



# Educalibre: Tecnología y Conocimiento Libre para una mejor Educación

## Abstract

Hoy existe un creciente consenso de que el software libre o el software de código abierto (FOSS, del inglés Free Open Source Software, de ahora en más) puede cumplir un importante rol en la educación. Dicho consenso se desprende de reconocer que los esfuerzos llevados a la fecha para incorporar la revolución de las Tecnologías de la Información (TICs de ahora en más) en la educación todavía es una deuda pendiente a pesar de las cuantiosas inversiones. Deuda que se relaciona con el mínimo impacto de las TICs, a pesar de su amplia masificación, frente a su promesa de mejorar el logro del aprendizaje y promover innovación en las prácticas pedagógicas. El paradigma y modelo de FOSS se eleva como una oportuna y necesaria opción frente al diverso y desigual contexto tecnológico de los establecimientos e instituciones educacionales y su carrera llena de obstáculos que incorporen las TICs a su quehacer. La siguiente ponencia ahonda en las diversas razones que hacen del FOSS la opción preferente en la educación.

## ¿Por qué el software libre es preferible para la educación?

### Porque es educativo

De las cuatro libertades fundamentales que definen al FOSS, y que tiene una relación directa con la misión de una institución educativa, es **la libertad de estudiar, de aprender** el funcionamiento del software.

Contar con software propietario es instar a sus usuarios al consuelo de sólo utilizarlo, porque podemos comprarlos hechos. Existe software que permite ser conocido y compartido, que nos permite saber cómo son realizados, nos permite experimentar y probar. ¿Es que los alumnos y profesores y alumnos se transformarán en "*computines*" de la cultura digital? Evidentemente que no, pero lo importante es **preservar el derecho y la libertad de alumnos y profesores a conocer y compartir el patrimonio tecnológico** según sus propias necesidades y capacidades. Es más, el establecimiento debiese estimular la curiosidad de un estudiante que desea aprender cómo funcionan los programas, y cuando esta se convierte en habilidad, la escuela debe aprovecharla y difundirla, como hace con las virtudes artísticas de sus estudiantes en actos y eventos comunitarios. [1]

Por el contrario, que la escuela sólo dependa de software propietario deja a la institución en una complicada situación: las licencias de los programas prohíben expresamente estudiar su funcionamiento, menos aún de intervenirlo y modificarlo. La curiosidad de los estudiantes frente al software termina en frustración ya que simplemente no tienen derecho ni existen los canales para adquirir el conocimiento que buscan.

Evidentemente el software operativo y el software de productividad de oficina genérico pueden ser utilizados como recursos educativos. Pero ¿debieran estos prohibir dos prácticas fundamentales de la comunidad educativa y necesarios para el aprendizaje: el conocer y el compartir? Ambas prohibiciones impiden que los alumnos puedan llevar el software a sus casas, a su vecindario, que "jueguen" y experimenten con él, que lo investiguen, que indaguen cómo funciona. Conocer por dentro el software entrega la oportunidad de descubrir algo importante: que el software no es solamente algo ajeno, algo que se compra, sino también algo que se aprende, algo que puede ser desarrollado por sus propios compañeros de clase. Que el software es lenguaje, es conocimiento y, por lo tanto, merece ser aprendido y compartido.

Lo anterior cobra vital importancia en los albores de la naciente Sociedad del Conocimiento. Todos los augurios aseveran que las habilidades electrónicas (e-skills) para el manejo de flujos de información y conocimiento en la población es condición habilitante para el desarrollo futuro y competitividad de las naciones. La **técnica cultural del software** es hoy la que controla y regula el flujo de la información en nuestra sociedad, y con ella porciones cada vez más importantes de la interacción entre las personas, las empresas y el Estado. El software es el lenguaje de nuestro tiempo.[\[2\]](#)

La posibilidad de conocer el software, en distintos y crecientes grados de profundidad, permite acuñar el conocimiento de cómo funcionan, y por ende, las habilidades y destrezas para el manejo eficiente, productivo y virtuoso de las herramientas fundamentales de nuestro tiempo. La tan temida "Brecha Digital" no está determinada al acceso a la tecnología o la información, su verdadera amenaza es el abismo creciente que existe entre quienes pueden hacer uso efectivo de dichas herramientas y extraer beneficios, y quienes no saben hacer uso efectivo. El verdadero valor del conocimiento está en la habilidad de utilizarlo, y debemos perseverar para que las nuevas generaciones adquieran dichas habilidades. Más aún cuando sabemos que nuestros niños poseen muy bajas habilidades científicas y matemáticas, y como era de esperar, bajas habilidades informáticas.

