II Conferencia Internacional sobre Brecha Digital e Inclusión Social
(Leganés, Madrid, del 28-30 de octubre de 2009)

¿LIBERTAD O PRAGMATISMO?: EL PAPEL DEL SOFTWARE LIBRE EN LA ALFABETIZACIÓN TECNOLÓGICA

Daniel Martínez Ávila
Departamento de Biblioteconomía y Documentación, Universidad Carlos III de Madrid.
dmartine@bib.uc3m.es
Oscar Prieto Gordo
Departamento de Ingeniería y Arquitecturas Telemáticas, Universidad Politécnica de Madrid
oprieto@diatel.upm.es

RESUMEN: En el estudio se revisan los conceptos de alfabetización tecnológica, software libre y open source. Se analizan los requisitos necesarios para una correcta alfabetización tecnológica, y se ponen en relación con las características filosóficas de los distintos tipos de software, el concepto de libertad del software libre y la visión pragmática del open source. Se contemplan las implicaciones del software libre respecto a la Administración y la disminución de la brecha digital, y se pone especial énfasis en las ventajas para la economía y la educación. Por último, se repasan proyectos concretos de alfabetización con software libre, analizándose el grado de libertad del software utilizado.

PALABRAS CLAVE: Software libre, Open Source, Alfabetización Tecnológica, libertad, Pragmatismo, GNU/Linux

1. Introducción

Los términos software libre y open source, pese a considerarse sinónimos y ser generalmente equivalentes, encubren importantes diferencias filosóficas que se plasman en su filosofía y sus fundamentos. Un tercer tipo de software es aquel llamado propietario (o privativo), el cual se encuentra diametralmente opuesto por sus características tanto al software libre como al open source. Si las iniciativas de alfabetización tecnológica tienen su raíz en decisiones políticas, el movimiento del software libre se trataría sin duda de uno de los más firmes grupos de presión social, el software open source de la visión más liberal y pragmática, y el software propietario una visión monopolística y colonial. Por lo tanto, el tipo de software que se utilice en los proyectos de alfabetización, tendrá implicaciones tanto políticas como económicas para sus fundamentos y resultados.

2. Objeto y metodología

Objetivo de estudio

El objetivo del estudio es determinar las características necesarias para una adecuada alfabetización tecnológica, así ponerlas en relación con las características e implicaciones que tienen los distintos tipos de software para los proyectos de alfabetización.

Metodología

Para llevar a cabo el estudio, se ha realizado una revisión conceptual de los términos alfabetización tecnológica, software libre y open source, poniendo especial énfasis en sus implicaciones para la educación y la brecha digital, y sus ventajas económicas para la Administración. Se han analizado algunas iniciativas concretas, como el uso de software libre en Brasil por la SERPRO, la migración a Ubuntu por la policía francesa, el proyecto gnuLinEx en Extremadura, el proyecto One Laptop Per Child (OLPC) de Nicholas

Negroponte, o la reciente propuesta de Zapatero para repartir portátiles entre los alumnos de primaria en España. Para todas las iniciativas se ha tratado de determinar que tipo de software se utiliza, y las implicaciones que tiene para la alfabetización tecnológica y la economía de la Administración. Por último, también se ha tratado de recoger las opiniones y posturas de algunos de los principales actores del movimiento software libre/open source.

3. Concepto de alfabetización tecnológica

El concepto de alfabetización tecnológica, tal como establece Richard W. Budd (Budd 1997), consiste en "la habilidad para buscar, encontrar, ordenar, categorizar y organizar información para el uso personal y profesional. Cómo conseguir el acceso y el uso de Internet y otros bancos de datos en línea relevantes". La alfabetización tecnológica, junto a los conceptos de alfabetización visual (comprensión de las imágenes que nos rodean), alfabetización organizativa (comprensión de dinámicas de grupos sociales y profesionales), alfabetización "mediática" (comprensión de la influencia de los medios en el mundo actual), y alfabetización cultural (comprensión de los símbolos que construyen las identidades culturales), compone el concepto más amplio denominado alfabetización informativa o Alfabetización en Información. Tal como indican Miguel Ángel Marzal et al. (Marzal García Quismondo, Miguel Ángel et al. 2005), Alfabetización en Información hace alusión a un nuevo modo de conocer en entornos tecnológicos una vez adquiridas la competencia en las cinco alfabetizaciones establecidas por Budd. Tanto en las definiciones de Budd como de Marzal se denota un énfasis en el carácter de habilidad adquirida y competencial que se debe suponer con la alfabetización tecnológica ¿pero en qué consisten y qué connotaciones tienen esas competencias?

Una revisión terminológica del concepto de Alfabetización en Información realizada por David Bawden (Bawden 2002) pone en relación a los conceptos de alfabetización informática, alfabetización en Tecnologías de la Información, y alfabetización electrónica, agrupándolos dentro de una misma categoría de "Alfabetizaciones basadas en destrezas". Según el Chambers English Dictionary, una definición de alfabetizado en Informática (computer literate) es "competente en el uso de ordenadores". Bawden a esto añade:

"Esta simple definición lexicográfica oculta el hecho de que hay un espectro de puntos de vista en relación con lo que 'competencia' implica, más o menos como ocurre con la variabilidad de la definición de alfabetización per se. Lo más común ha sido un enfoque pragmático basado en destrezas. Un ejemplo típico es el programa de formación en alfabetización informática de la Royal Society of Arts en Gran Bretaña (Attwood et al., 1993)*, que define como su objetivo 'la acreditación de aquellas destrezas prácticas en tecnologías de la información necesarias para el trabajo, y, sin duda, para la vida diaria'. En la práctica, esto se traduce en una introducción en aquellas destrezas que se requieren para poner en marcha un conjunto de paquetes de aplicaciones informáticas".*

Del anterior párrafo deben resaltarse las palabras de Bawden cuando dice "un enfoque pragmático basado en destrezas" y lo que la Royal Society of Arts llama "destrezas prácticas en tecnologías de la información necesarias para el trabajo y para la vida diaria". Definiciones similares a la de la RSA son dadas por Hunter (Hunter 1983), Husen y Postlethwaite (Husen et al. 1985), o Haigh (Haigh 1983), todas ellas con el denominador común de la adquisición de conocimientos y destrezas necesarias para el desarrollo diario. Cabe preguntarse, entonces, qué tipo de software, de entre todos los tipos existentes, será capaz de permitir la adquisición de este tipo de destrezas necesarias para la vida, y hasta dónde llega o debe llegar ese pragmatismo del que habla Bawden, así como qué implicaciones tiene en el proceso de adquisición de destrezas.

La revisión conceptual de Bawden también repasa diferentes autores y puntos de vista sobre el concepto de alfabetización informática. Estos puntos de vista abarcan desde el entrenamiento concreto en los sistemas de Tecnologías de la Información existentes (Ovens 1991), lo cual implicaría una total dependencia del individuo

de dichas tecnologías y herramientas, hasta lo que Tucket llama "autoconfianza" en el manejo de ordenadores:

"Tuckett (Tuckett 1989) *considera que la alfabetización informática consta de tres componentes. Los dos primeros ya se han mencionado: un conocimiento general acerca de qué pueden hacer los ordenadores, y las destrezas necesarias para utilizarlos como una herramienta eficaz. Él añade un tercer componente: la muestra de autoconfianza en el manejo de ordenadores. El autor argumenta que, incluso si alguien utiliza un ordenador para llevar a cabo tareas útiles todos los días, y tiene una noción de lo que, en su totalidad, puede incluirse en las posibilidades de uso, no pueden llamarse, sin embargo, competentes en este sentido si dependen completamente de que otros les ofrezcan las instrucciones para el uso del ordenador, o les ayuden si algo va mal. Este elemento de autoconfianza aparece también por toda la literatura sobre AI, aunque a menudo de forma más implícita que explícita".*

Para poder hacer efectiva la autoconfianza, Tuckett enfatiza que es indispensable desprenderse de la dependencia de terceros en el uso del ordenador, así como cuando sea necesario modificar todos los factores pertinentes (por ejemplo el código) en el caso de que algo vaya mal. El alfabetizado debe ser capaz de poder adaptar la tecnología a sus necesidades concretas, con total confianza de que dispone de los medios y sus acciones no incumplen las leyes, y sin depender de las restricciones impuestas por terceros. El alfabetizado en Informática debe ser capaz de elegir las herramientas que quiere utilizar y adaptarlas en función de la comunidad a la que pertenece, beneficiando y beneficiándose de las mejoras que se aporten desde esa comunidad, y sin depender de las características y funcionalidades cerradas que las empresas desarrolladoras de tecnologías impongan. En definitiva, si en la alfabetización informática la independencia de terceros y la autoconfianza son componentes indispensables, todo software que no permita la libertad de uso, distribución, de modificación y de distribución de las modificaciones del mismo no tendrá cabida dentro del concepto de alfabetización informática, y, como se verá más adelante, el software que cumple con estas características es el software libre.

