

Software libre en la formación de traductores: entre el pragmatismo y la utopía

Oscar DÍAZ FOUCES
Universidade de Vigo

Como citar este artículo:

DÍAZ FOUCES, Oscar (2005) «Software libre en la formación de traductores: entre el pragmatismo y la utopía», en ROMANA GARCÍA, María Luisa [ed.] *II AIETI. Actas del II Congreso Internacional de la Asociación Ibérica de Estudios de Traducción e Interpretación. Madrid, 9-11 de febrero de 2005*. Madrid: AIETI, pp. 25-43. ISBN 84-8468-151-3. Versión electrónica disponible en la web de la AIETI: <http://www.aieti.eu/pubs/actas/II/AIETI_2_ODF_Software.pdf>.



Software Libre en la formación de traductores: entre el pragmatismo y la utopía¹

Oscar Díaz Fouces
(Universidade de Vigo)

*If you want to accomplish something in the world, idealism is not enough
—you need to choose a method that works to achieve the goal.
In other words, you need to be “pragmatic”. Is the GPL pragmatic? Let’s look at its results.*

Free Software, Free Society: Selected Essays of Richard M. Stallman. Boston: GNU Press (2002), 91.

Globalización y homogeneidad

Parece legítimo relacionar el notable incremento de las actividades de traducción al que estamos asistiendo en los últimos cincuenta años, con el vertiginoso desarrollo experimentado en ese periodo por las tecnologías de la información y las (tele)comunicaciones. Este último fenómeno resulta indisociable, por otro lado, de la extensión masiva de la economía de mercado, después del desplome de la Unión Soviética. El proceso de *globalización* que determina esa red mundial de intereses comerciales y de aplastante homogeneidad ideológica depende, en lo más esencial, de unas poderosas redes de comunicación capaces de hacer llegar los productos, los servicios, la información y, en último extremo, el propio modelo político-económico a todos los rincones del planeta. Por supuesto, la traducción, entendida en su sentido más genuino de mediación interlingüística e intercultural, es una pieza fundamental de ese mecanismo.

Paradójicamente, ese mismo proceso parece poco beneficioso —cuando menos a medio plazo— para la supervivencia de la diversidad lingüística y cultural de la Humanidad. En efecto, a nadie se le oculta que, en la lógica de la economía de mercado los gastos de localización de los productos son asumibles como un coste adicional, del que es conveniente liberar cuanto antes a las transacciones comerciales (como ocurre con todos los costes, empezando por los de mano de obra), en nombre de la maximización de los beneficios. El mercado mundial no parece tener mayor interés en trabajar con grupos heterogéneos de consumidores.

¹Este artículo puede ser libremente distribuido y/o reproducido en los términos que establece la licencia *Creative*

Sin salir del Estado español, encontramos dos buenos ejemplos de lo expuesto en el párrafo anterior en las reticencias de alguna empresa multinacional y de alguna distribuidora cinematográfica para adaptar sus programas informáticos y sus productos audiovisuales a las lenguas minorizadas². La propia Unión Europea podría estar reconsiderando su costosísima opción por el multilingüismo institucional. Las continuas negativas a aceptar como lenguas de trabajo de las instituciones europeas otros idiomas diferentes de los estatales son, desde luego, indicios que apuntan en la misma dirección. Sin entrar aquí en más detalles, pensamos que hay suficientes motivos para sospechar que el nuevo modelo de sociedad global hacia el que caminamos a pasos agigantados tendrá (o tiene) muy poco de ecológico, en lo que al mantenimiento de la diversidad lingüístico-cultural se refiere.

La homogeneización que señalamos no se detiene, por supuesto, en la dimensión lingüístico-cultural más abstracta. Uno de los productos que antes habíamos apuntado, los programas informáticos, constituye un buen ejemplo de esa más que incipiente tendencia uniformizadora global. Desde cualquier punto de vista, la primacía en este mercado del programario desarrollado por cierta empresa de Redmond está fuera de discusión. Este fenómeno resulta especialmente evidente en los productos estratégicos. El mejor ejemplo es, en este caso, el sistema operativo que se ha (con)fundido con la interface que interactúa con los usuarios (el entorno *Windows*, un bloque unitario a partir de la versión *Chicago* o *Windows 95*). En simultáneo, algunos de los productos creados por esa empresa han determinado los diversos modelos de intercambio de documentos, superando incluso algunas especificaciones patrocinadas por ella misma. Por poner un nuevo ejemplo, es muy significativo que los archivos DOC sean en la práctica el formato de intercambio más habitual en documentos de texto, aún cuando esa misma empresa promueva el uso de un formato orientado precisamente al intercambio, el RTF³.

Commons Attribution-NonCommercial-ShareAlike 2.0 <<http://creativecommons.org/licenses/by-nc-sa/2.0/>>.

²Como recuerda J. Mas (2004), “el Govern de la Generalitat va haver de pagar 80.5 millions per la traducció del Windows 98 al català o el Govern basc una quantitat, pràcticament, cinc cops superior per una traducció parcial del Microsoft Office a l'euscar.” Otro buen ejemplo es el arduo debate sobre el Decreto 273/1998 “Sobre mesures de foment de l'oferta cinematogràfica doblada i subtitulada en llengua catalana” del gobierno autonómico catalán, que pretendía establecer cuotas de pantalla para la exhibición de cine en lengua catalana. V. un extenso dossier documental sobre el asunto, recopilado por el *Centre de Documentació en Sociolingüística* de la *Direcció General de Política Lingüística de la Generalitat de Catalunya*, en <<http://www6.gencat.net/llengcat/cds/docs/mono8.doc>>.

