis) es necesario aclarar que se refiere a “*libre*” en el sentido de tener las libertades descritas y ello no implica que sea gratuito necesariamente.

En contraposición a este modelo se llama software privativo o propietario (SP) a todo programa que prive a los usuarios de las libertades para usarlo, modificarlo y distribuirlo o que requiera que se solicite autorización para ello. Este término también es aplicable al software cuyo costo es tan elevado que no puede ser afrontado por un particular, que son casos de los programas diseñados para realizar tareas muy específicas.

Hacia 1986 se puso en marcha el proyecto GNU, se buscaba concebir un sistema operativo versión libre como alternativa al Unix existente que estuvo casi listo en 1989, y solo faltaba el núcleo o kernel. Recién en 1991 se completó el sistema operativo GNU/Linux cuando Linus Torvald liberó el *kernel*⁵ Linux bajo la licencia GPL⁶ y lo adoptó el proyecto GNU.

⁴ <http://www.fsf.org/>

⁵ es la parte fundamental de un sistema operativo.

⁶ GPL es General Public License.

Las siglas (GPL, GNU, etc.) vistas representan las distintas licencias que existen sobre el software libre. En la Tabla 3 se resumen las más populares en la actualidad.

Durante 1988 algunos miembros de la comunidad del *SL* comenzaron a utilizar la denominación *Código Abierto* (OSS, por su sigla en inglés) con el objeto de evitar la ambigüedad de la palabra libre (free) en inglés. Con el tiempo nació un nuevo movimiento que si bien es compatible con el *SL*, sus intereses finales difieren.

<i>Tipo de Licencia</i>	<i>Característica</i>	<i>Principales representantes</i>
<i>Licencia de software libre sin protección heredada</i>	Se pueden crear una o varias obras derivadas de la original sin necesidad de respetar la licencia original, es decir, a partir de un producto con licencia libre se puede generar uno con licencia no libre.	Academic Free License Apache Software License BSD License, MIT License. NCSA Open Source License; W3C Software Notice and License.
<i>Licencia de software libre con protección heredada</i>	Todas las obras derivadas de la original tendrán algunas de las características de la licencia original, aunque no todas.	Common Public License GNU General Public License GNU Lesser General Public License Mozilla Public License
<i>Licencias semilibres</i>	Las aplicaciones serán siempre para obra final sin fines de lucro.	Para desarrollos de software individuales y por lo tanto se deben analizar las aplicaciones por separado.
<i>Licencias semilibres Antagónicas</i>	Existen restricciones a la cantidad de usuarios o licencias: por ejemplo para n usuarios con licencias libres y para n+1 aplican licencias no libres.	Para desarrollos de software individuales, se deben analizar las aplicaciones por separado. Un ejemplo de esto podría ser el MS SQL para un solo usuario que no posee cargo asociado, mientras que para más si.
<i>Licencias no libres</i>	Existen restricciones a su uso, copia, redistribución, etc.	Son la mayoría de las aplicaciones de software a medida que se desarrollan y también la de los “enlatados” más utilizados.

Tabla 3. Distintas licencias de software libre

El software de código abierto (OSS) es software cuyo código fuente está disponible públicamente, aunque los términos de licenciamiento específicos varían. Las licencias deben cumplir diez condiciones para ser consideradas licencias de OSS (Open Source Initiative, 2005):

1. Libre redistribución: el software debe poder ser regalado o vendido libremente.
2. Código fuente: el código fuente debe estar incluido u obtenerse libremente.
3. Trabajos derivados: la redistribución de modificaciones debe estar permitida.
4. Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.
5. Sin discriminación de personas o grupos: nadie puede quedar afuera.
6. Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.
7. Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa
8. La licencia no debe ser específica de un producto: el programa no puede licenciarse sólo como parte de una distribución mayor.
9. La licencia no debe restringir otro software: la licencia no puede obligar a que otro software que sea distribuido con el software abierto deba también ser de código abierto.
10. La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic del mouse o de otra forma específica del medio de soporte del software.

A partir de aquí surgen las dos vías conceptuales del SL. Por un lado, se encuentra la visión de la FSF en la cual el objetivo final está relacionado con el carácter moral y éste prima sobre la excelencia técnica del software. Y por otro, se sitúa la *OSI* (Open Source Initiative) donde compartir el código fuente es un medio para la búsqueda de la excelencia técnica.

