

<title>

# El software libre en la creación visual digital

</title>

Una vez explorado algunas problemáticas del uso de la “caja negra”, este capítulo pretende hablar de la historia, características, problemáticas y ventajas del uso de un tipo de software distinto al usado comercialmente. El objetivo es reconocer el origen, definición y características del software libre y las implicaciones de su uso en el terreno de la creación visual digital; así mismo, la diferenciación con la “caja negra” y el porqué puede representar una alternativa “abierta y transparente”.

## 2.1.Un poco más de historia, surge una contra-propuesta, el movimiento de software libre.

### 2.1.1. Richard Stallman y el proyecto GNU. Linus Torvalds y LINUX

Siguiendo el tema de algunas problemáticas con el uso de la “caja negra”, nos referiremos a una anécdota que marca el inicio de una contrapropuesta al software de patente y su consecuente ocultamiento del código fuente.

En la década de los 70's, Richard Matthew Stallman<sup>1</sup>, programador estadounidense del Laboratorio de Inteligencia Artificial (IA LAB) del Massachusetts Institute of Technology (MIT), cuenta que el laboratorio había recibido una impresora donada por la Compañía XEROX, que era una versión modificada de una fotocopidora de la misma empresa. Dicha impresora, era usada por todos los trabajadores del Laboratorio, de tal suerte que todos podían mandar a imprimir a esta impresora desde sus computadoras conectadas en red. Sin embargo, algo parecía no estar funcionando muy bien a pesar de tratarse de un prototipo de vanguardia. Uno de sus principales problemas era que cada cierto tiempo el papel se atascaba y no se le notificaba al usuario de la situación con ningún aviso enviado por red. Evidentemente la pérdida de tiempo se volvió una constante, ya que en varias ocasiones, los trabajadores enviaban sus trabajos a imprimir y al ir a buscarlos se encontraban la impresora atascada y una cola enorme de trabajos pendientes.

Stallman quiso resolver el problema de la manera que estaba a su alcance siendo un programador y eso era, haciendo código. Se le ocurrió que una manera de resolverlo era haciendo que la impresora enviara por red una notificación de que el papel se había atascado y así, uno o más trabajadores resolverían el problema antes de que la cola de documentos pendientes se hiciera más grande. Para hacer esto, era necesario tener acceso al código fuente de los controladores de la impresora y así poder escribir las instrucciones para que la impresora las ejecutara.

Así que habló con XEROX y les explicó lo que quería hacer. La empresa se negó a darle el código fuente debido a su política de privacidad y propiedad del código. Cuando intentó pedir-

selo a un colega suyo que trabajaba para XEROX, éste le dijo que había prometido a la compañía no entregar una copia del código fuente a nadie, proveniente de un acuerdo contractual con la empresa.<sup>2</sup> "Actualmente este es un ítem estándar en el negocio de la industria de software, pero el acuerdo de no revelar, o NDA por sus siglas en inglés, era un desarrollo nuevo en esa época, un reflejo tanto del valor comercial de la impresora láser de XEROX como de la información requerida para manejarla. "Xerox estaba en esa época tratando de hacer de la impresora láser un producto comercial y habría sido una locura el entregar el código fuente"<sup>3</sup>.

El software se había convertido en una posesión tan valiosa que las compañías dejaron de sentir la necesidad de publicar el código fuente, especialmente cuando la publicación significaba entregar a potenciales competidores la oportunidad de duplicar algo de manera económica. Desde el punto de vista de Stallman, la impresora era un caballo de Troya. Después de una década de fracaso, el software apropiado privadamente había ganado una posición firme dentro del Laboratorio de IA a través uno de los métodos más solapados. Había llegado disfrazado de regalo.

"Fue mi primer encuentro con un acuerdo de no revelar (NDA), e inmediatamente me enseñó que los acuerdos de no revelar tenían víctimas," dice Stallman. "En este caso yo fui una víctima. Mi laboratorio y yo fuimos víctimas."<sup>4</sup>

Richard Stallman se fijó firmemente defender la práctica del intercambio y la "libertad" de los usuarios del software, tal como lo menciona Sam Williams en *Libre como en libertad*:

"Desde ese día en adelante, decidí que esto era algo en lo cual yo nunca participaría," dice Stallman, "Decidí nunca convertir a otras personas en víctimas así como yo había sido una víctima."<sup>5</sup>

A partir de esta experiencia, Richard Stallman tomó la iniciativa en 1984 de escribir un sistema operativo que denominó GNU, cuya característica principal consistiría en que cada parte que lo compone y lo hace funcionar fuera realizado por una comunidad de programadores que lo mejorara y compartiera, así cada persona que tuviera acceso a él podría modificarlo y adaptarlo según sus necesidades y plataforma que se utilizara, aspirando así a crear el sistema operativo perfecto a través de un principio básico de compartición del conocimiento<sup>6</sup> y

lejos de los acuerdos de confidencialidad. En 1985, Stallman fundó la Free Software Foundation (Fundación Software Libre) mientras que GNU comenzaba a utilizarse entre algunas comunidades de programadores. Inició el movimiento del **software libre**<sup>7</sup>

GNU es un acrónimo recursivo que significa Gnu is Not Unix o Gnu no es Unix. *UNIX* se refiere al sistema operativo popular de los 70's que posee una patente y del que hablamos en el capítulo anterior, con el cual GNU es compatible, pero no resulta idéntico ya que, contrario a UNIX, GNU puede ser modificado por cualquier persona. No obstante, a ningún distribuidor se le permitirá restringir su redistribución posterior. Es decir, "no estarán permitidas modificaciones propietarias. Hay que asegurarse de que todas las versiones de GNU permanezcan libres"<sup>8</sup>.