³V. también Mas (2002). Sin ir más lejos, la organización de este congreso estableció como requisito indispensable

Dejando de lado el modo en que este tipo de estándares factuales ha determinado el aspecto externo de las aplicaciones desarrolladas en los últimos años, que han debido rendirse sin condiciones a las omnipresentes «ventanas», resulta especialmente inquietante (y especialmente ilustrativa) la manera en la que se ha reducido la diversidad en los sistemas operativos. La cuota de mercado ocupada por *Solaris*, las ramas de *Unix*, *MacOS* y todo el resto de potenciales competidores se ha visto reducida de un modo drástico en los últimos años (aunque, como veremos, con algunos matices). Otro claro exponente de esa tendencia es la depauperación del ecosistema⁴ de los navegadores (*browsers*), otro sector estratégico que determina la relación de los usuarios con Internet, en favor del omnipresente *Explorer*. Seguramente, aunque con más matices, podría decirse algo muy parecido en cuanto a los clientes de correo electrónico o la mensajería instantánea, que han revolucionado los parámetros de la comunicación humana —y no sólo en el mundo occidental— en la última década.

La posición hegemónica de determinados productos en las áreas críticas que representan el corazón de los sistemas informáticos (el sistema operativo) y sus extensiones comunicativas con el resto del mundo ha determinado la labor de los desarrolladores de programas. En consecuencia, el trabajo de la mayoría de estos profesionales ha estado condicionado, por un lado, por el criterio de compatibilidad con el estándar factual que ese tipo de productos representa y, por otra parte, por el modelo de mercado que los acoge. El *software propietario* se define por su distribución a partir de *licencias*, que permiten el uso del mismo para un número determinado de personas y/o de estaciones de trabajo y sólo para ciertos propósitos; que restringen el derecho de copia y de distribución; y que impiden explícitamente cualquier estudio o modificación por parte de usuarios no autorizados del código fuente de los programas.

La generalización del modelo de software propietario ha coincidido en el tiempo con la extensión de la alfabetización informática en el mundo occidental, de modo que la práctica totalidad de las personas que se han socializado académica o profesionalmente en los últimos

para su inclusión en las Actas que las comunicaciones presentadas se entregasen en formato DOC. En honor a la verdad, esta práctica es (afortunadamente) menos habitual en los ámbitos académicos tradicionalmente considerados “científicos”, en los que resulta bastante frecuente trabajar con formatos no propietarios, fundamentalmente (*La*)*TeX*. Quizás merezca la pena reflexionar sobre ello.

⁴El estimulante trabajo de Scacchi (2003) es una buena fuente de inspiración para el uso de este tipo de “metáforas biológicas”.

años lo han hecho a partir de esas premisas. Tanto es así que, en la práctica, la actual situación parece ser la única posible. Nadie se pregunta, cuando adquiere un equipo informático, por qué motivo el sistema operativo que le ofrecen preinstalado es el que es y no otro diferente. La condición de monopolio factual de ciertos productos en el mercado se hace patente en el hecho de que su presencia, a la que nadie presta atención, sea asumida como obvia y natural, como algo para lo que no existiesen alternativas o no tuviese mucho sentido buscarlas.

Por supuesto, tanto la práctica profesional de la traducción, como la capacitación académica de traductores, cuyo periodo de máxima expansión coincide en buena medida con estas últimas décadas, no han quedado al margen de esa tendencia (¿acaso podía ser de otro modo?)

Los profesionales de los servicios lingüísticos utilizan procesadores de textos, programas para administrar glosarios y memorias de traducción, editores de lenguajes de marcas, herramientas de productividad para gestionar proyectos, navegadores de Internet y clientes de correo electrónico, programas de autoedición, de retoque y creación de ficheros gráficos, aplicaciones contables... que raramente escapan al modelo propietario y que, en general, están reducidos a un puñado de marcas comerciales, más o menos dependientes del modelo global de desarrollo que estipula la empresa de Redmond.

Los formadores de traductores, en el mejor de los casos, utilizan en su docencia esos mismos productos, para reproducir en condiciones pedagógicamente adecuadas el entorno profesional. En el peor de los casos, sea por falta de medios o sea por desidia, continuarían anclados en el bolígrafo, el bloc de notas y los diccionarios de papel. Para la segunda situación — si es que se da— poco remedio puede haber. Para la primera sí, por lo menos en lo que a software se refiere. Ese remedio, además, no constituye una alternativa para abaratar costes (una alternativa *pragmática*), sino que implica, en buena medida, poner en cuestión el actual modelo de desarrollo y de mercado del software y, en último extremo, abrir una interesante discusión sobre los sistemas de generación y de intercambio del conocimiento. No parece exagerado calificar de *utópico* a un objetivo de ese calibre.

Y en eso llegó Linux

A mediados de los años 80, un grupo de programadores, liderados por Richard Stallman comenzó a desarrollar el proyecto GNU (*GNU is not UNIX*), con el objetivo de crear un sistema informático libre, una alternativa al modelo de software propietario imperante en aquellos momentos, después de que *Unix*, inicialmente libre también, asumiese esa nueva condición. Para evitar a terceras personas la tentación de patentar sus productos (aprovechándose de la libertad de acceso al código fuente), idearon la GPL (*General Public License*) y constituyeron la FSF (*Free Software Foundation*). El papel de la GPL puede resumirse en la voluntad de preservar las llamadas «cuatro libertades» del software libre⁵:

- 0.Libertad para utilizar el programa para cualquier propósito.
- 1.Libertad para estudiar el funcionamiento del programa y adaptarlo a nuestras necesidades.
- 2.Libertad para distribuir copias del programa.
- 3.Libertad para mejorar el programa y hacer públicas las mejoras, en beneficio de la comunidad.

A diferencia de otras licencias, la GPL se caracteriza por el llamado «efecto viral»: tanto las copias como los trabajos derivados del software que se acoge a ella (las eventuales modificaciones) también deben respetarla⁶. De ese modo, las libertades citadas «contagian» las copias y las nuevas obras derivadas.

A principios de los años 90, un joven estudiante finlandés, Linus Torvalds, diseñó el núcleo (*kernel*) de un sistema operativo inspirado en *Minix*, otro sistema operativo (orientado a la enseñanza), inspirado a su vez en *Unix*. Torvalds decidió colocar su creación en Internet, de modo que otras personas pudieran estudiarlo, utilizarlo y contribuir a mejorar su rendimiento. El proyecto fue recibido con entusiasmo por la comunidad de usuarios, que rápidamente empezaron

⁵V. <<http://www.fsf.org>>, especialmente <<http://www.fsf.org/gpl>>, y <<http://www.gnu.org>>.