Siguiendo en la línea de Tuckett, otros autores como Shapiro y Hughes (Shapiro et al. 1996) amplían el concepto de alfabetización informática en función de un programa basado en otras siete alfabetizaciones: alfabetización en herramientas (conocimiento y uso de las herramientas dentro de las Tecnologías de la Información, incluyendo el hardware y el software), alfabetización en recursos (conocimiento de los sistemas de acceso a los recursos de información), alfabetización socio-estructural (comprensión de la situación social y de producción de la información), alfabetización investigadora (uso de las herramientas de las Tecnologías de la Información aplicadas a la investigación y el trabajo académico), alfabetización para la publicación (habilidad para difundir y publicar información), alfabetización en las tecnologías incipientes (capacidad para comprender las innovaciones en Tecnologías de la Información y la tomar decisiones con respecto a las nuevas tecnologías), y alfabetización crítica (capacidad para evaluar de forma crítica los beneficios y costes de las tecnologías de la información). Respecto a la alfabetización en herramientas, se hace necesario un mecanismo dentro de las Tecnologías de la Información que garantice la posibilidad de conocimiento del software y hardware por la comunidad. Un software cuyo código no esté disponible para su estudio, o un hardware cuyos controladores y funcionamiento sean secretos, no es adecuado para la alfabetización informática ya que no fomenta la independencia y la autocapacitación. Tal como indica el profesor Jesús González Barahona (González Barahona 2002c), en un contexto de enseñanza universitaria:

"Si se usan programas libres, el alumno puede reproducir todo el entorno de prácticas, con total exactitud, en cualquier otro ordenador. En particular, por ejemplo, en el ordenador de su casa, donde podría practicar cómodamente. Y todo esto, naturalmente, sin ningún problema de licencias, y sin costes extra para el alumno. […] En el caso de que la enseñanza sea para informáticos, para gente que puede entender (y tiene que entender) las interioridades de las herramientas, la disposición del código fuente es fundamental. Esto permite, con gran facilidad y sin problemas de licencias ni acuerdos especiales con los fabricantes, ver cómo están hechas algunas herramientas reales, de calidad comercial. Y de esta forma, enseñar con el ejemplo, que es una de las mejores formas de enseñar informática".

Por lo tanto, se pone de manifiesto que para ejercer una alfabetización informática de forma efectiva se debe disponer del código fuente y de la posibilidad legal de disfrutar de ciertas libertades relacionadas con la modificación y distribución del código. Desde un punto de vista práctico, la manera más eficiente de adquirir destrezas relacionadas con la tecnología de forma independiente, es con una disposición abierta del código, la cual es una de las características del software libre. Sin embargo, y tal como concluye Bawden en su revisión conceptual:

"Todas estas alfabetizaciones basadas en destrezas emergieron para responder a las necesidades de un entorno informacional más complejo, con nuevas tecnologías, y una mayor variedad de medios de comunicación y de servicios; formas de alfabetización que, centradas en torno a un núcleo de destrezas, y extendiéndose más allá de éstas, muestran que, como la alfabetización misma, requieren de un amplio espectro de habilidades, conocimiento, concienciación, y actitudes. Esto nos lleva a considerar una forma de alfabetización que irrumpe basándose en premisas quizá más amplias que lo que una o dos destrezas pueda suponer: la Alfabetización en Información".

Esto significa que, el enfoque pragmático basado en la adquisición de destrezas concretas, tampoco es el único punto de vista existente sobre alfabetización tecnológica, sino que puede y debe formar parte de un concepto más amplio, o incluso enfrentado, en el que se tengan en cuenta otras consideraciones y aspectos de índole filosófica o social. En el plano del software libre, este debate quizás se sitúe más allá de la mera disposición abierta y pragmática del código. Estos matices, tal como se comprobará a continuación, podrán verse reflejados en las diferentes filosofías del movimiento del software libre y open source, dos caras de una misma moneda que, tal como se ha indicado anteriormente, se presenta como la manera más viable para llevar a cabo la alfabetización informática dentro de un espectro más amplio como es el concepto de Alfabetización en Información.

4. Software libre y open source: libertad o pragmatismo

El software libre, como movimiento, surgió en la primera mitad de los años 1980, anteriormente a ello el software siempre había sido libre como tal, sin necesidad ni siquiera haber sido definido. El modelo de desarrollo informático, en los años 1960 y 1970, estaba basado en la disponibilidad del código y la libertad para adaptar y redistribuir los programas en función de las necesidades de hardware de sus desarrolladores, la práctica habitual entre las comunidades desarrolladoras era la de compartir y la formación y aprendizaje a través del estudio del código fuente. Sin embargo, a principios de los años 1980, este modelo informático que había funcionado correctamente entró en una profunda crisis debido a una mercantilista maniobra empresarial. Desde algunos ámbitos, las empresas empezaron a comercializar su software de forma independiente y cerrada, ocultando el código fuente y distribuyendo únicamente el código binario, inútil para su estudio y modificación por terceros. Esta práctica privativa poco a poco fue emulada por cada vez más empresas y laboratorios de investigación, que obligaron a su personal a firmar restrictivas cláusulas que impedía compartir el código y ayudar a sus semejantes, por considerarlo una violación de su ventaja competitiva, sin caer en la cuenta que esta ventaja competitiva siempre se había adquirido de forma colaborativa. Es en este escenario, cuando Richard M. Stallman, reconocido hacker del Laboratorio de Inteligencia Artificial del Massachussets Institute of Technology (MIT) decide fundar en 1984 el Proyecto GNU (acrónimo de GNU's Not UNIX) y en 1985 la Free Software Foundation (FSF), estableciendo con las bases éticas del movimiento del software libre y abandonando su puesto en el MIT.

Una definición de software libre formalizada por la FSF (Anon.) es aquella en la que se cumplen las cuatro libertades:

- La libertad de ejecutar el programa, para cualquier propósito (libertad 0).
- La libertad de estudiar cómo trabaja el programa, y adaptarlo a sus necesidades (libertad 1). El acceso al código fuente es una condición necesaria.

- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- La libertad de mejorar el programa y publicar sus mejoras, y versiones modificadas en general, para que se beneficie toda la comunidad (libertad 3). El acceso al código fuente es una condición necesaria.

Estas cuatro libertades se convierten en puntos fundamentales a la hora de ejercer una dinámica de desarrollo similar a la de los años 1960 y 1970, época en los que se impulsaron algunos de los avances tecnológicos más fundamentales para el desarrollo de internet. De la misma manera, estas libertades se convierten en fundamentales para la adquisición de competencias necesarias para la alfabetización en un entorno no sólo profesional sino también educativo. En la definición ampliada de software libre, también se pone un especial énfasis en las connotaciones y ventajas filosóficas del concepto, explicitando, por ejemplo, que "el software libre es una cuestión de libertad, no de precio" (ya que nadie debería impedir que se distribuyan las modificaciones gratis o a cambio de dinero), y recomendando el uso de licencias copyleft, concepto ideado también por Richard Stallman, como el mecanismo legal más adecuado para proteger estas libertades. Uno de los ejemplos más extendidos de licencias copyleft es la licencia GNU GPL (General Public License), cuya característica más relevante es su naturaleza vírica que protege legalmente la concesión de derechos tanto del software que la aplica como de sus modificaciones redistribuidas.

Como resultado de todos los esfuerzos del Proyecto GNU y la Free Software Foundation (así como de todos los programadores independientes alrededor del mundo que participaron, incluyendo la inestimable aportación de un kernel por parte de un estudiante finlandés llamado Linus Torvalds) surge a principios de los años 1990 el sistema operativo GNU/Linux, ejemplo paradigmático de éxito en el desarrollo del mundo del software libre y cabeza más visible para la adopción de medidas concretas en políticas de crecimiento económico y alfabetización en información.