Por otro lado, el conocimiento del funcionamiento del software permite asegurar que los futuros profesionales se independicen de un determinado entorno de software. La evolución tecnológica de hoy es tan vertiginosa que debemos ser suficientemente flexibles para acomodarnos a los seguros cambios propiciados para la natural evolución tecnológica de hoy.

Pero si esto es realmente importante, es para el ámbito de la **educación informática**. Una de las características principales de FOSS es que el código fuente está disponible por los usuarios para examinarlo y modificarlo. Esto da a los estudiantes la oportunidad de aprender y estudiar programas de alta calidad y presentes en la vida real. Por el contrario, el software propietario es proporcionado normalmente en la forma binaria y el código fuente rara vez es liberado para usuarios a estudiar. Así, FOSS representa para la educación informática un espacio fértil de estudio y experimentación, en el que no hay límites arbitrarios: cada uno puede elegir por sí mismo cuánto quiere aprender sobre los programas, limitado solamente por su propia capacidad y dedicación. Miles de programas de los que aprender, miles de oportunidades mediante las que participar, desde la misma escuela, en la construcción comunitaria más grande de la que tiene registro la humanidad. *Permite que los estudiantes hagan uso de su creatividad e ingenio tomando software ya existente desarmándolo, modificándolo y rearmándolo, creando mejor software y mejorando el entendimiento de los conceptos de programación usados hoy en día. Lo cual es un valor agregado, ya que los estudiantes de programación pueden trabajar con software ya existente y que puede ser similar al que tengan que usar o crear en el momento en que sean profesionales.*[\[3\]](#)

## Porque es constructivista

No sólo el FOSS es educativo, sino que promueve cierta experiencia de aprendizaje, un experiencia constructivista. El **Constructivismo**[\[4\]](#) es la conjunción de teorías en torno al desarrollo de la psicología educacional que se ha planteado conocer cómo aprenden mejor los seres humanos. El constructivismo entiende el aprendizaje como un proceso en el cual el estudiante construye activamente nuevas ideas o conceptos basados en conocimientos presentes y pasados. En otras palabras, "*el aprendizaje se forma construyendo nuestros propios conocimientos desde nuestras propias experiencias*" (Ormrod, J. E., 2003). Aprender es, por lo tanto, un esfuerzo muy personal por el que los conceptos interiorizados, las reglas y los principios generales puedan consecuentemente ser aplicados en un contexto de mundo real y práctico.

Así, el marco de libertad y cooperación que promueve el modelo FOSS es una suerte de paraíso constructivista para la educación ya que otorga acceso libre a:

- la totalidad conocimiento y su construcción, no solamente el código fuente, sino todo tipo de recursos (manuales, videotutoriales, documentación, etc.)
- a colaborar en comunidades científicas de interés que promueven los proyectos software libre. Cabe recordar que el software libre es la comunidad científica más grande del mundo.
- a construir socialmente según las capacidades, necesidades, interés y especificidades de cada sujeto.[\[5\]](#)

En este marco, las TICs se transforman en recursos que catalizan la innovación educativa relacionándose a una concepción de aprendizaje que apuesta a un aprendizaje profundo y de calidad a través del **proceso social de la construcción de conocimiento**. El constructivismo, columna vertebral de la Reforma Educativa chilena y del mundo, no es posible a través de un marco que promueve el oscurantismo y la represión del conocimiento presente en el software propietario. [\[6\]](#) Por el contrario, el FOSS provee un entorno educacional constructivista que asegura la diseminación abierta del conocimiento y la información, y en el cual los estudiantes de forma participativa:

- trabajan activamente para resolver problemas reales en entornos colaborativos.
- pueden trabajar en entornos que encuentren personalmente desafiantes y motivadores.
- donde los problemas serán reales y lo suficientemente complejos para requerir un amplio rango de habilidades, herramientas y aptitudes.
- necesitarán aprender como trabajar en dichos problemas.[\[7\]](#)

La generación de software y la mejora de los mismos es un proceso incremental y participativo sin pausa, es la verdadera clave para descubrir cómo este paradigma se ha probado tanto o más eficiente que el paradigma industrial-comercial. Optar por el software libre permitiría a nuestros alumnos acceder a la punta de lanza de la innovación tecnológica mundial. Optar por el software libre permitiría a nuestros alumnos acceder a la comunidad científica más grande del mundo. Un ejemplo sintomático: la dirección del desarrollo del "kernel" (núcleo) de Linux, sistema operativo libre y junto a GNU e Internet, uno de los pilares de la revolución tecnológica del software libre, fue traspasada desde su creador Linus Torvalds a Marcelo Tosatti, joven brasileño de entonces 18 años.