A pesar de las diferencias filosóficas de ambos movimientos pocas veces suele haber interferencia en el desarrollo y en la colaboración entre ambos proyectos. Sólo a los fines prácticos se llamará a ambas iniciativas con la denominación de SL con el objeto de facilitar la lectura.

Debido a que el SL permite el libre uso, modificación y redistribución, es recomendable su aplicación en los países del tercer mundo donde el costo de las licencias del software privativo es a veces inaccesible. También, brinda la posibilidad de modificarlo localmente, permite que sea posible su traducción a idiomas que no son necesariamente rentables comercialmente.

3. El software libre en educación

Las características del SL lo hacen propicio, ya que los usuarios tienen la libertad de usarlo, ejecutarlo, copiarlo, distribuirlo ó mejorarlo sin necesidad solicitar permiso a nadie ya que goza de las libertades de la Tabla 1.

Estas condiciones lo tornan ventajoso respecto del software propietario ya que en éste no hay autonomía para usarlo, copiarlo y redistribuirlo, el usuario no tiene el código fuente y las modificaciones que desee hacer deben ser solicitadas al propietario del mismo.

“La primera (razón para usarlo en las escuelas) es que el software libre supone un ahorro de costos para las escuelas. El software libre le da a las escuelas, igual que a cualquier otro usuario, la libertad de copiar y redistribuir el software, por lo que pueden hacer copias para todas las computadoras que tengan. En los países pobres esto puede ayudar a reducir la brecha digital” (Stallman, 2004).

El software libre les permite a los estudiantes aprender cómo funciona y para aprender a escribir buen software, los estudiantes necesitan escribir y leer mucho código, leer y comprender programas reales y el software privativo no les permite aprender en este sentido. Las escuelas deberían decirle a sus alumnos si llevan software a la escuela, deben compartirlo con los demás niños, por ello el software que instalen debería estar disponible para que los alumnos lo copien, se lo lleven y lo redistribuyan tanto como quieran (Stallman, 2004).

El SL, como se señaló, es una cuestión de libertad no de costos, ya que el usuario de este software tiene la libertad de distribuir copias modificadas o no, ya sean gratis o cobrando por su distribución, de este modo no hay exigencias de pagar o pedir permisos.

Harari (2006) señala que una entidad puede continuar con la actualización de su software propietario o considerar la posibilidad de migrar sus sistemas a software libre, para lo cual deberá analizar ciertos aspectos tales como los que se señalan en la Tabla 4.

El software libre se puede conseguir en forma gratuita o pagando por su distribución. En el mercado existen diferentes precios y las instituciones pueden elegir la opción que más convenga. Se pueden realizar todas las copias que se requieran del mismo en diferentes computadoras, sin necesidad de pagar por ellas. No se necesita realizar actualizaciones y

siempre se encontrará soporte técnico. El software libre permitiría a las instituciones reutilizar hardware obsoleto, ya que existen más posibilidades para hacerlo.

El desafío que enfrentan las organizaciones al evaluar si un producto *Open Source* particular es factible o no, difiere significativamente del producto comercial análogo. El desafío de la organización en la elección de un paquete comercial, se centra en identificar cuál es el vendedor que le ofrece el producto más completo: es decir, software, soporte, entrenamiento, etc. Los usuarios de *Open Source*, a su vez deben localizar y evaluar los componentes individuales del producto para armar un paquete completo que cubra las necesidades de la organización.