La calidad de los proyectos de software libre, y tomando como ejemplo el sistema operativo GNU/Linux, es algo que a día de hoy está fuera de toda duda. Existen numerosos estudios que ponen de manifiesto las ventajas técnicas y superioridad del software libre sobre el propietario, por ejemplo, el autor francés Emmanuel Cornet, en su página web "Why Linux is better" (Anon.), sintetiza las ventajas de GNU/Linux sobre el sistema operativo propietario Microsoft Windows en las siguientes: menor riesgo de adquisición de virus, mayor estabilidad del sistema, mayor protección, posibilidad de obtenerlo a menor precio de forma legal (o incluso gratis), mayor libertad e independencia, mayor cantidad de software útil preinstalado, no hace falta instalar controladores separados (todos se encuentran en el kernel Linux), mayor facilidad de actualización, mayor existencia de software equivalente gratuito y legal, mayor facilidad de localización de software a través del sistema, mejores entornos de escritorios gratuitos y con menos requerimientos de hardware, no existe las fragmentación de disco, mayor variedad de gestores de ventanas, mayor rapidez, mayor ecologismo debido a una disminución de los soportes físicos y empaquetado, desaparición de las puertas traseras del software (o capacidad de detectarlas en el código), soporte gratuito e ilimitado, posibilidad de utilizar espacios de trabajo en lugar de ventanas, mayor facilidad de informar y solucionar bugs, menor necesidad de reiniciación, menor requerimiento de hardware, mayor cantidad de juegos gratuitos, mejor reparto de la riqueza (al poder beneficiar a una gran cantidad de empresas medianas y pequeñas que proporcionen servicios en lugar de a la estadounidense Microsoft), capacidad de integrar diversos programas de mensajería instantánea en Pidgin, disponibilidad de utilizar Amarok como reproductor de música, y posibilidad de consultar el clima desde el escritorio.

Richard Stallman, por su parte, también es consciente de la superioridad técnica del software libre, aunque siempre haya enfatizado que el principal objetivo del Proyecto GNU no es tanto la calidad del resultado como la libertad del mismo, una libertad permite una mejor adquisición de competencias por parte de la población a alfabetizar, a la vez que asegura un mayor respeto de los derechos fundamentales de los usuarios. Sin embargo, esa superioridad del software libre también ha provocado colateralmente que surjan nuevos movimientos mucho más preocupados por su finalidad pragmática que en sus fundamentos filosóficos. Movimientos que, para distanciarse del objetivo original del Proyecto GNU y la Free Software Foundation y

aunque sigan compartiendo metas comunes, utilizan términos como Linux para referirse a GNU/Linux y open source (código abierto) como sustituto del software libre. Richard Stallman lo expresa de la siguiente manera (Stallman 2001):

"*Finalmente, la gente observó este fenómeno. En la década de 1980 muchos de nosotros pensábamos que tal vez el software libre no sería tan bueno como el software no libre, porque no tendríamos tanto dinero para pagar a la gente. Y por supuesto gente como yo, que valora la libertad y la comunidad, dijo, «bueno, usaremos software libre de todos modos». Merece la pena hacer un pequeño sacrificio y renunciar a algunas comodidades técnicas a cambio de tener libertad. Pero lo que la gente empezó a observar, hacia 1990, es que nuestro software era en realidad mejor. Tenía más capacidad, era más fiable que las alternativas propietarias. […] En cualquier caso, existe aún un grupo de gente que considera este beneficio particular como la principal razón por la cual a los usuarios se les debería permitir tener esas libertades. Si me habéis estado escuchando, os habréis enterado de que, para hablar a favor del movimiento de software libre, hablo sobre temas éticos y sobre el tipo de sociedad en la que queremos vivir, sobre qué produce una buena sociedad, así como sobre los beneficios prácticos y materiales. Todo esto es muy importante, todo esto constituye el movimiento de software libre. Ese otro grupo de gente —que es llamado movimiento open source— sólo cita los beneficios prácticos. Niegan que esta sea una cuestión de principios. Niegan que la gente tenga derecho a la libertad de compartir con su vecino y de comprobar lo que está haciendo el programa y de cambiarlo si no les gusta. Dicen, de todos modos, que permitirle hacer estas cosas es algo útil. Así que van a las empresas y les dicen, «podríais hacer más dinero si dejáis que la gente haga esto». De modo que podréis ver que, hasta cierto punto, llevan a la gente en una dirección parecida, pero por motivos totalmente distintos, por razones filosóficas fundamentalmente distintas".*

El término open source (código abierto) surge en 1998 como intento de desambiguar el doble significado del término free software en inglés (software libre y software gratuito). Una de las mayores confusiones existentes alrededor del software libre es su supuesta condición de gratuidad, software libre no es sinónimo de software gratuito, ya que como afirma la FSF "free software is a matter of freedom, not price" (el software libre es un asunto de libertad, no de precio). Por lo tanto, no se debe confundir free software (software libre) con freeware (software gratuito). El freeware podría ser perfectamente gratuito y privativo, no disponiendo del código fuente ni de la posibilidad de hacer uso de las cuatro libertades que definen al software libre. Por otra parte, el software libre también podría ser distribuido a cambio de dinero, con valores añadidos en su presentación, adaptaciones ad-hoc, servicios de instalación o soporte, etc. Richard Stallman resulta rotundo en este aspecto (Stallman 1999): "la filosofía del software libre rechaza una práctica empresarial concreta y muy generalizada, pero no rechaza el negocio en general. Cuando una empresa respeta la libertad de los usuarios, le deseamos mucho éxito". Sin embargo, y lejos de desambiguar el significado de free en el conjunto free software, el término open source se convirtió en la bandera de un nuevo movimiento, encabezado por el hacker Eric S. Raymond y la Open Source Initiative (OSI), cuyo principal objetivo es defender y demostrar la finalidad pragmática del software libre por encima del concepto de libertad promovido por Stallman. Una definición estricta de software open source es aquel que cumple con el siguiente decálogo (Anon.):

1. Libre redistribución.
   La licencia no debe restringir a ninguna parte de vender o regalar el software como componente de una distribución que contenga programas de distintas fuentes. La licencia no debe requerir un royalty u otra cuota por su venta.
2. Código fuente.
   El programa debe incluir el código fuente, y debe permitir su distribución tanto como código fuente como compilado. Cuando alguna forma de un producto no sea distribuida con el código fuente, deberá haber algún medio bien publicitado para obtener el código fuente por no más del coste de una razonable reproducción, preferiblemente descargándose vía internet sin cargo. El código fuente debe ser la forma preferida por la cual un programador modificaría el programa. Un código fuente deliberadamente confuso no está permitido. Formas intermedias como los resultados de un preprocesador o traductor no están permitidas.

3. Trabajos derivados.
   La licencia debe permitir modificaciones y trabajos derivados, y debe permitir que estos se distribuyan bajo las mismas condiciones de la licencia del software original.
4. Integridad del código fuente del autor.
   La licencia puede restringir la distribución del código fuente en una forma modificada sólo si la licencia permite la distribución de "parches" con el código fuente con el propósito de modificar el programa cuando sea compilado. La licencia debe permitir explícitamente la distribución de software compilado desde el código fuente modificado. La licencia puede requerir que los trabajos derivados lleven un nombre o número de versión diferente al del software original.
5. No discriminación de personas o grupos.
   La licencia no debe discriminar a ninguna persona o grupos de personas.
6. No discriminación de campos de iniciativa.
   La licencia no debe restringir a nadie de usar el programa en un campo específico de iniciativa. Por ejemplo, no puede restringir el programa de ser usado en un negocio, o de ser usado para investigación genética.
7. Distribución de la licencia.
   Los derechos adjuntos al programa deben aplicarse a todas las personas a quienes se redistribuya el programa sin la necesidad de ejecución de una licencia adicional para aquellas partes.
8. La licencia no debe ser específica de un producto.
   Los derechos adjuntos a un programa no deben depender de que el programa sea parte de de una distribución particular de software. Si el programa es extraído de esa distribución y usado o distribuido dentro de las condiciones de la licencia del programa, todas las partes a quienes el programa se redistribuya deberían tener los mismos derechos que aquellos que son concedidos en conjunción con la distribución original de software.
9. La licencia no debe restringir otro software.
   La licencia no debe establecer restricciones sobre otro software que sea distribuido con el software al que se le aplica la licencia. Por ejemplo, la licencia no debe insistir en que todos los demás programas distribuidos en el mismo medio deben ser software open-source.
10. La licencia debe ser tecnológicamente neutral.
    La licencia no debe ser predicada sobre ninguna tecnología individual o estilo de interfaz.
    Desde un punto de vista práctico, las definiciones de software libre y open source, dadas por la FSF y la OSI, son prácticamente equivalentes. En este sentido Mark Webbink, de Red Hat, señala que (Webbink 2003):

"Contrastando las definiciones de Open Source y Software Libre, se observa que todo el Software Libre es Open Source, pero administrado por la Free Software Foundation, no todo el Open Source es Software Libre. La diferencia surge principalmente de la denominada licencia de compatibilidad, pero en gran medida, las diferencias son meramente filosóficas y no sustanciales".