⁶Claramente, las características de la GPL la convierten en una especie de antítesis del *copyright*. En realidad, no es infrecuente encontrar utilizado el término *copyleft* para etiquetar este tipo de productos. Para consultar las diferentes licencias de distribución de software y documentación, cfr. <<http://http://www.gnu.org/philosophy/license-list.es.html>>. También es cada vez más frecuente la publicación de otro tipo de obras de creación intelectual (música, literatura, etc.) acogéndose a licencias de características semejantes, que, más de lo que restringir, limitan deliberadamente los derechos de autor en beneficio de la comunidad. Cfr. por ejemplo la licencia *Creative Commons*, a la que se acoge este trabajo, en <<http://creativecommons.org>>.

a colaborar en el desarrollo de lo que después sería conocido como *Linux*, desde todos los rincones del mundo. Aunque el proyecto GNU había empezado a desarrollar su propio núcleo (*Hurd*), lo cierto es que la combinación de *Linux* con las herramientas y aplicaciones GNU resultó ser un robusto, estable y eficaz sistema operativo. Pese a que inicialmente su configuración resultaba complicada, pronto aparecieron las llamadas *distribuciones*, paquetes precompilados de programas organizados alrededor de un núcleo cuya instalación y uso fueron resultando cada vez más sencillos⁷.

En la actualidad, un usuario medio puede instalar una distribución básica *GNU/Linux* en menos de media hora. En cuanto a la curva de aprendizaje, los esfuerzos realizados por los desarrolladores, en el sentido de crear programas capaces de interactuar con el usuario de una forma amigable, han conseguido reducirla sustancialmente. De todos modos, no conviene

⁷Algunos de los ejemplos más conocidos son estos:

Red Hat, que mantiene una línea comercial para empresas (*Red Hat Linux Enterprise*) y otra que desarrolla en colaboración con la comunidad (el *Proyecto Fedora*), orientada al usuario final. El sistema de paquetería de *Red Hat / Fedora* (RPM) es en el que se inspiran otras distribuciones, como *Mandrake*, *Aurox* o *Conectiva*. En realidad, RPM fue elegido el sistema de paquetes estándar por el *Free Standards Group*. La conversión entre diversos sistemas (por ejemplo entre los RPM y los DEB de *Debian*) es posible a través de herramientas como **alien**. *Red Hat* salió a bolsa en 1999 y es, sin duda, un caso que ilustra la viabilidad de la explotación comercial del software libre.

Slackware, una de las distribuciones “históricas”. Es posible encontrar algunas distribuciones antiguas, basadas en *Slackware 3.5*, como *Basic Linux* o *Grey Cat Linux*, que permiten recuperar para su uso equipos obsoletos (las dos citadas pueden trabajar en particiones *vfat*, en equipos *Pentium* o i486 con sólo 8 Mb de RAM (o incluso menos).

Debian es la distribución que mejor encarna el espíritu del software libre. Desarrollada inicialmente por Ian Murdock, su gestión depende íntegramente de una entusiasta comunidad de usuarios de todo el mundo. Mantiene permanente tres versiones de desarrollo, *stable*, *unstable* y *testing* que pueden ser descargadas libremente de sus repositorios. Las versiones estables únicamente son liberadas después de haber pasado un tiempo más que prudencial de pruebas (ya que no existen presiones comerciales para lanzar continuamente versiones nuevas). Algunas conocidas distribuciones españolas, como *Linex* (de la Comunidad Autónoma de Extremadura) o *Guadalinux* (de la Comunidad Autónoma de Andalucía) están basadas en *Debian*.

Knoppix es el ejemplo más representativo de las *live-cd*, distribuciones que pueden ejecutarse directamente desde un cederrón, utilizando la memoria RAM, sin necesidad de emplear para nada el disco duro del ordenador. Es, por supuesto, un magnífico sistema para conocer *GNU/Linux* sin riesgos. Está basada en *Debian* y existen, a su vez, diversas distribuciones basadas en *Knoppix*, como la catalana *Catix* (patrocinada por el *Ajuntament de Barcelona*), la portuguesa *Paipix* (orientada al trabajo científico, desarrollada en la *Faculdade de Ciências* de la *Universidade de Lisboa*), las brasileñas *Kurumin*, *Kalango* o *Famelix*, o la española *Max* (desarrollada por la Comunidad de Madrid).

La referencia imprescindible en lo que a información sobre distribuciones se refiere es la página de *Distrowatch* (<<http://www.distrowatch.com>>), en la que puede encontrarse una guía completa y actualizada de la mayoría de las que existen, aún cuando cada vez es más frecuente la generación de distribuciones personalizadas, en algunos casos a partir de proyectos como *Metadistros* (<<http://metadistros.software-libre.org>>).

engañarse: la migración a *GNU/Linux* es un proceso que lleva su tiempo y que implica un necesario esfuerzo de reciclaje de conocimientos y destrezas y de estudio. En realidad, también en el mundo del software propietario el aprendizaje del funcionamiento de un programa requiere una importante inversión de tiempo... y de dinero. En contrapartida, una persona que se introduce en ese nuevo mundo percibe, desde el primer momento, que está recuperando el control de su ordenador. El incremento en las destrezas del usuario medio de *GNU/Linux* es exponencial, mientras que los usuarios del sistema operativo más conocido tienden a fosilizar en una fase inicial, en la que suelen ser incapaces de resolver problemas relativamente sencillos, si no cuentan con ayuda técnica.