## Porque es ético

El movimiento de Open Source se ha opuesto al de Software Libre básicamente porque el primero intenta depurar los argumentos éticos y morales del desarrollo tecnológico. Puede ser que esto sea positivo para el desarrollo de una industria determinada, pero el proceso

educativo no puede desprenderse los aspectos éticos de la tecnología ni el conocimiento.

Tal como reflexiona el ex-subSecretario de Educación, José Weinstein: *la alfabetización tecnológica, si bien contempla en uno de sus aspectos la adquisición de lenguajes técnicos asociados a productos específicos, no puede ni debe limitarse a una visión tecnocéntrica. Debemos desarrollar una noción de finalidad del acto creativo tecnológico, es decir reflexionar acerca de los efectos que este acto tiene sobre las personas y el medio ambiente y el riesgo que a veces supone para su integridad*[\[8\]](#). No basta recibir un adiestramiento procedimental de algún software de una sola empresa para lograr una alfabetización digital que permita seguir accediendo a niveles más complejos y virtuosos de uso de la tecnología. Menos aún para nuestros niños, cuyas prácticas son verdaderamente impredecibles frente a tan vertiginoso desarrollo que genera cambios dramáticos en la cultura entre generaciones.

El FOSS dota a los actos tecnológicos de un cariz ético, completamente ausente en la perspectiva puramente técnica-utilitaria. Desarrolla las habilidades y competencias que propenden a una educación integradora que forme ciudadanos con un pensamiento crítico y flexible; que sean creativos, innovadores, emprendedores; dispuestos a apropiarse del conocimiento y descubrir las preguntas pertinentes que éste concita; que valoren la creación de riqueza, tanto productiva como social. El mismísimo Richard Stallman así también lo acredita: *"la escuela debe practicar lo que pregona: todo el software instalado por la escuela debe estar disponible para que sea copiado por los estudiantes, para llevar a casa y luego ser redistribuido. Enseñar a los estudiantes a usar software libre, y participar en la comunidad del software libre es una lección cívica. También enseña a los estudiantes el papel del modelo de servicio público, en vez del de magnates."*[\[9\]](#)

Puede ser que algunos creen que el rol de la escuela es formar para el trabajo y la universidad. De ser así, si la misión de la escuela es entrenar trabajadores sumisos y baratos para mejorar la rentabilidad de las empresas. Pero si la idea es educar a ciudadanos libres, conscientes de sus derechos y responsabilidades, capaces de cuestionar la verdad establecida, de apreciar el arte, de imaginar el mundo que desean y aportar a su concreción, entonces es ineludible usar Software Libre: programas que los estudiantes y educadores pueden usar, estudiar, modificar y distribuir a su antojo. Para entrenar, cualquier software sirve, pero para educar, sin dudas es preferible el FOSS, ya que fomenta valores y actitudes positivas, que se basan en el mérito personal y la solidaridad social. Actitudes como:

- que no todo está hecho y que aún hay retos
- que las cosas siempre se pueden mejorar.
- adoptar una postura constructiva.
- cooperar con la comunidad local e internacional, sin distinción de edades, razas, nivel social, títulos, etc.
- que hay muchas formas de acercarse a la verdad/perfección y que cada grupo o persona aporta la suya.
- que lo mejor para cada quién/comunidad se escoge libremente de acuerdo a las necesidades/expectativas de cada sujeto o grupo social, y no debe ser impuesto por casas comerciales o estándares externos.
- la posibilidad de aprender de otros y que otros pueden aprender de nosotros. Nadie es todopoderoso o autosuficiente por completo
- propagar el conocimiento de forma libre (básicamente, lo que se trata de hacer cuando uno da una clase).
- la no discriminación.
- trabajar en equipo de forma colaborativa.
- la libertad de investigar, crear, modificar y aprender.[\[10\]](#)