Sin embargo, Richard Stallman, padre fundador del movimiento del software libre, sí que ha mostrado su desacuerdo con el término open source en diversas ocasiones. En su trabajo "Por qué «software libre» es mejor que «open source»" de 1998 (Stallman 1998), escrito ya en los albores del movimiento open source, se indica que la difusión de este movimiento puede responder a una política de difusión pragmática y mercantilista, provocada quizás por el miedo a la libertad:

"El argumento principal del término «software open source» es que el término «software libre» hace que algunas personas se sientan incómodas. Esto es cierto: hablar sobre libertad, sobre asuntos éticos, sobre responsabilidades así como sobre conveniencia, es pedirle a la gente que piense sobre cosas que preferiría ignorar. Esto puede causar malestar y algunas personas pueden rechazar la idea por eso. De esto no se debe deducir que la sociedad estaría mejor si dejamos de hablar de este tipo de cosas. Años atrás, los desarrolladores de software advirtieron esta reacción de malestar y comenzaron a explorar otro enfoque para evitarlo. Se imaginaron que manteniendo el silencio sobre cuestiones de ética y de libertad, y hablando

únicamente de los beneficios prácticos inmediatos de cierto software libre, podrían ser capaces de «vender» software más efectivamente a ciertos usuarios, especialmente a las empresas. El término open source se ofrece como una forma más de hacer esto —una forma de ser «más aceptable a las empresas». Los puntos de vista y los valores del movimiento open source se derivan de esta decisión. Este enfoque ha demostrado ser efectivo en sus propios términos. Hoy mucha gente se está cambiando al software libre por razones puramente prácticas. Esto es bueno, mientras siga ocurriendo, ¡pero eso no es todo lo que necesitamos hacer! Atraer a los usuarios al software libre no es todo el trabajo, es sólo el primer paso. Tarde o temprano a estos usuarios se les invitará a regresar al software propietario por alguna ventaja práctica. Innumerables empresas intentan ofrecer tal tentación, y ¿por qué iban a rechazarla los usuarios? Sólo si han aprendido a valorar la libertad que les da el software libre, en su propio interés. Depende de nosotros difundir esta idea — y para eso tenemos que hablar de libertad. En buena parte, el enfoque «guarda silencio» para acercarse a las empresas puede ser útil para la comunidad, pero también debemos tener bastante libertad de expresión".

La analogía entre "valorar la libertad" gracias al software libre y la autoconfianza e independencia en alfabetización informática de la que habla Tuckett por lo tanto parece clara. Una adopción de software libre por razones meramente pragmáticas significa que, tal como indica Stallman, esa independencia es efectiva únicamente mientras tal ventaja práctica se mantenga. Si los usuarios se quedan atascados de nuevo en el software propietario, y las iniciativas de software libre desaparecen por falta de recursos o intereses de terceros, todo el camino recorrido y cualquier opción de independencia para la alfabetización informática queda automáticamente abortado. Es por ello que la finalidad primera del software libre será la concienciación ética y social en la libertad, mucho más allá de las ventajas técnicas y económicas que el movimiento open source pueda descubrir a las empresas. En su posterior actualización de 2007 "Por qué el código abierto pierde el punto de vista del Software Libre" ("Why "Open Source" misses the point of Free Software") (Stallman 2007), Richard Stallman añade en este sentido:

"Estas libertades son de vital importancia. Son esenciales, no solamente para el bien del usuario individual, porque promueven la solidaridad social: el compartir y cooperar. Estas libertades se vuelven aún más importantes mientras nuestra cultura y actividades de la vida diaria se vuelven más y más digitales. En un mundo de sonidos, imágenes y palabras digitales, el software libre viene a representar a la libertad en general. Decenas de millones de personas alrededor del mundo utilizan ahora software libre; las escuelas de regiones de India y España enseñan a todos los estudiantes a utilizar el sistema operativo libre GNU/Linux. La mayoría de estos usuarios nunca han escuchado las razones éticas por las cuales desarrollamos este sistema y construimos la comunidad de software libre, porque este sistema y esta comunidad son descritos como «de código abierto», y atribuidos a una filosofía diferente, en la cual estas libertades son rara vez mencionadas. […] no todos los usuarios y programadores de software libre estaban de acuerdo con las metas del movimiento del software libre. En 1998, una parte de la comunidad de software libre se bifurcó y empezó a hacer una campaña en nombre del «código abierto» («Open Source» en inglés). El término se propuso originalmente para evitar un posible malentendido con el término «software libre», pero pronto se asoció con visiones filosóficas diferentes a las del movimiento del software libre. Algunos de los defensores del «código abierto» lo consideraron una «campaña de marketing para el software libre»; la cual atraería a los ejecutivos de empresas al citar los beneficios prácticos, mientras evitaba las ideas de correcto e incorrecto que quizá no deseaban oír. Otros defensores rechazaban frontalmente los valores éticos y sociales del software libre. Cualesquiera que hayan sido sus perspectivas, cuando hacían campaña por el «código abierto» no citaban o abogaban por esos valores. El término «código abierto» fue rápidamente asociado con la costumbre de citar solamente los valores prácticos, como el hacer software potente y confiable. La mayoría de simpatizantes del «código abierto» han llegado a dicho movimiento desde entonces y ése es el significado que le atribuyen. Casi todo el software de código abierto es software libre. Los dos conceptos describen casi la misma categoría de software, pero representan puntos de vista basados en valores fundamentalmente diferentes. El código abierto es una metodología de programación, el software libre es un movimiento social. Para el movimiento del software libre, el software libre es un imperativo ético porque solamente el software libre respeta la libertad del usuario. En cambio, la filosofía del código abierto considera los asuntos bajo los términos de cómo hacer «mejor» al software, en un sentido práctico solamente. Plantea que el software que

no es libre no es una solución óptima. Para el movimiento del software libre, sin embargo, el software que no es libre es un problema social, y usar en su lugar software libre es la solución. Software libre. Código abierto. Si es el mismo software, ¿importa acaso qué nombre se utiliza? Sí, porque las diferentes palabras expresan ideas diferentes. Mientras que un programa libre, con cualquier otro nombre, le dará la misma libertad; establecer la libertad de una manera perdurable depende sobre todo de enseñar a las personas a valorar la libertad. Si desea ayudar a hacer esto, es esencial que hable de «software libre». Nosotros, en el movimiento del software libre, no vemos el ámbito del código abierto como al enemigo; el enemigo es el software privativo, el que no es libre. Pero queremos que la gente sepa que defendemos la libertad, así que no aceptamos que se nos identifique como partidarios del código abierto".

En relación a los términos software libre y open source, Richard Stallman viene a corroborar lo que decía Webbink sobre sus diferencias, los dos conceptos describen casi la misma categoría aunque representan puntos de vista basados en valores diferentes. En el texto, también se afirma que el open source es una metodología de programación mientras que el software libre se trata de un movimiento social. Las ventajas del software libre/open source como metodología de trabajo y desarrollo han sido expuestas de forma conocidas por Eric S. Raymond en su obra clásica "The Cathedral and the Bazaar" (Raymond 1999), una superioridad que, tal como señala J.J. King (King 1999), podría responder a una ideología neo-liberal basada en el libre mercado. Richard Stallman, en este sentido, añade:

"El open source puede ser el mejor modelo para el desarrollo de software. Podría producir mejor software que el Software Libre. No lo sé - no lo puedo decir. Pero la cosa más importante para el futuro de la comunidad del software es que las personas piensen sobre la cuestión de la libertad. Entonces podremos estar en guardia contra las personas que nos invita a dejarlo. Porque la gente constantemente nos invita a dejarlo. Hay un montón de programas propietarios que puedes obtener que funcionarán en un sistema operativo libre como GNU/Linux. Y cada uno de ellos es una oportunidad de abandonar algo de libertad por conveniencia".