En cuanto a la segunda cuestión, que hemos abordado únicamente de un modo tangencial, la económica, es interesante recordar que software libre *no* es sinónimo de software gratuito. A pesar de que la mayoría de desarrolladores permiten la libre descarga de sus productos a través de Internet, nada impide venderlos (esos mismos productos), respetando escrupulosamente la licencia GPL. En la práctica, ese nicho de mercado es muy reducido (a causa de la libertad de copia y la diversificación de las fuentes que proporcionan el producto), y se justifica más bien por la comodidad de disponer de una distribución entera o de un buen número de programas en uno o varios cederróns, sin necesidad de descargarlos de Internet. Los precios son muy reveladores, desde luego. En este mes de Noviembre de 2004 es posible adquirir en los quioscos españoles una buena y completa distribución basada en *Red Hat*, en siete cederróns que acompañan a una revista, por el precio total de nueve euros. Esa distribución incluye una potente suite ofimática (y muchos otros recursos de esa categoría), programas de diseño vectorial y de creación de gráficos, juegos, reproductores de vídeo y audio digital, grabadores de CD y DVD, navegadores de Internet, clientes de correo electrónico, entornos de programación, servidores HTTP, editores HTML, gestores de productividad, recursos educativos, documentación y un impresionante número de programas de todo tipo (además, por supuesto, del sistema operativo). Compárese con el gasto equivalente en software propietario y añádase la posibilidad que se da al usuario de realizar libremente copias de esos programas y de distribuirlos. Sobran los comentarios.⁸

⁸ Por supuesto, en la elección de software interviene otro tipo de criterios, además de los económicos. Según las estimaciones de González Barahona, Seoane & Robles (2003:299), *Microsoft Windows 2000* contaba con 29.000.000 líneas de código escritas; *Red Hat Linux 8.0* tenía 50.000.000; y *Debian 3.0* “Woody” nada menos que 105.000.000.

Herramientas libres para los servicios lingüísticos

Unos párrafos más arriba nos hemos referido a las herramientas informáticas que resultan más habituales en la práctica profesional de la traducción. En síntesis, podríamos concretarlas en las siguientes (además, por supuesto, del sistema operativo):

- ✂ procesadores de texto, marcado o no (por ejemplo para editar texto puro o para localizar archivos HTML);
- ✂ programas para el tratamiento/retoque de gráficos (por ejemplo para localizar el texto o la imagen de archivos gráficos);
- ✂ programas para tratar/convertir formatos específicos (por ejemplo, ficheros PDF)
- ✂ programas auxiliares para elaborar presupuestos o para efectuar y mantener asientos contables (por ejemplo, contadores de palabras u hojas de cálculo);
- ✂ programas auxiliares para gestionar el flujo de trabajo (por ejemplo, agendas de clientes o diagramas de Gantt para administrar proyectos);
- ✂ programas para aumentar la productividad (por ejemplo, programas para gestionar glosarios y memorias de traducción);
- ✂ programas para gestionar la información en la Red (navegadores, clientes de correo electrónico o de mensajería instantánea).

Además de esas aplicaciones específicas, un traductor puede disponer de una variada gama de recursos, dependiendo de su propia especialidad o de las características más específicas de su trabajo. Por ejemplo, la localización de contenidos de Internet o la de presentaciones multimedia pueden implicar la creación de subtítulos para *clips* de vídeo digital, el tratamiento de archivos sonoros o la traducción de diapositivas (*slides*); la localización de software requiere de programas específicos para ese cometido, etc.

Las primeras preguntas que podemos plantearnos serían, seguramente, las siguientes: ¿existen recursos libres capaces de sustituir adecuadamente al software propietario en todas esas funciones? y ¿es posible mantener una dedicación profesional continuada en esas condiciones? Veámoslo a continuación.

OpenOffice (<<http://www.openoffice.org>>) es una suite ofimática que integra procesador de textos (*OOWriter*), presentaciones (*OoImpress*), hoja de cálculo (*OOCalc*) y diseño vectorial (*OODraw*). Se trata, en realidad, de la versión libre y gratuita del paquete *StarOffice* (a partir de la v. 5.2), creado por *SunMicrosystems*, que decidió liberar esta versión en una estrategia

comercial ciertamente novedosa. La existencia de un producto paralelo al comercial, que cuenta con la colaboración de usuarios para su desarrollo y su localización a diversas lenguas (como es habitual en el software libre) permite contar con un magnífico banco de pruebas para su «hermano».

OpenOffice no monopoliza, en absoluto, la oferta de programas orientados al procesamiento de textos (por motivos obvios, una de las áreas críticas en las profesiones lingüísticas). Otro buen ejemplo es *Abiword* (<<http://www.abisource.com>>) que, como el anterior, cuenta con versiones para los entornos *Windows* y *Linux*, permite importar y exportar documentos a los formatos más habituales (incluyendo DOC), aunque en este caso no permite la conversión directa a PDF, que sí admite *OOoffice* (pero sí a TEX, lo que hace posible obtenerlos compilando el resultado con **pdflatex** –v. *infra*). Una ventaja sobre *OOWriter* (el procesador de textos de *OOoffice*) es que *Abiword* es mucho más ligero. No obstante, su nivel actual de desarrollo es claramente inferior. *Abiword* está integrado en GNOME (*GNU Network Object Modelling Environment*, v. <<http://www.gnome.org>>), un entorno de escritorio que pretende facilitar la accesibilidad de *GNU/Linux*. Juntamente con KDE (*K Desktop Environment*, vid. <<http://www.kde.org>>) es, sin duda el más conocido de este tipo de productos, aún cuando exista un buen número de alternativas, que generalmente consumen menos recursos del sistema y son por ello más adecuadas para reciclar equipos antiguos (por ejemplo **twm**, **icewm**, **blackbox** o, con mayor consumo de recursos, **wmaker**). Aunque *OOWriter* y *Abiword* sean los dos ejemplos más populares, el mundo del software libre ofrece también una cantidad ingente de alternativas en este ámbito (en realidad, como el lector podrá intuir, la existencia de diversas alternativas para los mismos propósitos es una constante del mundo *GNU/Linux*). Sin ir más lejos, el entorno KDE cuenta también con su propia suite ofimática, *KOffice*, que dispone de procesador de textos (*KWrite*), un hoja de cálculo (*KSpread*) y un gestor de presentaciones (*KPresenter*), que podemos obtener en <<http://www.koffice.org>>.