Es menester que nuestros niños y jóvenes tengan acceso a la innovación tecnológica para que la realización del acto tecnológico creativo permita generar usos más complejos, productivos y virtuosos. Mientras más disminuye la edad, más se osterga el acceso a la innovación tecnológica, respecto del mundo adulto (lo mismo ocurre con la tercera edad). FOSS ofrece a nuestros niños y jóvenes un espacio para liberar sus inquietudes y desarrollar sus habilidades sin que sean discriminados por su edad, status social, raza o cultura. Basado únicamente en la meritocracia, el software libre se desarrolla a partir de la colaboración y la cooperación mutua. El espacio virtual se organiza de forma donde cada cual puede realizar su mejor aporte, según sus competencias y capacidades, desde la programación hasta el reporte de errores o traducir.

En suma, el contexto de aprendizaje que propone el FOSS no sólo permite integrar el acceso inclusivo y desarrollo constructivo de conocimientos siempre disponibles para cualquier uso, sino también formar habilidades y destrezas cognitivas, digitales y procedimentales para el trabajo en red y fomentar actitudes positivas que equilibran el desarrollo individual y el social. Vale decir, un contexto educativo que forma **competencias**, la integración efectiva en el hacer de distintos tipos de saber para desempeñarse exitosamente en la economía del conocimiento. El trabajador "competente" en diversos dominios y actitudes transversales tendrá mayor valor futuro, haciendo del aprendizaje por competencias el paeradigma principal del aprendizaje continuo (long-life learning).

## Porque es legal

El uso indiscriminado de software no-licenciado en la educación es una realidad indesmentible. Las instituciones educativas que no pueden asumir el pago de licencias recurren a copias ilegales de software propietario. Esta práctica común en la educación es un hecho incoherente con su misión: la escuela no puede desprestigiar la ley ni promover ningún actos delincuenciales.

Por el contrario, con FOSS, licenciado de forma pública y por ello no viola leyes de propiedad intelectual, las instituciones educativas pueden utilizar todas las copias de software que requieran, sin importar la cantidad de copias ni el tipo de uso (académicos, administrativos, comercial, etc.). Lo anterior permitiría regularizar la flagrante situación legal de uso de software en las instituciones educativas.

El uso de FOSS desalienta también la piratería en estudiantes, muchos de quienes no pueden acceder a la compra de copias licenciadas de software propietario. Si software propietario es utilizado para enseñar, los estudiantes no tienen otra opción que utilizar copias ilegales del software para hacer los deberes y las tareas en su hogar o fuera de la institución. Utilizar software privativo para enseñar, es incitar a los alumnos a cometer un delito. Aunque alguna empresa de software propietario ofrezca regalar su software a la institución, se debiera consultar: ¿también para todos los profesores y alumnos?, ¿aseguras que será gratis ahora y en el futuro?, ¿aunque los alumnos dejen de ser alumnos y empiecen a usar para su trabajo? Por contraste, no hay restricción alguna de hacer copias de FOSS dentro y fuera de los establecimientos educacionales.

## Porque es diverso

El aporte del patrimonio tecnológico del FOSS es enorme, en cantidad y tipos de software. SourceForge, la maestranza más grande de FOSS, tiene incubado cerca de 160 mil proyectos!!! Junto con esta tremenda diversidad es usual encontrar varios tipos de software para un mismo fin lo que permite mayor libertad de opción, según los objetivos de aprendizaje que se buscan lograr.

A su vez, podemos encontrar numeroso software [para cada área curricular y las más diversas disciplinas curriculares y extracurriculares](#). En distribuciones Linux construidas especialmente para el escritorio educativo vienen pre-instaladas con las aplicaciones educaivas más recurrentes. En el caso de [Debian-Edu](#), esta contiene 80 aplicaciones de caracter educativo en su última versión DVD. Esta diversidad en cantidad y calidad es un verdadero aporte en especial, en algunas disciplinas y niveles educativos. Asunto relevante para nuestras instituciones que están cubiertas de software genérico-ofimático, pero donde hay mucha carestía de software específico, en especial, software educativo. Por ejemplo, existe una grave carestía de software en algunos niveles como la educación pre-escolar y la básica, normalmente de muy alto precio debido a su producción audiovisual y/o su complejidad-asertividad.