Para J.J. King "sería una tragedia dejar que los privativos infecten el sistema con trozos de código propietario, sólo por razones de conveniencia. Eso sería un triunfo del pragmatismo bruto -un pragmatismo que evita toda noción de poner la ética en primer lugar, de prever el futuro, de pensar otra cosa que no sea el momento- sobre los considerados esfuerzos políticos de todo el movimiento del Software Libre". Sin embargo, y tal como pone de manifiesto Richard Stallman, el ámbito del open source no es el principal enemigo, "el enemigo es el software privativo, el que no es libre", ya que al fin y al cabo, éste será el que prive de libertad al usuario final. El software privativo (o propietario), se define como aquel que no cumple con las cuatro libertades del software libre, es decir, un software que priva de libertad a los usuarios y niega la autoconfianza e independencia tecnológica para llevar a cabo su capacidad de elección. Este tipo de software, distribuido de manera global y cerrada, está mayoritariamente representado por productos de empresas como Microsoft o Apple, que, tal como se ha puesto de manifiesto, técnicamente no son ser más potentes o fiables que otras alternativas de software libre. Por otra parte, y a una menor escala, el concepto de software propietario también incluye a todos los extractos de código no libre que, no cumpliendo con las cuatro libertades de la FSF, se distribuyen de forma independiente o junto a otros proyectos y productos. Y lo cierto es que, la inclusión de software no libre en las distribuciones de GNU/Linux, es uno de los problemas que más daño hace a la comunidad del software libre actualmente, y esto, aunque muchas veces ocurre por la pasividad a la hora de buscar alternativas libres, en otras ocasiones ocurre debido a intereses creados y conveniencias de terceros. Richard Stallman lo señala de la siguiente manera (Stallman 2000):

"Se justifica la inclusión de software no libre en nombre de la «popularidad de Linux» —en efecto, valoran más la popularidad que la libertad. Algunas veces se admite abiertamente. Por ejemplo, en Wired Magazine, Robert McMillan, editor de Linux Magazine, afirma que «el movimiento por el software de código abierto debería impulsarse sobre la base de decisiones técnicas, no políticas». Y el presidente de Caldera animó públicamente a los usuarios a abandonar el objetivo de la libertad y trabajar en cambio por la «popularidad de Linux». Incluir software no libre en el sistema GNU/Linux puede aumentar su popularidad, si por popularidad entendemos el número de personas que usan GNU/Linux en combinación con software no libre. Pero al

mismo tiempo, implícitamente se está animando a la comunidad a aceptar el software no libre como algo positivo, y a olvidar el objetivo de la libertad. De nada sirve caminar más rápido si nos apartamos del camino".

Actualmente, la mayoría de las distribuciones más populares de GNU/Linux incluyen elementos privativos que no respetan la libertad de los usuarios (Anon.). Debian, en la práctica, comete excepciones sobre la inclusión de software no libre en el kernel proporciona acceso e información sobre repositorios no libres, Ubuntu proporciona repositorios específicos de software no libre y el instalador advierte de forma automática sobre la existencia de este software, SUSE y openSUSE proporcionan acceso a repositorios de software no libre, Fedora, Mandriva y Red Hat no disponen de una política clara de qué software se incluye en la distribución, Gentoo facilita la instalación de software no libre desde los paquetes principales, etc. Basicamente, los problemas existentes en las distribuciones de GNU/Linux se dividen en dos grandes grupos: ausencia de una política clara de eliminación de software no libre cuando sea descubierto en el sistema, e inclusión de blobs (partes de código para algún propósito distribuidos sin la fuente) en la versión del kernel Linux. Contabilizadas por la Free Software Foundation (Anon.), existen únicamente siete distribuciones de GNU/Linux 100% libres en la actualidad (gNewSense, Ututo, Dragora, Dynebolic, Musix GNU+Linux, BLAG y Trisquel), y una más que está en proceso de revisión (GNUstep Live). Una guía con los requisitos necesarios para ser considerado 100% libre se encuentra disponible en la página web del Proyecto GNU (Anon.).

La inclusión de software privativo, en los proyectos y distribuciones GNU/Linux, puede añadir funcionalidades extras al sistema, ventajas a corto plazo en el caso de que dichas funcionalidades estén patentadas o todavía no existiera alguna alternativa libre. Sin embargo, esta visión pragmática del software libre lo único que hace es aprovecharse de los beneficios económicos de disponer del código abierto a corto plazo, olvidando los conceptos de libertad y capacitación que los fundamentan y que podrán hacerlos perdurables a lo largo del tiempo. El movimiento del software libre ha sido tachado de idealista desde sus comienzos, y sin embargo eso no ha impedido grandes resultados. En palabras de Richard Stallman (Stallman 2000):

"El proyecto GNU es idealista y cualquiera que hoy promueva el idealismo se enfrenta a un gran obstáculo: la ideología dominante anima a la gente a descartar el idealismo por ser «poco práctico». Nuestro idealismo ha sido extremadamente práctico: es la razón de que existe un sistema operativo GNU/Linux libre. La gente que disfruta de este sistema debería saber que se trata de nuestro idealismo hecho realidad."

Pero, ¿cuáles son esos beneficios económicos que proporciona el modelo del software libre para que haya tanta gente interesada? ¿podrían aplicarse estos beneficios a las administraciones públicas y beneficiar con ello a la sociedad? y, sobre todo, ¿qué relación existe con la brecha digital?

5. Software libre, beneficios económicos y brecha digital

Según un informe publicado por la Unión Internacional de Telecomunicaciones (UIT) en 2009 (ITU Telecommunication Development Sector 2009), la magnitud de la brecha digital global se mantuvo inalterable entre 2002 y 2007, a pesar de los avances producidos en las TIC (Tecnologías de la Información y la Comunicación), lo cual se asocia a los costes que tienen dichas tecnologías en los países en vías en desarrollo.

Si existiera algún tipo de correlación entre la brecha digital y los beneficios económicos provenientes de las tecnologías, entonces podría establecerse una correlación directa entre la disminución de la brecha digital y la aplicación más pragmática del software libre/open source. Sin embargo, no todas las TIC que se aplican en los países en vías de desarrollo están basadas en software libre, ni, tal como indica el autor Michel J. Menou (Menou 2004), todas las TIC, sean o no software libre, son aplicadas de la forma más adecuada en los campos de alfabetización informacional y educación:

"Un rasgo interesante de las TICs consiste en que pueden ser al mismo tiempo objeto y canal de aprendizaje. Ello atrae naturalmente la atención de gestores y agencias de financiación que buscan continuamente la

solución directa, simple, a gran escala y barata – de ahí la idea de que los materiales y programas de autoaprendizaje en Internet serán suficientes para resolver la mayor parte de los problemas de la alfabetización informacional y en el uso de ordenadores […] Da miedo hasta cierto punto ver cómo se aplican, en ambas direcciones, a la instrucción por medio de ordenadores los mismos conceptos erróneos que se aplicaron hace muchos años a la televisión educativa".

Por lo tanto, se pone de manifiesto que hace falta mucho más que una mera aplicación pragmática de las TIC para paliar la brecha digital. Todo parece indicar que, según algunas perspectivas, para aumentar la alfabetización en información y disminuir la brecha digital hacen falta una serie de implicaciones políticas y sociales que superen la mera adopción de tecnologías sin una implicación ética. Para Richard Stallman (Stallman 2009), un uso incorrecto de las TIC podría ser incluso peor que su inexistencia:

"La idea de la brecha digital presupone que la cuestión es tener o no tener computadora. No estoy de acuerdo, porque tener acceso a las computadoras no es bueno si el software te priva de la libertad. Windows y Macintosh son propietarios. Llevar a más gente a usar Windows o Macintosh no es un avance social. La distinción principal es usar software propietario o evitarlo. El software libre ofrece un camino para usar computadoras. El software libre respeta las libertades esenciales del consumidor: ejecutarlo como quieras, cambiar el código fuente para que el programa haga lo que quieras, distribuir copias exactas cuando quieras y distribuir copias de tus versiones cambiadas cuando quieras. La ventaja del software libre es respetar tu libertad".

En una línea similar, de desarrollo socioeconómico basado en la información y el conocimiento, los autores Verónica Xhardez y Martín Olivera con aportes de la Comisión de Agenda Digital de Solar - Software Libre Argentina (Xhardez et al. 2009), ponen en relación el software libre con la construcción de una Agenda Digital Argentina basada en tres pilares: la soberanía e independencia tecnológica del país, los derechos humanos, y la participación ciudadana. Respecto a la soberanía e independencia tecnológica, el informe indica que el software libre ofrece el único camino posible para asegurar la autonomía en materia de información, además de reconocer la capacidad del software libre para impulsar la economía a través de una competencia menos polarizada del sector, y una mayor generación de trabajo a nivel local. Respecto a los derechos humanos, se indica sobre la capacidad del software libre para dar acceso a la cultura, la comunicación y expresión, y a la educación sin diferencias de género, raza, religión, ocupación o pertenencia a grupos, instituciones o partidos. Además, también se enfatiza sobre la cultura de compartir y colaboración que se fomenta desde los proyectos de software libre. Por último, respecto a la participación ciudadana, se señala que una utilización de software libre por parte de la Administración permite una relación más transparente con los ciudadanos, especialmente en el manejo de la información pública.