Uno de los productos estrellas del software libre, sobradamente conocido en entornos propietarios, es el software para *tipografiar* documentos *TeX*, desarrollado por el matemático Donald Knuth entre 1977 y 1978. Se trata, en realidad, de un lenguaje de descripción de página, orientado a la creación de textos de gran calidad, especialmente (aunque no sólo) de tipo

científico, por su gran eficacia para reproducir fórmulas matemáticas. En 1982, Leslie Lamport creó *LaTeX*, un conjunto de macros que permiten simplificar el uso de *TeX*. Con este software, el autor deja de tener que preocuparse por la forma de los documentos y puede pasar a concentrar su atención en el contenido, a diferencia de lo que ocurre con los procesadores WYSIWYG (*What You See Is What You Get*, «el resultado final será el que estás viendo»), como los que antes presentábamos. *LaTeX* es multiplataforma. Puede correr tanto en el entorno *Windows* (utilizando interfaces amigables, como *MikTeX* <<http://www.miktex.org>> y *WinEdt* <<http://www.winedt.com>>), como en *Apple* (con *iTeXMac*, por ejemplo <<http://itexmac.sourceforge.net>>) o, por supuesto, *GNU/Linux*, con el paquete **tetex** y, si lo preferimos, con los interfaces de usuario *Lyx* <<http://www.lyx.org>> o *Kile* <<http://kile.sourceforge.net>>. Es posible generar directamente archivos PDF a partir de documentos TEX, compilándolos con **pdflatex**, así como también generar presentaciones (*slides*), gestionar bibliografías extensas (con **bibtex**) o crear documentos HTML. *LaTeX* permite a cualquier usuario desarrollar tareas de edición de carácter profesional, sin ningún requisito de hardware especial y sin ningún coste. A propósito de la edición, no podemos dejar de citar aquí a *Scribus* (<<http://www.scribus.com>>), una herramienta multiplataforma que permite maquetar y publicar documentos complejos (carteles, folletos, revistas) sin desmerecer a otros rivales propietarios.

Por supuesto, también contamos con alternativas libres para crear y editar documentos escritos en (otros) lenguajes de marcas. Podemos trabajar cómodamente con documentos HTML a partir de programas no comerciales y de código abierto, como *Quanta* <<http://quanta.sourceforge.net>>, *Bluefish* <<http://bluefish.openoffice.nl>>, *Screem* <<http://www.screem.org>> o *NVU* <<http://www.nvu.org>>. Algunos de ellos permiten únicamente interactuar con el código (como *Screem*). Con otros, como *Quanta*, podemos ir analizando los resultados parciales en modo *wysiwyg*. En este caso, contamos además con una serie de opciones que permiten automatizar las tareas más farragosas (la creación de tablas, la inserción de gráficos, la gestión de hiperenlaces...) y que simplifican las más específicas (como la modificación de las DTD); *Bluefish* permite personalizar las barras de tareas. *NVU* es un producto multiplataforma, que cuenta con una versión para el entorno *Windows*. Conviene recordar, una vez más, que no

pretendemos (ni mucho menos) agotar el repertorio de alternativas. Sin ir más lejos, *OOoffice* permite editar también documentos HTML y algunos editores del entorno *GNU/Linux* (el mejor ejemplo es, sin duda, **emacs**) simplifican el trabajo con este tipo de código mediante el uso de algunas prestaciones adicionales (como el coloreado sintáctico).

Ya que hemos mencionado al código HTML, será inevitable apuntar aquí que contamos con un amplísimo abanico de alternativas libres para escoger navegadores de Internet, clientes de correo electrónico, gestores de transferencia de ficheros (FTP), clientes de *chat* y videoconferencia, mensajería instantánea y, en general el tipo de productos que un profesional de los servicios lingüísticos podría necesitar utilizar. En cuanto a los dos primeros, recordaremos ahora que el actual *Mozilla* (<<http://www.mozilla.org>>) no es más que el resultado de la liberación del código fuente del antiguo *Netscape* que, en los primeros tiempos de la Red, fue, sin duda, el software más extendido para viajar por ella. El actual *Mozilla* cuenta con navegador web, editor HTML y un cliente de correo electrónico, con un buen sistema preconfigurado para filtrar el correo no deseado (*spam*), es multiplataforma y dispone de uno de los mejores soportes a las especificaciones del W3C (*World Wide Web Consortium* <<http://www.w3c.org>>) en cuanto a hojas de estilo (CSS), lenguajes de *scripts*, etc. Sin embargo, y como quiera que se trata de un producto ciertamente «pesado» (consume una buena cantidad de los recursos del sistema), la *Fundación Mozilla* está embarcada en el proyecto de desarrollar programas más ligeros. En estas fechas contamos ya con un navegador (*Firefox*), un cliente de correo (*Thunderbird*) y un gestor de calendario (*Sunbird*) que son magníficas alternativas al producto anterior. Uno de los mejores ejemplos de integración de cliente de correo y organizador de tareas (calendario, agenda, etc.) es *Evolution* (<<http://www.novell.com/products/evolution>>), que formó parte del módulo *Ximian* desarrollado por Miguel de Ícaza, orientado al mundo empresarial⁹, integrado en GNOME. A propósito de GNOME, y para no extendernos más, conviene apuntar que este entorno de escritorio (creado en 1997) cuenta él sólo con una aplicación ofimática (*Abiword*) y un editor web (*Bluefish*) a los que ya nos hemos referido, además de un navegador web (*Epiphany* <<http://www.gnome.org/projects/epiphany>>), cliente de mensajería instantánea (*Gaim*, en

⁹ *Ximian* (inicialmente *Hélis*) fue adquirida por la empresa *Novell* en 2003. El mejicano Miguel de Ícaza es también el desarrollador del programa *Midnight Commander*, un utilísimo clon del *Norton Commander* con que contaban los usuarios de *MSDOS*, y uno de los “padres” del proyecto GNOME.

<<http://gaim.sourceforge.net>>, que permite asociar cuentas de *Napster*, *Yahoo*, *MSN*, etc.) y una aplicación de videoconferencia (*GnomeMeeting* <<http://www.gnomemeeting.org>>, que permite efectuar llamadas de PC a teléfono, VoIP, etc.), por no salirnos del tipo de aplicaciones a las que estamos haciendo referencia en estos momentos. Obviamente, también el proyecto KDE cuenta con su propio navegador (*Konqueror* <<http://kde.konqueror.org>>) y cliente de correo electrónico (*Kmail* <<http://kde.kmail.org>>) integrados.