Cabe destacar que dicha variedad se hace cargo de la infinita diversidad de usuarios finales como la diferencia de edad, habilidad tecnológica, nivel educativo, entre otros. No existen algunas aplicaciones claves, por ejemplo, para un niño de 5 años. De seguro, este niño estará más motivado a aprender a usar un procesador de textos con [TuxType](#) o dibujar con [TuxPaint](#).

A su vez, debido a la posibilidad de intervención, dicho software puede ser adaptado a las necesidades educativas o formativas que pretenda cubrir.

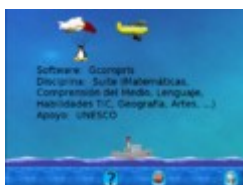
Es cierto que mucho software FOSS no está a la altura de la calidad de mucho software propietario, pero primero, eso puede ser irrelevante y a veces favorable para la calidad del proceso educativo, y segundo, dicha brecha se acorta cada día más.

No sólo hay diversidad de software para apoyar el aprendizaje, también hay excelente software para labores de gestión del aprendizaje (personalización del aprender). La plataforma de aprendizaje electrónico [Moodle](#), por ejemplo, se ha transformado en un referente de la industria del e-learning. También hay tareas de caracter admisnitrativo orientadas a las necesidades educativas (informes de notas, gestión de matrícula, horarios de clases, etc.)

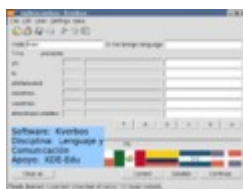
Una pequeña muestra:



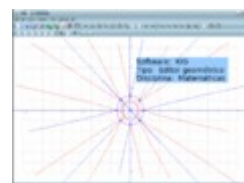
**TuxPaint**, programa de dibujo orientado a niños entre 3 y 8 años de edad.



**GCompris**, suite educativa para niños en diversas materias, apoyada por UNESCO.



**KVerbos**, herramienta de lecto-escritura para geométrico, que conjugación de verbos puede reemplazar a Cabrí.



**ChemTool**, editor de estructuras moleculares, para Química.



**GRASS**, sistema de información geográfica (SIG).



**Moodle**, plataforma de gestión del aprendizaje electrónico (LMS).



**Alba**, sistema argentino de gestión administrativa para

## Porque es económico

Uno de los principales asuntos que deben tener en cuenta los diseñadores de políticas respecto del uso de TICs en la educación son el costo. El costo de proporcionar infraestructura comunicacional, computacional y de hardware, como el software necesario intimida a los sectores más necesitados.

FOSS puede bajar las barreras de acceso y escalamiento de TICs reduciendo el costo del software. El costo inicial de la adquisición de FOSS es insignificante, ya que no hay pago de licencias de uso. Es generalmente posible descargar FOSS sin costo alguno. Si hay limitada conectividad o ancho de banda, puede ser más conveniente conseguir el software en un CD-ROM con una tarifa nominal. Pero no hay costo de licencias para cada usuario o computadora y puede ser distribuido libremente una vez que la copia es descargada o dispuesta en un CD-ROM. De ahí, el costo inicial de adquirir FOSS es mucho menor que el costo de adquirir software propietario, que además licencia por cada usuario o computadora. Las actualizaciones de FOSS pueden ser obtenidos generalmente de una manera semejante, cuyo costo también sea insignificante. Frente a la obligación del Estado de distribuir software a los colegios, el FOSS permite un enorme ahorro de recursos públicos.

No debemos pensar que FOSS es gratuito. Su acceso es gratuito pero debemos contemplar la implementación del software, que se lleva gran parte del costo total en la infraestructura tecnológica. Normalmente las licencias de uso cubren 20 a 40% del costo total de implementación del software. Aún así existe evidencia empírica que el Costo Total de Propiedad (TCO) del software, metodología desarrollada por la propia industria TIC, los ahorros oscilan entre un 20 y 60% menos, dependiendo el tipo de escuela y sus capacidades<sup>[11]</sup>:

## BECTA (2005), Open source software in schools: A study spectrum of use and related ICT infrastructure costs

Annual TCO per PC (£)	OSS primary	Non-OSS primary	OSS secondary	Non-OSS secondary	OSS as % non-OSS primary	OSS as % non-OSS secondary
Hardware	131.71	280.53	151.93	221.88	47	68
Software	44.68	64.14	32.56	67.10	70	49
Network	14.11	66.94	61.58	56.76	21	108
Consumables	21.89	53.13	24.21	28.33	41	85
Training	57.50	53.31	27.04	11.53	108	235
Formal support	273.17	406.16	206.61	385.62	67	54
Self-support	148.86	303.83	283.39	264.48	49	107
<b>TOTAL</b>	<b>691.92</b>	<b>1228.04</b>	<b>787.32</b>	<b>1035.70</b>	<b>56</b>	<b>76</b>

Table 4: Comparative TCO per PC for OSS and non-OSS schools, by cost category for primary and secondary schools

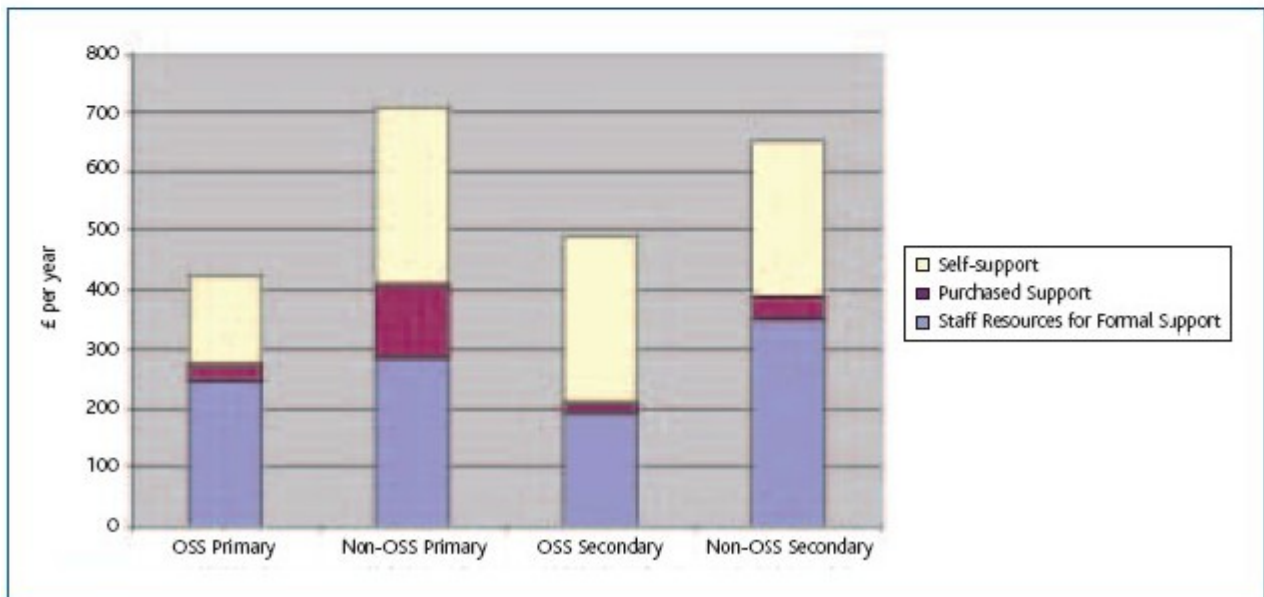


Figure 5: Average annual support costs

Otro punto a tener en cuenta, es que el costo de mantenimiento y soporte del software libre es menor debido a su gran estabilidad y calidad que evita tener que reinstalar y mantener los equipos en lapsos cortos de tiempo. Debido a la escasez de capacidades tecnológicas en los colegios, el soporte es siempre un dolor de cabeza que representa un alto costo para los menguados presupuestos tecnológicos de las instituciones educativas.

El FOSS puede ser abordado a través de la participación en comunidades de interés que apoyan voluntariamente la diseminación del conocimiento teórico y práctico del software libre. Además la interacción con estas comunidades de aprendizaje informal permiten desarrollar capacidades tecnológicas donde hay pocas, lo que promueve independencia y evita el **vendor lock-in**, dependencia extrema del proveedor comercial. Lo importante es el destino la inversión tecnológica: ¿es mejor para un establecimiento educacional invertir



en licencias a una empresa extranjera o capacitar debidamente a sus cuadros técnicos? Aún así, si no hay capacidades técnicas que permitan instalar software libre, se puede recurrir a usos no invasivos del hardware como un Live-CD, sistemas abiertos que no necesitan ser instalados en la memoria física y corren desde la memoria virtual.