El profesor Jesús González Barahona, en su trabajo "La imparcialidad de los estados y la industria del software" (González Barahona 2002b), analiza la relación existente entre la imparcialidad de los estados y el funcionamiento de la industria del software. En su trabajo, Barahona, cuestiona la imparcialidad de los estados indicando que se priman unos modelos de negocio sobre otros. Principalmente, esto se produce a través de la legislación pertinente en materia de propiedad intelectual y propiedad industrial:

"Visto desde este punto de vista, las legislaciones actuales difícilmente pueden considerarse como imparciales con respecto al modelo de negocio. De hecho, el modelo de negocio tradicional en el mundo del software-producto, la venta de licencias de uso limitadas (lo que adquirimos cuando compramos un programa propietario) sólo es posible porque se ha legislado de forma que lo sea. Y al hacerlo, no sólo el estado ha decidido que éste sea el modelo más popular, sino que ha hecho posibles consecuencias como que cuanto mayor sea el mercado donde se venden licencias, más beneficios se consigan (de forma casi directa: los costes aumentan poco, los ingresos mucho), o que haya empresas que detenten monopolios de facto en amplios nichos (los usuarios prefieren usar el producto mayoritario, nadie más que la empresa que lo produce puede proporcionárselo, y nadie puede competir proporcionando el mismo producto ".

Respecto a la financiación pública de la formación y la investigación, componentes clave para el desarrollo del capital intelectual de las naciones, el profesor Barahona también señala algunas de las deficiencias de la Administración:

"Y hay otros muchos casos, como cuando con dinero público se pagan cursos de formación, supuestamente de ofimática, que se convierten en realidad en clases de capacitación para el uso de ciertas herramientas de software propietario. O cuando se malgastan cantidades ingentes de recursos públicos en programas de investigación y desarrollo destinados a subvencionar la creación de programas propietarios, que sólo podrán comercializar, y convertir en beneficios, los mismos que reciben las subvenciones. Y más, y más y más."

Como conclusión, Barahona indica que el Estado debe favorecer el modelo de sector informático que maximice el bien común, es decir, que produzca un mejor software para la sociedad medido en términos de recursos consumidos, calidad y cantidad de desarrollos, y difusión a número de usuarios.

Para las administraciones públicas, gran parte de las ventajas del uso del software libre vendrá dada por su carácter de independencia y capacidad de adaptación, consecuencia directa de las cuatro libertades que lo definen. Estas ventajas son: mejor aprovechamiento de los recursos invertidos, independencia hacia los proveedores, adaptación a las necesidades exactas, disponibilidad del software a largo plazo, fomento del tejido tecnológico local, mayor formación del personal informático a través de la capacitación, y mayor transparencia en el modo de actuación de la Administración. En la misma línea, Barahona señala que las administraciones públicas actúan en el mercado del software al menos de tres formas (González Barahona 2002a):

- Comprando programas y servicios relacionados con ellos. Las administraciones, como grandes usuarios de informática, son un actor fundamental en el mercado del software.
- Promoviendo de diversas formas el uso (y la adquisición) de ciertos programas en la sociedad. Esta promoción se hace a veces ofreciendo incentivos económicos (desgravaciones fiscales, incentivos directos, etc.), a veces con información y recomendaciones, a veces por *efecto ejemplo*.
- Financiando (directa o indirectamente) proyectos de investigación y desarrollo que están diseñando el futuro de la informática.

Por lo tanto, se indica que además de beneficiarse de las ventajas del software libre, las administraciones públicas también tienen la posibilidad de influir en qué medida el software libre beneficia a la sociedad, y cómo la sociedad devuelve esas ventajas a la comunidad del software libre. En este sentido, Barahona indica que:

"No es muy buena idea promover el uso de Internet en la sociedad de una forma que fomente hasta la exageración un determinado monopolio, que a la larga sería perjudicial para todos (salvo para la empresa monopolística, claro). […] No se suele considerar que la misma cantidad de dinero se puede usar en comprar una cierta cantidad de licencias de un programa propietario, o en adquirir una copia de uno libre, y contratar soporte o adaptaciones para él. Así, en lugar de modernizar la sociedad, se fomenta la compra de ciertos productos, desincentivando la de otros que habrían producido más beneficios en la sociedad".

Y para el caso concreto de la informatización en escuelas, añade:

"Imaginemos por un momento que parte de la cantidad destinada a un programa de informatización de escuelas se dedica a crear una distribución de GNU/Linux adaptada a las necesidades de la docencia en enseñanza primaria. Y con el resto de los recursos, se contrata soporte para que el software sea mantenido en esas escuelas, de forma que no sea un simple software florero, sino que realmente haya gente encargada de su correcto funcionamiento. Así se cubren no sólo las necesidades del sistema educativo, también se genera un mercado para empresas, habitualmente de ámbito local, capaces de ofrecer servicios de mantenimiento. Y por supuesto, se deja completamente abierto el camino al futuro: el software no quedaría

obsoleto en pocos años, obligando a comenzar desde cero, sino que se podría ir actualizando incrementalmente, año a año, manteniendo los beneficios del programa con una inversión similar".

En un nivel universitario, el profesor Vicente Matellán afirma que la administración debería obligar el uso de tecnologías neutrales (libres) en la docencia en tecnologías de la información. Para ello esgrime dos razones fundamentes (Matellán Olivera 2002):

"La primera menos importante: por precio. Las licencias de campus son muy baratas comparadas con los precios individuales, pero aún así hay que pagarlas y las donaciones sólo se producen al subconjunto limitado de universidades de mucho prestigio con la idea de que arrastren a las demás. La segunda es la más importante desde mi punto de vista: las administraciones públicas deberían forzar el uso de tecnologías no propietarias para no colaborar en el mantenimiento de monopolios en ningún campo de estas tecnologías. Aceptar que se emplee como lenguaje Java, o que los únicos sistemas operativos que vean los alumnos sean los de la familia Microsoft Windows, me parece peligroso desde el punto de vista social".

Debe notarse que, la segunda razón, considerada de mayor importancia por el autor, puede ser extrapolada a cualquier ámbito de la Administración, incluyendo todos los proyectos e iniciativas en materia de ahorro de costes y alfabetización tecnológica. Por lo tanto, ni todas las iniciativas son iguales en cuanto a su filosofía, ni todos los tipos de software son adecuados para alcanzar esas características de independencia y autocapacitación necesarias para el proceso de alfabetización. Un proyecto de alfabetización tecnológica (o migración en las administraciones) que utilice software privativo, fomenta la dependencia sobre la empresa proveedora, imposibilita el aprendizaje y el conocimiento sobre cómo funciona el software, dificulta su capacidad de adaptación, y perjudica a la economía del país fomentando monopolios en su peor sentido. Tal como indica Barahona en su trabajo "Software libre, monopolios y otras yerbas" (González Barahona 2002d), el software libre en algún caso puede fomentar monopolios de producto, pero nunca monopolios de empresa. Esto significa que cualquier empresa podrá trabajar en mejorar un determinado software, fomentando así los mercados en competencia sana e impulsando la economía de un país a través de creación de puestos de trabajo.

En esta línea, algunos ejemplos de ahorro en la Administración gracias al software libre, son los casos de Brasil o Francia.

En Brasil, según se indica desde el Servicio Federal de Procesos de Datos (SERPRO), dependiente del Ministerio de Hacienda (Anon.), en 2008 se han ahorrado 370 millones de reales (unos 123,7 millones de euros) gracias el uso de sistemas operativos, navegadores de Internet, programas de correo electrónico y otro software libre. Esta cifra equivale al doble de la inversión para el sistema de declaración del impuesto de renta de personas físicas y de consulta del Sistema Integrado de Administração Financeira (Siafi), y cerca de un cuarto del presupuesto del propio Serpro, lo cual ha permitido destinarla a la instalación y funcionamiento de más de cinco mil telecentros en todo el país, influyendo en la alfabetización tecnológica y disminuyendo la brecha digital. En cuanto al grado de libertad del software utilizado, según se indica en el Programa Serpro de Software Livre PSSL (Anon.), en la mayor parte de los puestos se utiliza el sistema operativo Ubuntu, y por lo tanto, no es un sistema operativo 100% libre.