Dentro aún del ámbito de las comunicaciones por Internet, no podemos dejar de señalar que el de los servidores es uno de los nichos en que el software libre tiene una presencia muy significativa, como anticipábamos en los primeros párrafos de este trabajo. Efectivamente, la configuración *LAMP* (*GNU/Linux* + *Apache* + *MySQL* + *PHP*), por su estabilidad, robustez y economía, es claramente una buena opción para este tipo de sistemas. Con ella, disponemos de un sistema operativo (*GNU/Linux*) que se caracteriza por poder trabajar ininterrumpidamente sin padecer las frecuentes «caídas» a las que nos tienen (desgraciadamente) acostumbrados otros competidores; con un software de servidor multiplataforma (*Apache*) que trabaja sobre un potente gestor de base de datos (*MySQL*) y un lenguaje de programación que nos permite generar dinámicamente nuestro código (*PHP*). La mayor parte de las distribuciones *GNU/Linux* disponen de un sistema *LAMP* preconfigurado y listo para que lo utilicemos. De no ser así, siempre podemos descargar los componentes para la arquitectura adecuada (*Windows*, *GNU/Linux*...) desde <<http://httpd.apache.org/download.cgi>>, <<http://www.mysql.com/downloads>> y <<http://www.php.org>>. En lo que a nosotros nos interesa, conviene apuntar que cualquier usuario que disponga de una conexión permanente a Internet está en condiciones de configurar su propio servidor web (incluso reciclando equipos antiguos) e instalar sobre él su propio servidor de correo corporativo y, por supuesto, alguna aplicación de comercio electrónico (como *osCommerce* <<http://www.oscommerce.org>>) para ofrecer sus servicios y poder interactuar fácilmente con clientes, intermediarios o agentes subcontratados.

Acabamos de hacer referencia a las transacciones económicas, tan propias de las profesiones lingüísticas como de cualquier otra, y parece lógico referirnos ahora a las aplicaciones relacionadas con la contabilidad, la facturación, el control de pagos o el registro de clientes. Por supuesto, también contamos con buenas alternativas libres. Un buen ejemplo es

GNUCash (<<http://www.gnucash.org>>), que trabaja a partir de un sistema de doble entrada (a cada cambio en la hoja de gastos debe corresponderse otro en la hoja de ingresos) que mantiene una total transparencia contable. Permite generar informes sobre el estado de nuestras cuentas por tipo de gasto o ingreso, activos y pasivos, inversiones, etc. *Fisterra* (<<http://www.fisterra.org>> es una completa herramienta para la planificación de recursos empresariales (en inglés, ERP *Enterprise Resource Planner*), que combina el control de facturación, la gestión de pedidos, pagos, *stocks*, albaranes, información contable, etc. *Gestión LinEx* (<<http://www.linex.org>>) tiene unas características semejantes. Se trata de una herramienta integrada por dos aplicaciones, *ContaLinEx* y *FacturLinEx*. La primera sirve para administrar un terminal de punto de venta (TPV) y la segunda permite una gestión contable integral de cualquier tipo de empresa (Libro Mayor, Libro Diario, etc.) *Gestión LinEx* está relacionada con el proyecto *LinEx* (una distribución basada en *Debian*, para usuarios finales), patrocinado por la Junta de Extremadura, que cuenta con una versión orientada al mercado empresarial, *LinEx Empresa*. En el caso que nos ocupa, en el que el contacto con los clientes no se desarrolla habitualmente en situaciones de interacción que son más propias del comercio de bienes de consumo físico, una buena alternativa es *Galopín* (<<http://www.galopin.sinuh.org>>), basada precisamente en PHP y MySQL. En cuanto a los programas para administrar el trabajo en equipo (*groupware*), podemos citar al antaño propietario *Hipergate* (<<http://www.hipergate.org>>), que cuenta con herramientas de gestión de clientes, planificación de reuniones y gestión de proyectos o a *Opengroupware.org* (<<http://www.opengroupware.org>>), que permite a las personas que participan en un proyecto compartir los datos de calendarios, reuniones y carpetas, conectarse mediante mensajería instantánea, etc.

La joya del programario libre en materia de creación de diseño gráfico es, sin duda, *The Gimp* (<<http://www.gimp.org>>). Mucho más que un programa de retoque fotográfico, permite trabajar con diversas capas (y, por lo tanto, es una útil herramienta para la localización de ficheros gráficos que incluyen texto), importar y exportar gráficos en multitud de formatos, aplicar efectos... *Gimp*, que también es multiplataforma, tiene una versión para animación gráfica, *FilmGimp*¹⁰. Para el diseño vectorial contamos con herramientas como *Sodipodi*

¹⁰ *FilmGimp* habría sido la herramienta utilizada en la producción de la película *ScoobyDoo*, de acuerdo con la

(<<http://www.sodipodi.org>>), utilísimo para la creación de contenidos para Internet, por utilizar como formato nativo el SVG del W3C; *Blender* (<<http://www.blender.org>>) está orientado a la creación de gráficos y animaciones en 3D; *Qcad* (<<http://www.ribbonsoft.com>>) es una buena aplicación CAD para el entorno *GNU/Linux*.