<i>Aspectos</i>	<i>Al migrar al software libre</i>	<i>Actualizando el software propietario</i>
<i>Tecnológicos</i>	Se dispone del código fuente, se puede adaptar y modificar de acuerdo a las necesidades propias y aporta crecimiento tecnológico para la entidad.	No se dispone del código fuente y las modificaciones hay que requerirlas al creador del software. El crecimiento tecnológico es de las la empresa creadora del software.
<i>Formativos</i>	La formación de los usuarios en el uso de este tipo de software es una inversión, los conocimientos que se adquieran servirán para el futuro. Los programas <i>no</i> cambian por cuestiones comerciales.	La formación de los usuarios en este tipo de software representan un gasto, los conocimientos adquiridos pueden no servir en el futuro. Los cambios, a veces superfluos, se realizan por cuestiones comerciales.
<i>De actualización</i>	La actualización no depende de intereses comerciales, sino de la necesidad de la entidad. Siempre habrá soporte técnico para las mismas.	Generalmente pasado cierto tiempo hay que realizar actualizaciones por más que no las necesite, dado a veces se deja de dar soporte técnico al software obsoleto.
<i>De hardware</i>	Al no estar obligado a hacer actualizaciones del software no está obligado a hacer actualizaciones del hardware. El software libre permite posibilidades de volver a aprovechar el hardware obsoleto.	Las actualizaciones de software normalmente implican actualizaciones de hardware, ya que requieren hardware más potente.
<i>De estándares</i>	El software libre se basa en estándares, eso le da la posibilidad al usuario de seleccionar un espectro de aplicaciones amplio, actualizando el software propietario.	El software propietario no se basa, por lo general, en estándares, por lo tanto, el usuario se encontrará que es un cliente cautivo. El usuario deberá moverse dentro de las limitaciones del software propietario.
<i>De seguridad</i>	Al tener acceso al código fuente el usuario puede saber lo que hace el programa y si es seguro.	Con el software propietario no se puede saber si es seguro o no
<i>De compatibilidad</i>	Al almacenar, siempre, los datos en formato estándar, por más que las versiones evolucionen se puede acceder a la información..	La información almacenada en formato propietario, en general no es compatible con versiones mas antiguas.
<i>De costos</i>	Se puede instalar el software, en todas las máquinas que quiera, ya que se tiene la libertad de realizar y redistribuir todas las copias que desee. El software puede no tener costo, o se puede pagar por una distribución. Se puede elegir el precio que más conviene.	Se deberá pagar la licencia de cada máquina dónde se quiera instalar el programa propietario y en general los costos son elevados.
<i>De caducidad</i>	Se puede utilizar la versión del software que quiera todo el tiempo que lo desee ya que se podrá contratar a cualquiera para que le de soporte. Las licencias que no tienen tiempo límite, generalmente, no son válidas para las versiones posteriores.	Pagar por una licencia, no implica que sea para siempre ni que con ella se pueda acceder a las diferentes versiones del software. Algunas licencias tienen un tiempo límite y vencido ese lapso, se deberá volver a pagar por las ellas.

Tabla 4: Aspectos a considerar para la migración (Harari, 2006)

4. Algunos proyectos de SL para educación

A continuación se resumen algunas iniciativas:



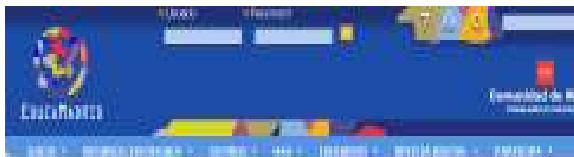
a) *Proyecto LinuxForKids*⁷: Fue creado en 1999 con el objetivo de promover el uso de Linux como una plataforma educativa y de entretenimiento para chicos.

En un comienzo fue pensado para una audiencia de menores de 10 años, porque se pensaba que no había mucho software para niños de esa edad, pero actualmente este sitio está pensado como un lugar para que los padres y los maestros puedan acceder en busca de software educativos.



b) *Proyecto SchoolForge*⁸: El objetivo fue de unificar las organizaciones independientes que utilizaban, y desarrollaban los recursos abiertos en

educación para la educación primaria y secundaria. La idea fue: trabajar juntos. En este proyecto participan escuelas, empresas y organizaciones aportando ideas, software y materiales didácticos.



c) *Proyecto EducaMadrid*⁹: Tiene como objetivos: 1) Promover el uso de tecnologías de código abierto en educación, 2) Facilitar a los centros educativos, y en general a los miembros de

la comunidad educativa, software que contribuya a su integración en la sociedad de la información y del conocimiento, 3) Facilitar aplicaciones de calidad e interés educativo sin costos adicionales para el alumnado y el profesorado y 4) Disponer de un entorno de código abierto que se utilice en procesos de formación e investigación. Han desarrollado su distribución propia: *MAX (MAdrid_LinUX)* que un sistema operativo con aplicaciones de código abierto.



d) *Proyecto LinEx Colegios*¹⁰: Es un proyecto educativo español de la región de Extremadura *LinEx Colegios* que nace a

partir de la creación del sistema operativo *LinEx*, que buscó adaptar el entorno tradicional de *LinEx* a las distintas etapas de la educación infantil y primaria.