El número insuficiente de computadores, una de las falencias de la infraestructura TIC, es un asunto que será crucial a futuro, debido a los niveles crecientes de cobertura necesarios. Los niveles de inversión requeridos para aumentar la infraestructura existente es alto, tanto a nivel público como privado, a menos que se pueda potenciar lo ya existente y encontrar vías alternativas a la adquisición unipersonal de computadores. Esta situación se opone al modelo comercial del software propietario, que vende licencias por computador. El sistema operativo Linux en su infinita variedad posee versiones ligeras que pueden hacer correr de forma eficiente computadores de segunda mano u obsoletos. Además, el proyecto [Linux Terminal Server \(LTSP\)](#) permite generar soluciones para reutilizar hardware antiguo como clientes (terminales tontos) conectados a un servidor central. Existen experiencias donde un servidor puede mantener operativo a 40 terminales. Por último, el [proyecto MultiHead](#) permite tener una CPU pero con posibilidad conectar periféricos (pantalla, teclado, ratón) para 4 estaciones de trabajo. Pues, no bote a la basura su computador viejo!!! dónelo a la escuela para que puedan reutilizarlo usando FOSS.

Cabe destacar que el FOSS basado en estándares abiertos y compartidos siempre tiene una vocación de interoperabilidad. FOSS no tiene ningún problema de co-existir con el software propietario, mientras este último normalmente hace lo contrario para forzar posicionamiento y propiciar dependencia. Aún así la interoperabilidad entre formatos ha mejorado sustancialmente en el último tiempo.

## Porque es robusto

El costo más bajo no es la única razón de por qué el uso de FOSS en servidores es predominante. El FOSS es infraestructura considerada de mejor fiabilidad, el desempeño y seguridad. Los administradores de instituciones educativas deben tomar éstos aspectos en cuenta a la hora de tomar decisiones en la infraestructura de TICs de sus instituciones, especialmente importante para las más grandes.

La metodología del desarrollo de FOSS tiende a asegurar de alta calidad del software. Los errores del software ("bugs", bichos) son arreglados rápidamente con la ayuda de muchos desarrolladores y usuarios, por lo que el software es depurado, lo que redundará en ser más fiable. Por ejemplo, en un análisis cuantitativo de software de base de datos fue encontrado que la base de datos de FOSS MySQL tiene seis veces menos defectos que las bases de datos propietarias. Otros estudios serios sugieren también revelan que el FOSS se desempeña mejor y es más seguro que las contrapartes propietarias.

Estas mejoras son posibles en gran medida dado que el código fuente está a disposición para mejorarlo. Este paradigma de desarrollo abierto es sin lugar a dudas la tendencia más innovadora en el desarrollo de la industria TICs. La innovación del FOSS es continua y permanente, y no depende de otro criterio que no sea la calidad del producto.

## Referencias

### Español

- <http://www.educalibre.cl/node/11>

- [http://es.wikibooks.org/wiki/El\\_software\\_libre\\_en\\_la\\_educaci%C3%B3n](http://es.wikibooks.org/wiki/El_software_libre_en_la_educaci%C3%B3n)
- [http://wiki.gleducar.org.ar/wiki/Razones\\_para\\_usar\\_software\\_libre\\_en\\_educaci%C3%B3n](http://wiki.gleducar.org.ar/wiki/Razones_para_usar_software_libre_en_educaci%C3%B3n)
- <http://www.gnu.org/education/education.es.html>
- <http://www.cignux.org.ar/pablo.htm>
- <http://www.colewebs.org/spip.php?article46>
- [http://www.educacionenvalores.org/article.php3?id\\_article=305](http://www.educacionenvalores.org/article.php3?id_article=305)
- <http://eloy sanz.com/SoftLibre/ponencia/ponencia.htm>
- [http://z-labs.com.ar/docs/tif/9-sl\\_educacion.html](http://z-labs.com.ar/docs/tif/9-sl_educacion.html)
- <http://www.educalibre.cl/node/623>

## Inglés

- [http://en.wikibooks.org/wiki/FOSS\\_Education](http://en.wikibooks.org/wiki/FOSS_Education)
- <http://www.iaa.upf.es/~xamat/FreeSoftware/FreeSoftwareEducation/>
- <http://www.offset.org/>