Lamentablemente, el capítulo de las herramientas específicas para la traducción no es el más desarrollado en el ámbito del software libre. Una prometedora excepción es *OmegaT* (<<http://www.omegat.org>>), un gestor de memorias de traducción desarrollado en el lenguaje *Java* de *Sun Microsystems* y que, por lo tanto, requiere para su instalación y uso que contemos con el entorno de desarrollo *Java (Java Runtime Environment)* v. 1.2 o posterior (pueden descargarse gratuitamente desde <<http://java.sun.com/j2se> >). La apariencia es aún muy simple, aunque efectiva, y todo parece indicar que todavía estamos en los primeros estadios de su evolución (de hecho, apareció en 2001). Claro que hay que recordar, para ser justos, que estamos hablando de un producto libre (cualquier usuario puede modificar el código a su antojo) y gratuito (cualquier persona que haya pagado por una licencia comercial de un gestor de memorias de traducción sabe a qué me refiero). La segmentación tiene lugar a nivel de párrafos y no de frases, como es más habitual. Esa opción presenta, desde luego, ventajas (el contexto es más extenso) y también inconvenientes (las correspondencias parciales o exactas son mucho menos frecuentes). Admite los formatos de texto puro, HTML y los nativos de *OOoffice* y *StarOffice*, de modo que los ficheros generados con *OmegaT* pueden ser modificados con un editor XML, ya que el formato nativo de *OOoffice*, que es libre, sigue también las especificaciones XML. Este hecho no es trivial, desde luego. Dado que *OOoffice* permite exportar documentos DOC para su propio formato nativo, podemos extraer de esos documentos la información XML (con cualquier [des]compresor de ficheros) y trabajar sobre ella (renombrándola como HTML). *OmegaT* no se integra con editores externos (y, en ese sentido, el sistema de trabajo es más similar al de *DéjàVu* que al del *Translator's Workbench* de *Trados*). En realidad, el único ejemplo que conocemos de intento de relacionar un gestor de memorias de traducción con un procesador de textos libres es el de *OOxlate* (<<http://translatewrite.com/ooxlate.html>>), que pretende integrarse con *OOoffice* aunque aún se encuentra en una fase inicial de desarrollo. El lector puede encontrar un análisis bastante

información aparecida en *Linux Journal* (v. <<http://www.linuxjournal.com/article/6148>>).

imparcial (aunque el autor no oculte sus preferencias) de los pros y los contras de *OmegaT* en Prior (2003). El mismo autor mantiene una interesante página sobre recursos libres para traductores en <http://www.linuxfortranslators.org>.

Existe una alternativa comercial muy reciente en el campo de los gestores de memorias de traducción para *GNU/Linux*. Se trata de *Heartsome TMX Editor*, que permite editar memorias de traducción que utilicen el formato de intercambio estándar TMX y *Heartsome XLIFF Translation Editor*, que trabaja a partir de ficheros XLIFF (una especificación XML), con los que podemos conseguir la mayoría de las prestaciones que pueden esperarse de un programa profesional de estas características (diferentes tipos de segmentación, validación de segmentos, introducción de notas, consulta de contextos, etc). *Heartsome* tampoco se integra con procesadores de texto externos, sino que cuenta con su propio editor. El hecho de que exista un producto comercial de estas características, en un área que parece ser, de momento, una de las menos atendidas en el ámbito del software libre nos hace pensar que lo mejor en este campo aún está por llegar (v. <http://www.heartsome.net>).

La localización es una de las estrellas del software libre. Las comunidades organizadas alrededor de las diferentes distribuciones, o incluso de los diferentes productos (el caso de *OOoffice* es paradigmático, en este sentido) suelen trabajar a buen ritmo para mantener actualizadas, en la medida de lo posible, versiones de los mismos en una impresionante cantidad de lenguas. Conviene aclarar, por lo tanto, que este tipo de trabajo tiene, prácticamente siempre, un carácter voluntario y gratuito (a diferencia de lo que ocurre con el software propietario), lo que no debería sorprender a nadie, considerando que el trabajo de los desarrolladores del software también lo es. El programario libre es localizado a partir de ficheros PO, de acuerdo con una serie de convenciones que los diferentes equipos intentan respetar¹¹. Para ello existen diversas aplicaciones, como *Kbabel* (<http://kbabel.kde.org>) que, en realidad, no son más de lo que gestores de memorias de traducción (que comparten los miembros de los diferentes equipos) y que, sin duda, podrían ser fácilmente adaptados como herramientas de traducción de propósito más general. Es interesante señalar aquí que el resultado del trabajo de esos equipos es libre, de

¹¹ V. por ejemplo el documento “Estructura dels fitxers PO –v0.1 (15-01-2004)” utilizado por el equipo vinculado a *Softcatalà* que trabaja en la localización de software libre para la lengua catalana (puede descargarse desde <http://www.softcatala.org>).

modo que es posible para cualquier persona obtener en línea impresionantes memorias multilingües con el trabajo de todos esos colaboradores voluntarios.

Muy bien, ¿y ahora qué?

En el apartado anterior hemos presentado una serie de aplicaciones que pueden representar alternativas maduras a diferentes programas propietarios en las actividades relacionadas con los servicios lingüísticos. Llegados a este punto, y dadas las limitaciones de un trabajo de estas características, no podemos por menos que intentar destilar algunas conclusiones. Podrían ser estas que siguen:

- ✍ El software libre ha alcanzado ya un grado de desarrollo que lo convierte en una alternativa plausible en muchas de las tareas que rodean la actividad de los profesionales de los servicios lingüísticos.¹²
- ✍ Esa condición de alternativa tiene que ver con diversos factores, y el precio es tan solo uno de ellos (aunque, desde luego, es importante). El software libre ha demostrado estabilidad, robustez y calidad de ejecución en entornos tan críticos como el de los servidores. La posibilidad de realizar copias y de acceder al código fuente, para modificarlo o encargar modificaciones a precios competitivos, no pueden dejar de ser tenidas en cuenta.
- ✍ Por lo tanto, la introducción de este tipo de material en la formación de los nuevos profesionales de los servicios lingüísticos parece no sólo conveniente, sino incluso necesaria, si lo que se pretende es lanzar al mercado traductores, editores o redactores técnicos con capacidad para competir por un espacio profesional sometido a unas condiciones cada vez más exigentes.
- ✍ En el momento actual, sin embargo, sería absolutamente ingenuo pretender que un profesional pueda salir al mercado si su formación prescinde por completo de contenidos relacionados con el software propietario, entre otros motivos porque los contratadores continúan exigiendo, de forma prácticamente unánime, el trabajo con ese tipo de software. Lamentablemente, además, el de los gestores de memorias de traducción es uno de los

¹² V. en McKay (2004) un buen listado de argumentos para el uso del software libre en la práctica profesional de los

escasos ámbitos en los que el software libre compite en desventaja con el software propietario, por lo menos ahora mismo¹³. Por supuesto, esa desventaja es relativa, y tiene que ver más con la diversidad de la oferta que con el precio, más con la primacía actual de los formatos propietarios que con la disponibilidad y flexibilidad de los productos. Por lo tanto, sería tan ingenuo ignorar esa desventaja como tendencioso no valorarla en su justa medida.