⁷ <http://www.linuxforkids.org/>

⁸ <http://www.schoolforge.net>

⁹ http://www.educa.madrid.org/web/madrid_linux/

¹⁰ <http://www.educarex.es/linexcolegios/>



e) *El Proyecto argentino Gleducar¹¹*: Es un proyecto educativo, colaborativo y cooperativo, cuyo objetivo fundamental es integrar a las aulas argentinas a las nuevas tecnologías de la información y comunicación, potenciando el uso de SL educativo.

En este proyecto se trabaja básicamente para:

- Concientizar a la comunidad acerca de la importancia de la computadora como herramienta de apoyo pedagógico y como facilitadora en la construcción del conocimiento.
- Promover el uso de Internet en los colegios como herramienta para la investigación y como medio de comunicación de ideas y conocimientos.
- Crear una comunidad para el intercambio de ideas, experiencias y proyectos relacionados la Integración Tecnológica y el uso del Software Abierto.
- Informar a la comunidad en general y a la educativa en particular, sobre los beneficios que ofrecen GNU/Linux y el Software Libre como alternativa respecto de otros sistemas de características similares.
- Asesorar a los colegios y los docentes en el área de Integración Tecnológica.



f) *Proyecto ALBA*: Se trata de un proyecto argentino, que es un sistema de gestión educativa libre denominado Proyecto ALBA patrocinado por la Secretaría de Ciencia, Tecnología e Innovación Productiva. Este es un Sistema Informático Abierto de Gestión Unificada para Unidades Educativas, en su primera

versión que se puede bajar de Internet. Verónica Xhardez¹², es la encargada de la administración y documentación del proyecto y dijo en razón de una entrevista de Canal¹³: *"ALBA está pensado para nuestras propias instituciones educativas a partir de información relevada mediante entrevistas a responsables del sector. Resumidamente, contemplará: Gestión de las Unidades Educativas (establecimientos, ciclos, calendarios, etc.), Gestión de Alumnos (Legajos, seguimientos, consultas, etc.) y Gestión de Docentes (Legajos, horarios, etc.)"*(...) *"El sistema está proyectado para ser utilizado en diferentes niveles de entidades educativas. Esta versión beta fue pensada originalmente para realizar pruebas en escuelas primarias de la Ciudad de Buenos Aires. En lo que respecta a la implementación técnica, una vez liberada su versión estable (o incluso ésta) puede ser bajado del sitio e implementado por cualquier emprendedor o empresa que le interese trabajar el producto."*

4. Software básico disponible libre o de código abierto

Ahora, se verán algunas de las variantes que se pueden utilizar para armar una configuración de programas utilizando software libre, pensado para una PC de un estudiante e inclusive un investigador. Para ello, en la Tabla 5 se resumen las principales

¹¹ <http://wiki.gleducar.org.ar/wiki/index.php/Portada>

¹² <http://www.mastermagazine.info/articulo/11134.php>

¹³ <http://www.canal-ar.com.ar/>

aplicaciones que una computadora actual debería poseer para brindar a sus usuarios las funcionalidades básicas y extendidas para realizar sus tareas:

<i>Tipo de programa</i>	<i>Objetivo</i>	<i>Software disponible en el mercado</i>	
<i>Sistema Operativo</i>	Es un conjunto de programas destinados a permitir la comunicación del usuario y la computadora y gestionar sus recursos. Comienza a trabajar cuando se enciende la computadora y gestiona el hardware de la máquina desde los niveles más básicos.	Free-BSD NetBSD Open-BSD FreeDOS Darwin, Linux	Minix OpenSolaris PcBSD Plan9 Reactos
<i>Herramientas de Ofimáticas</i>	No existe una norma estricta sobre los programas a incluir en una suite de ofimática, pero la mayoría incluyen un procesador de textos y una hoja de cálculo. La suite puede contener un programa de presentaciones, un sistema gestor de base de datos y herramientas menores de gráficos y comunicaciones. También pueden contener: un programa de organización (agenda), un navegador web y un cliente de correo electrónico.	Open Office Gnome Office StarOffice. Papyrus OFFICE	
<i>Herramientas de manejo de imágenes</i>	Permite la edición digital de fotografías y dibujos por medio de la computadora, pudiendo realizar correcciones, trabajo en capas, etc. para mejorar y resaltar todo tipo de imágenes.	GIMP	
<i>Navegadores de Internet</i>	Es un programa o grupo de programas que se utilizan para acceder a contenido HTML disponible en Internet. Puede estar complementado por otras aplicaciones, como gestores de correo electrónico, agendas, etc.	Firefox, Mozilla Mozilla Thunderbird, un cliente de correo electrónico Mozilla Sunbird, un programa de calendario	
<i>Otras aplicaciones</i>	Existen programas específicos para otras necesidades		

Tabla 3: Diferentes software disponibles

4.1. Open Office

Una de las aplicaciones de ofimática de *OSS* es *Open Office*¹⁴ que es un conjunto de herramientas con una interface de usuario muy similar a la que brinda Microsoft Office e incluye varias aplicaciones:

- un procesador de texto: OpenOffice Writer
- un editor de presentaciones: OpenOffice Impress
- un editor de hojas de cálculo: OpenOffice Calc
- un graficador: OpenOffice Graph
- Para las aplicaciones de Dibujo: OpenOffice Draw

OpenOffice originalmente estaba basado en la suite *StarOffice*, desarrollada por StarDivision y adquirida por Sun Microsystems en agosto de 1999. El código fuente de la suite fue liberado en julio de 2000, con la intención de hacer frente al dominio en el mercado de Microsoft Office dando así, una alternativa abierta, de bajo costo y alta calidad y con un código disponible con licencia LGPL¹⁵.

De manera paulatina, pero consistente, esta suite de escritorio fue ganando terreno en el mercado, ayudada por el hecho de que la mayoría de las distribuciones Linux del mercado para PC de escritorio, la traen instalada. En la Figura 1 se observa la pantalla del procesador de textos.

¹⁴ <http://www.openoffice.org>

¹⁵ LGPL es *Lesser General Public License* o *Licencia Pública General Menor*.



Figura 1: Pantalla del procesador de textos.

4.2. Software de manejo de imágenes

GIMP (GNU Image Manipulation Program) es un programa de manejo de imágenes del proyecto GNU que tiene licencia GNU (General Public License). GIMP sirve para procesar gráficos y fotografías digitales con usos típicos que incluyen la creación de gráficos y de logos, cambio de tamaño y recorte de fotografías, cambio de colores, la combinación de imágenes usando un paradigma de capas, la eliminación de elementos no deseados de las imágenes y la conversión entre distintos formatos de imágenes. También se puede utilizar GIMP para crear imágenes animadas sencillas y es la alternativa del software libre al programa de retoque fotográfico Photoshop. La primera versión se desarrolló para sistemas Unix y fue pensada especialmente para GNU/Linux, sin embargo actualmente existen versiones totalmente funcionales para Windows y para Mac OS X.



Figura 2: Pantalla de GIMP.

4.3. Acceso a Internet

Para el acceso a Internet el navegador web es Mozilla o Firefox¹⁶, que fue desarrollado por la Corporación Mozilla y un gran número de voluntarios externos. Firefox, comenzó como un derivado del *Mozilla Application Suite* y terminó por reemplazarlo como el producto bandera del proyecto Mozilla. Firefox es un navegador web multiplataforma, que está disponible para Microsoft Windows, Mac OS X y GNU/Linux. Sin embargo el código ha sido portado a otros sistemas operativos como FreeBSD, OS/2, Solaris, SkyOS, BeOS y más recientemente a Windows.

El código fuente de Firefox está disponible bajo la triple licencia de Mozilla como un programa libre y de código abierto. En la Figura 3 se observa la página de Wikipedia, desde el navegador Mozilla Firefox.



Figura 3: Interfaz del navegador Mozilla Firefox.