En Francia, según se comenta desde el Observatorio de Open Source de la Comisión Europea (Anon.), la policía ha ahorrado 50 millones de euros desde 2004, debido a sus migración a Ubuntu y OpenOffice. Esto ocurrió cuando Microsoft trató de forzar a la Administración a comprar más licencias de Windows y Microsoft Office, y decidieron migrar a alternativas de software libre. Debido a la utilización de Ubuntu, el grado de libertad por parte de la administración francesa sigue sin ser pleno, sin embargo, esto no implica que no sea un gran avance respecto al uso de Windows XP, del que se dice que "las dos grandes diferencias son los iconos y los juegos, y los juegos no son nuestra prioridad".

En España, existe la experiencia de Extremadura, comunidad autónoma pionera en el desarrollo del sistema operativo gnuLinEx y que ha servido de modelo a otras comunidades como Andalucía para su sistema Guadalinex. Según declaraciones de Vicente Parejo, coordinador general de Tecnologías de la Información y la Comunicación de la Consejería extremeña de Educación (Anon.), "El software libre permitió generalizar el uso de las tecnologías, porque con el coste de las licencias privativas hubiese sido imposible". gnuLinEx (Anon.) está basado en Debian, y también incluye software que no es totalmente libre o incluso propietario, como son los navegadores Firefox y Opera, por lo tanto, tampoco se trata de un sistema operativo 100% libre y la capacidad de independencia no es total.

En materia de alfabetización, la iniciativa más paradigmática es OLPC - One Laptop Per Child (Una Laptop por Chico). OLPC (Anon.) es un proyecto fundado por Nicholas Negroponte en Estados Unidos cuya meta es "proveer a los chicos del mundo con nuevas oportunidades para explorar, experimentar y expresarse", enfatizándose que "es un proyecto educativo, no de laptops". La idea consiste en vender ordenadores portátiles (llamados XO) a un precio de 100 dólares. Para ello, se utiliza un sistema operativo GNU/Linux, basado en Fedora, y algunos intentos de aumentar la gama no exentas de polémica en el mundo del software libre. Uno de los aspectos más controvertidos fue la decisión de trabajar también con Windows para diversificar sus productos, hecho que provocó que Richard Stallman retirara su apoyo y a cambiara a la alternativa Lemote (Stallman). En su artículo "¿Se puede rescatar OLPC de Windows?" (Stallman 2008), Richard Stallman afirma:

"Algunos entusiastas del sistema GNU/Linux están sumamente decepcionados ante la posibilidad de que el XO, si alcanza el éxito, no sea una plataforma para el sistema que aman. Aquellos que han apoyado el proyecto OLPC con esfuerzo o con dinero, podrían sentirse traicionados. Sin embargo, esas preocupaciones quedan soslayadas por otros factores en juego: que el XO sea un instrumento para alcanzar la libertad o un instrumento de sumisión. Desde que se anunció OLPC por primera vez, lo hemos imaginado como una forma de conducir a millones de niños en todo el mundo hacia una vida en la que puedan hacer sus tareas de computación en libertad. El proyecto anunció sus intenciones de proporcionar a los niños un medio que les permita aprender sobre computadoras, dándoles la posibilidad de estudiar y manipular el software. Es posible que esto aún sea así, pero existe el peligro de que no lo sea. Si la mayoría de los XO que efectivamente se usan funcionan con Windows, el resultado global será el opuesto. El software privativo mantiene a los usuarios divididos e indefensos. Su funcionamiento es secreto, por lo tanto es incompatible con el espíritu del aprendizaje. Enseñar a los niños a usar un sistema privativo (no-libre) como Windows no hace del mundo un lugar mejor, porque los pone bajo el poder del desarrollador del sistema -tal vez de forma permanente. Sería como iniciar a los niños al uso de una droga adictiva. Si el XO se transforma en una plataforma para propagar el uso de software privativo, su efecto general en el mundo será negativo".

Otros ejemplos de proyectos similares a OLPC alrededor del mundo son el citado Lemote YeeLoong en China (utilizando software 100% libre) o Shakshat en India.

En el caso de España, en abril de 2009 también se anunció (Anon.) que iba a ponerse en marcha un programa inspirado por OLPC con la finalidad de dar un portátil a los niños de primaria, presumiblemente equipado con software de Microsoft. Las reacciones en contra no tardaron en sucederse. Richard Stallman se refirió en diversas conferencias como algo no ético, y desde Hipalinux (la asociación de usuarios españoles de GNU/Linux) se realizó un completo y urgente análisis de lo que una situación como esta podría suponer para la nación. Según el informe de Hispalinux (Anon.), la primera irregularidad consistiría en la ausencia de concurso público, lo que se considera contraria al principio legal de Neutralidad Tecnológica, a la igualdad de oportunidades que debe regir en a Administración Española, así como a los principios de la Ley 30/2007 de Contratos en el Sector Público y de la Ley 11/2007 de Acceso Electrónico de los Ciudadanos a los Servicios Públicos. La adjudicación unilateral del contrato a Microsoft, multinacional condenada por monopolio en la Unión Europea y Estados Unidos, también fomentaría un monopolio de empresa en el que quedarían excluidas todas las empresas nacionales, generando así importantes pérdidas económicas para el país y de puestos de trabajo. Por supuesto, todo ello sin contar con el riesgo que supondría la dependencia hacia una

empresa, afectada por la crisis, que: reduce recursos en educación (como el abandono de la Enciclopedia Encarta), despide empleados en todo el mundo (incluyendo España), y no invierte en investigación y desarrollo (según sus cuentas registradas). En definitiva, se señalan los peligros de depender tecnológicamente de una empresa que mantiene los servicios y productos en función de su rentabilidad en el mercado y no del interés o necesidad de sus clientes, dándose el antecedente de los problemas causados a Corea del Sur, cuando Microsoft discontinuó el desarrollo de la tecnología ActiveX utilizada su Administración y no pudo hacer nada. Por último, también se señala el problema del coste económico. La suma de la licencia del sistema operativo y el paquete ofimático Microsoft Office ascendería a 100 millones de euros anuales, a lo que también habría que sumar el coste de antivirus indispensable en Windows e innecesario en GNU/Linux. Se indica que una solución basada en software libre tendría un coste cero para las arcas del Estado y permitiría invertir en la creación de empleo estable, cualificado e innovador en las empresas nacionales. Sin embargo, según se informó desde la agencia Europa Press con fecha de 15 de mayo (Anon.), los portátiles que regale el PSOE finalmente sí podrían tener software libre, aunque de momento tampoco se ha especificado en sus características.

6. Conclusiones

Los conceptos de independencia y autoconfianza son características indispensables en los proyecto de alfabetización tecnológica. De una manera efectiva, estas características serán alcanzadas a través de las libertades y preceptos establecidos por la definición de software libre/open source. Los términos software libre y open source son generalmente equivalentes en la práctica, aunque no así en su política, ya que filosóficamente tan sólo el movimiento del software libre defiende de forma clara los principios de libertad que aseguran una independencia y autoconfianza a largo plazo. En los proyectos de alfabetización tecnológica (y migración en las administraciones), los usos de software libre gozan cada vez de mayor presencia, debido fundamentalmente a las ventajas económicas manifestadas en los estudios. Sin embargo, el software libre en los proyectos no siempre es completamente libre, ya que se normalmente se incluyen extractos de código propietario que muchas veces son provocados por esa visión pragmática del movimiento open source. Para el presente estudio, se han analizado algunos proyectos concretos en los que se observa una clara preponderancia del uso del software libre, sin embargo, en ellos resulta realmente difícil encontrar proyectos que se utilice únicamente software 100% libre, y por tanto la libertad de las iniciativas no es plena. Un aumento de la libertad del software traería mayores ventajas para los proyectos, afectando positivamente a los principios de independencia y autoaprendizaje que tan necesarios son para la alfabetización.

REFERENCIAS BIBLIOGRÁFICAS

Attwood, Gaynor; y Royal Society of Arts (Great Britain). Examinations Board. *RSA CLAIT: Computer Literacy and Information Technology.* 2nd ed. Oxford: Heinemann Educational in association with the RSA Examinations Board, 1993. ISBN 0435451871

Bawden, David. Revisión De Los Conceptos De Alfabetización Informacional y Alfabetización Digita*l. Anales De Documentació*n, 2002, vol. 5, p. 361-408.

Budd, Richard W. Información, Interacción, Intercomunicación: Tejiendo La Red Global. El Impacto De Internet En El Futuro De La Educació*n. Ze*r, 1997, vol. 2.