A la vista de todo lo anterior, nuestra recomendación sería incorporar progresivamente el software libre en la formación de profesionales de los servicios lingüísticos, en paralelo con el software propietario, de modo que los formandos pudiesen ir escogiendo lo mejor de ambos mundos. En realidad, esa es, en buena medida, la estrategia por la que están optando muchos profesionales consolidados. Nada impide optar por soluciones híbridas. De hecho una buena formación de profesionales de servicios lingüísticos debe tender a priorizar la busca de soluciones adecuadas a cada situación y, por lo tanto, a diversificar la oferta también en el ámbito propietario. Existen en estos momentos magníficas alternativas *shareware* y *freeware* para continuar trabajando en plataformas propietarias capaces de competir con programas comerciales más consolidados¹⁴. Es perfectamente posible compatibilizar el uso de software propietario con software libre: una distribución *GNU/Linux* puede servir como plataforma para utilizar software propietario utilizando paquetes como *CrossoverOffice* (<<http://www.codeweavers.com>>), un producto comercial, aunque de bajo coste. Por otro lado, nada impide utilizar software libre en entornos propietarios (v. una buena compilación de programas de estas características en <<http://gnuwin.epfl.ch>>). Existe aún una última alternativa, técnicamente muy simple, que es la de mantener un sistema libre y otro propietario configurados para arrancar con cualquiera de ellos en dos particiones del mismo disco duro (en *dual boot*). Todas esas posibilidades permiten establecer una clara progresión pedagógica en la modificación de los hábitos de uso y, eventualmente, en la transición desde entornos propietarios a entornos libres.

traductores.

¹³ Cfr., a título de ejemplo para la situación en otros ámbitos, el trabajo de González Barahona (2002).

¹⁴ En Díaz Fouces (2004) hemos desarrollado con algo más de detalle ese tema.

No debemos perder de vista, además, que también es responsabilidad de los formadores educar la sensibilidad de los formandos en este tipo de asuntos: la educación nunca es neutral. La opción de priorizar los formatos libres, de presentar una oferta diversificada de productos y de estimular el intercambio libre del conocimiento, responde a valores formativos que van más allá de los objetivos estrictamente pragmáticos, aunque estos no puedan abandonarse. En el mismo sentido, el fomento de la investigación en este ámbito podría llevar, a muy corto plazo, a colmar las lagunas que ahora podemos detectar (como en el caso de las memorias de traducción): la disponibilidad del código fuente permite contar con toda la comunidad mundial de usuarios como potenciales cooperantes. A los formadores corresponde también contribuir a crear la necesidad que estimule este tipo de trabajos, liderarlos, participar en ellos o simplemente darles publicidad.

A propósito del papel de los formadores, aunque aquí no hayamos insistido en ello, debemos recordar que existe un importantísimo volumen de software libre orientado a la formación (especialmente la formación *on-line*): gestores de foros de discusión, de exámenes, de distribución de material o de tutorías¹⁵. En el caso de los centros públicos, en los que la optimización y la gestión eficiente de los recursos deben ser una prioridad, parece obvio que ese tipo de productos tendría que ser objeto de especial atención¹⁶.

Demostrar a los estudiantes la eficacia del software libre y su eventual situación de ventaja sobre el software propietario en todas las situaciones en las que ello sea posible (y, desde luego, no son pocas) supone que pongamos una actitud pragmática al servicio de unos valores éticos. Y esos dos factores son, seguramente, dos pilares básicos de la formación académica.

¹⁵ V. el sitio del proyecto *Join* <<http://www.ossite.org/join/sp>>, un interesante proyecto de investigación que pretende compilar una base de datos con todo ese tipo de aplicaciones, clasificadas de acuerdo con una serie de criterios de calidad.

¹⁶ V. Amatriain (2004).

Referencias bibliográficas¹⁷

Amatriain, X. 2004. “El Programari Lliure a l'Educació: guia per a la seva justificació i implementació”. Comunicación para las *III Jornades de Programari Lliure*. Escola Universitària Politècnica de Manresa. Universitat Politècnica de Catalunya. En línea <<http://jornadespl.upc.es>>

Díaz Fouces, O. 2004. “Novas ferramentas para (ensinar a) traduzir”. C. Garrido (ed.) *Ferramentas para a tradução*. Corunha: Associação Galega da Língua, 143-171.

González Barahona, J.M. 2002. “Software libre en la enseñanza de la informática”. *TodoLinux* 23:12-13. Reproducido en V. Matellán e.a. *Sobre software libre. Compilación de ensayos sobre software libre*. Universidad Rey Juan Carlos. Escuela Superior de Ciencias Experimentales y Tecnología. Departamento de Informática, Estadística y Telemática, 67-72. En línea <<http://gsync.escet.urjc.es/~grex/sobre-libre/libro-libre.pdf>>

González Barahona, J.M., J. Seoane Pascual & G. Robles. 2003. *Introducción al software libre*. Barcelona: Fundació per a la Universitat Oberta de Catalunya. En línea <<http://www.uoc.edu/masters/softwarelibre/esp/materiales.html>>

McKay, C. 2004. “Free and Open Source Software for Translators”. Comunicación para la *45th Annual Conference. American Translators Association*. Toronto, Canadá. En línea <<http://translatewrite.com/ATApaper.html>>

Mas, J. 2002. “Els estàndards oberts garanteixen la llibertat a escollir”. En línea <<http://www.softcatala.org/articles/article11.htm>>

Mas, J. 2004. “Les cultures minoritàries i les multinacionals del software”. En línea <<http://www.softcatala.org/articles/article38.htm>>

Prior, M. 2003. “Close Windows. Open Doors”. *Translation Journal* 7(1). En línea <<http://www accurapid.com/journal/23linux.htm>>

Scacchi, W. 2003. “Understanding Free/Open Source Software Evolution: Applying, Breaking and Rethinking the Laws of Software Evolution”. En línea <http://opensource.mit.edu/online_papers.php>

¹⁷ Las referencias a documentos de Internet citadas en este trabajo han sido verificadas a 15.12.2004.