4.4. Otras aplicaciones

En la actualidad existen aplicaciones libres de todo tipo que se utilizan para actividades variadas. A continuación se presentan algunas de ellas que pueden asistir a la educación:

- *Arduino*¹⁷: Es una plataforma *Open Source* de computación física basada en una simple placa de Entrada Salida (I/O: Input/Output) y un entorno que emplea el lenguaje de programación Processing/Wiring. Es una herramienta orientada a enriquecer el uso tradicional de la tecnología y el proceso de comunicación que se puede utilizar para crear objetos interactivos autónomos, o para interactuar con otras aplicaciones de software como Flash.
- *Blender*¹⁸: Es un software de código fuente abierto para creación gráfica en 3D: diseño, animación, postproducción y creación interactiva. Actualmente es

¹⁶ <http://www.mozilla.org>

¹⁷ www.arduino.cc

¹⁸ www.blender.org

compatible con todas las versiones de Microsoft Windows, Linux, Solaris, FreeBSD, IRIX y MacOS X.

- *Criptored*¹⁹: En un software de criptografía de libre distribución publicado en el servidor de *CriptoRed* por diferentes autores en el sitio de la Universidad Politécnica de Madrid. El material docente y las aplicaciones para su desarrollo son de distribución libre.
- *EVE* (ellite veejay engine): Es un software de código abierto orientado a la mezcla de video en directo y programado con Pure Data en y para Linux donde toda su música tiene licencia Creative Commons
- *Jahshaka, Powering the New Hollywood*: Es un software de código abierto para postproducción audiovisual digital que permite la edición en tiempo real y la creación de efectos y es compatible con Linux, OsX, Iris, Windows, y Solaris.
- *Red Libre Red Visible*²⁰: Este proyecto permite observar los flujos de información intercambiados mediante redes inalámbricas conectadas a Internet como medio para generar una comunidad para el acceso libre y gratuito a la red.

5. Conclusiones

Existe una gran diversidad de aplicaciones de SL y abierto y el paquete de programas se debe armar de acuerdo a las necesidades de cada proyecto educativo. El SL se puede conseguir en forma gratuita o pagando por su distribución. En el mercado existen diferentes precios y se puede elegir la opción más conveniente y se pueden hacer las copias que se requieran del mismo en diferentes computadoras, sin necesidad de pagar por ellas.

En general *Open Office* resuelve el problema de las herramientas ofimáticas con una interface de usuario muy similar a la que brinda Microsoft Office e incluye aplicaciones similares. Las necesidades de acceso a Internet deben cubrirse con Mozilla Firefox y para otras que surjan deberán buscarse específicamente.

Las condiciones del SL lo tornan ventajoso respecto del software propietario ya que en éste no hay autonomía para usarlo, copiarlo y redistribuirlo porque el usuario no tiene el código fuente y las modificaciones que requiera hacer deben ser solicitadas al propietario del mismo.

6. Bibliografía

Arena, H. F. (2002), *La biblia de Linux, manual de uso, instalación y configuración*, MP Ediciones.

Free Software Foundation (2005), *Definición del software libre*, disponible en: <http://www.fsf.org/licensing/essays/free-sw.html>, consultado el 19 de Mayo de 2005, 18:23.

Open Source initiative (2005), *Definición del software de código abierto*, disponible en: <http://www.opensource.org/docs/definition.php>, consultado el 27 de Mayo de 2005, 20:12.

¹⁹ www.criptored.upm.es

²⁰ www.lalalab.org

Wikipedia (2005), Ubuntu GNU/Linux, disponible en

http://es.wikipedia.org/wiki/Ubuntu_%28Linux%29, consultado el 8 de Junio de 2005, 16:47.

Harari, I. (2006) *Software Libre en las Escuelas*. Facultad de Informática: UNLP.

Stallman, R. (2004) *Porqué las escuelas deberían usar software libre?*

<http://www.gnu.org/philosophy/schools.es.html> traducción del 23 julio de Miguel Abad Pérez

CVs.

Zulma Cataldi y Fernando Salgueiro son docentes investigadores de la Facultad de Ingeniería de la Universidad de Buenos Aires. <http://www.fi.uba.ar/laboratorios/lie/lie.htm>