El Gobierno Dará Un Ordenador a Cada Niño De Primaria | Radiocable.Com - Radio Por Internet. f [Consulta:6/23/2009] Disponible desde Internet: http://www.radiocable.com/ordenador-nino-primaria7658.html.

Explaining Why we don't endorse other systems - GNU Project - Free Software Foundation. [Consulta: 09/09/2009]. Disponible desde Internet: http://www.gnu.org/philosophy/common-distros.html

Extremadura, Referente Estatal En 'software' Libre Educativo Desde 2002 - Xornal.Com. [Consulta: 09/09/2009]. Disponible desde Internet: http://www.xornal.com/artigo/2009/05/31/sociedad/extremadura-referente-estatal-software-libre-educativo/2009053022525658054.html.

FR: Gendarmerie Saves Millions with Open Desktop and Web Applications — Open Source Observatory. d [Consulta: 09/09/2009]. Disponible desde Internet: http://www.osor.eu/news/fr-gendarmerie-saves-millions-with-open-desktop-and-web-applications.

GnuLinEx.Org. [Consulta: 09/09/2009]. Disponible desde Internet:: http://www.linex.org/joomlaex/

González Barahona, Jesús M. ¿Qué Se Hace Con Mi Dinero*?. TodoLinu*x, 2002a, vol. 17, p. 12-13.

González Barahona, Jesús M. La Imparcialidad De Los Estados y La Industria Del Softwar*e. TodoLinu*x, 2002b, vol. 22, p. 12-13.

González Barahona, Jesús M. Software Libre En La Enseñanza Informátic*a. TodoLinu*x, 2002c, vol. 23, p. 12-13.

González Barahona, Jesús M. Software Libre, Monopolios y Otras Yerba*s. TodoLinu*x, 2002d, vol. 22, p. 12-13.

Guidelines for Free System Distributions - GNU Project - Free Software Foundation. g [Consulta:5/25/2009] Disponible desde Internet: http://www.gnu.org/philosophy/free-system-distribution-guidelines.html.

Haigh, R. W. Planning for Computer Literac*y. Journal of Higher Educatio*n, 1983, vol. 56(2), p. 161-171.
Hispalinux Censura El Coste Económico y La Dependencia Tecnológica De La "Solución" Microsoft Para Educación | Hispalinux. h [Consulta:6/23/2009] Disponible desde Internet: http://hispalinux.es/minipc-primaria.

Hunter, Beverly. *My Students use Computers: Learning Activities for Computer Literacy.* Reston: Reston Publishing, 1983. ISBN 0835948048.

Husen, T.; y POSTLETHWAITE, T. N. *The International Encyclopedia of Education: Research and Studie, vol1 2C.* Oxford: Pergamon Press, 1985.

ITU Telecommunication Development Sector. *Measuring the Information Society: The ICT Development Index.* Geneva: International Telecommunication Union, 2009.

King, J. J. Free Software is a Political Action: In Conversation with Richard M. Stallma*n. Heise-Telepoli*s, 1999, vol. 2009, no. 16/06/2009 [Consulta:16/06/2009]. Disponible desde Internet: http://www.heise.de/tp/r4/artikel/6/6469/1.html.

La definición de Software Libre - Proyecto GNU - Free Software Foundation (FSF). a [Consulta: 09/09/2009]. Disponible desde Internet: http://www.gnu.org/philosophy/free-sw.es.html.

List of Free GNU/Linux Distributions - GNU Project - Free Software Foundation (FSF). j [Consulta:5/25/2009] Disponible desde Internet: http://www.gnu.org/distros/free-distros.html.

Los Portátiles Que Regalará Zapatero Tendrán Software Libre. Europapress.Es. n [Consulta:6/23/2009] Disponible desde Internet: http://www.europapress.es/tecnologia/equipos-00448/noticia-portatiles-escolares-tendran-software-libre-20090515154409.html.

Marzal García Quismondo, Miguel Ángel; CUEVAS CERVERÓ, Aurora y COLMERO RUIZ, Mª Jesús. Actas del V Congreso Internacional Virtual en Educación (7-27 de febrero de 2005), 200*5. La Biblioteca Escolar Como Centro De Recursos Para El Aprendizaje (CRA*), p. 1-15.

Matellán Olivera, Vicente. ¿Qué Tiene Que Estudiar Un Informático*?. TodoLinu*x, 2002, vol. 23, p. 12-13.
Menou, Michel J. La Alfabetización Informacional Dentro De Las Políticas Nacionales Sobre Tecnologías De La Información y Comunicación (TICs): La Cultura De La Información, Una Dimensión Ausente*. Anales De Documentació*n, 2004, vol. 7, p. 241-261.

Mensching, G. E.; Y Mensching, T. B. eds., Ann Arbor MI: Pieran Press, 1989*. Computer Literacy, Information Literacy and the Role of the Instruction Librari*an, p. 21-31.

Ovens, C. S. H. Computer Literacy and Librarie*s. Electronic Librar*y, 1991, vol. 9(2), p. 85-89.

Portal do SERPRO — Governo Economiza R$ 370 Milhões Com Sistemas Operacionais De Computador. l [Consulta:6/23/2009] Disponible desde Internet: http://www.serpro.gov.br/serpronamidia/2009/abril/governo-economiza-r-370-milhoes-com-sistemas-operacionais-de-computador/.

Portal do SERPRO — Programa Serpro De Software Livre - PSSL. m [Consulta:6/23/2009] Disponible desde Internet: http://www.serpro.gov.br/tecnologia/software-livre/programa-serpro-de-software-livre-pssl
.
Raymond, Eric S. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* Rev ed. Beijing ; Cambridge, Mass.: O'Reilly, 1999. Disponible desde Internet: http://www.free-soft.org/literature/papers/esr/cathedral-bazaar/cathedral-bazaar.html.

Shapiro, J. J.; Y Hughes, S. K. Information Technology as a Liberal Art: Enlightenment Proposals for a New Curriculu*m. Educom Revie*w, 1996, vol. 31(2).

Stallman, Richard M. *¿Se Puede Rescatar OLPC De Windows? - Free Software Foundation.* , 2008 [Consulta: 6/23/2009] Disponible desde Internet: http://www.fsf.org/blogs/rms/se-puede-rescatar-olpc-de-windows.

Stallman, Richard M. *Computing Info.* [Consulta: 6/23/2009] Disponible desde Internet: <http://stallman.org/stallman-computing.html.

Stallman, Richard M. Entrevista: Perspectivas De Año Nuevo Richard M. Stallma*n. ELPAÍS.Co*m, 2009 [Consulta: 16/06/2009]. Disponible desde Internet: http://www.elpais.com/articulo/semana/inquieta/uso/informatica/vigilar/controlar/gente/elpeputeccib/20090101elpciblse_6/Tes

Stallman, Richard M. *Por Qué «software Libre» Es Mejor Que «open Source»*. En: S Software libre para una sociedad libre. Madrid: Traficantes de sueños, 199*8.*, p. 75-82. ISBN 84-933555-1-8.

Stallman, Richard M. Por Qué El Código Abierto Pierde El Punto De Vista Del Software Libre, 2007.

Stallman, Richard M. *Qué Encierra Un Nombr*e. En: Software libre para una sociedad libreTraficantes de sueños, 2000., p. 71-74. ISBN 84-933555-1-8.

Stallman, Richard M. *Software Libre: Libertad y Cooperació*n. En: Software libre para una sociedad libre. Madrid: Traficantes de sueños, 2001, p. 223-271. ISBN 84-933555-1-8.

Stallman, Richard M. *The GNU Operating System and the Free Software Movemen*t . En: Open Sources: Voices from the Open Source RevolutionO'Reilly, 1999. ISBN 1-56592-582-3.

The Open Source Definition | Open Source Initiative. [Consulta:15/06/2009] Disponible desde Internet: http://www.opensource.org/docs/osd.

Una Laptop Por Chico (OLPC), Una Laptop De $100 Para La Educación De Los Chicos Del Mundo. [Consulta:6/23/2009] Disponible desde Internet: http://www-static.laptop.org/es/.

Webbink, Mark. Understanding Open Source Softwar*e. Groklaw*, 2003.

Why Linux is Better. [Consulta: 09/09/2009]. Disponible desde Internet: http://www.whylinuxisbetter.net/.

Xhardez, Verónica; Y OLIVERA, Martín. Agenda Digital, Software Libre y Solidaridad Tecnológic*a. Solar: Software Libre Argentin*o, 2009.