Paulatinamente, el proyecto GNU causó una transmisión acelerada de información por compartición, ya que varios programadores decidieron contribuir con él. Fue en 1991 cuando el finlandés Linus Torvalds programó el núcleo y la parte fundamental para el sistema operativo GNU, el kernel, que fue llamado LINUX, basándose en un sistema operativo con fines didácticos, Minix. Hoy día, LINUX es uno de los programas libres más utilizados para el sistema operativo GNU<sup>9</sup>. A partir de que LINUX es liberado, dicho sistema operativo empezó a evolucionar rápidamente debido a que muchos programadores han contribuido a su desarrollo y mejora, lo que lo convierte en uno de los sistemas operativos más estables.

Con GNU y LINUX juntos, comienza uno de los crecimientos más importantes del software libre, la incursión en el mundo empresarial, ya que algunas empresas comenzaron a basar su



Richard Stallman, fundador de la Free Software Foundation y padre del software libre.

estrategia de negocios en hacer consultorías, distribución y servicios alrededor de este nuevo y creciente sistema operativo. La estrategia fue comenzar a vender “misterios” al rededor del software libre los cuales podían ser resueltos por quién supiera de lenguaje de programación y no “secretos” indescifrables que una sola empresa conociera. No se trata de vender el secreto del código, sino la libertad del servicio.

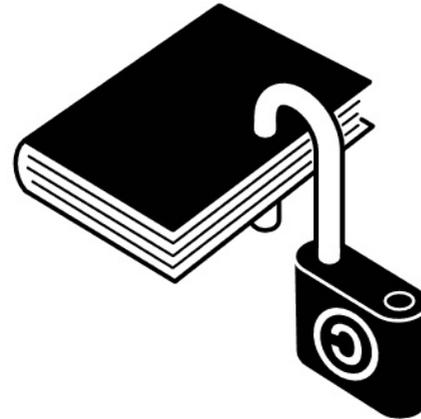
Con estos antecedentes, podemos acercarnos a una definición formal del software libre, cuya ideología está basada en el concepto de “libertad”, tal como se explica a continuación.

### 2.1.2. Definición de Software Libre

El término **Software Libre** o Free Software se refiere a aquellos programas cuyo código está abierto para que cualquier usuario pueda modificarlo, usarlo, estudiarlo y/o distribuirlo sin que esto represente un problema legal. El término “**código abierto**” o en ingles, “open source”, se refiere a que las líneas de texto en las cuales están escritas las instrucciones que sigue la computadora para ejecutar dicho programa, están disponibles para que todos los usuarios puedan conocerlas y así, mejorar o adaptar dicho código según sus necesidades. Sin embargo, el S.L. no se trata de una copia no autorizada ni mucho menos de aquel descargable desde Internet de manera gratuita, el llamado freeware.

El software libre debe poseer la autorización para utilizar un código abierto. Free software y open source como términos distintos pero complementarios. El software libre con sus líneas de código abierto, representa la libertad de que cualquiera puede hurgar en él, lo cual a la larga permitiría que el programa sea modificado e incluso mejorado a una velocidad mucho más eficiente que el desarrollo del software patentado, llamado también software propietario, dando como resultado la creación de un mejor software con la colaboración de muchas personas a la vez, tal como lo explica Raymond en La catedral y el bazar en lo que él llama la ley Linus:

“Dada una base lo suficientemente amplia de probadores y colaboradores, casi todos los problemas se identificarán con rapidez y su solución será obvia para alguien. O expresado con menor formalidad: ‘con un número de ojos suficiente, todos los errores son irrelevantes’<sup>10</sup>



Ahora bien, el término free software fue acuñado en inglés, por lo que, puede existir una confusión en dicho idioma ya que la palabra Free se refiere a "libre" y al mismo tiempo al concepto de "gratis", es por eso que la Free Software Foundation, aclara que:

Candado cerrado y candado abierto como analogía al código privativo y al código abierto, respectivamente.

"El software libre es una cuestión de libertad, no de precio. Para entender el concepto, debería pensar en libre como en libre expresión, no como en barra libre."<sup>11</sup>

Es por eso que el S.L. puede estar disponible para su distribución comercial, sin que eso le quite su calidad del "libre", esto es, que se pudo pagar alguna cantidad para obtener una copia de S.L. para cubrir algunos costos de distribución o bien, obtenerlo sin costo alguno, ambas opciones no afectan que un software siga siendo libre o no siempre y cuando se cumplan las 4 libertades

Estas 4 libertades son a su vez los 4 principios del software libre, son 4 libertades que tienen los usuarios del software y que necesariamente deben de ser cumplidos para que se le considere como tal. Dichas libertades fueron acuñadas por el propio Richard Stallman y posteriormente fueron bandera de la Free Software Foundation.

## **Libertad 0: Usar el software para cualquier propósito**

Dichas libertades son las siguientes:

Libertad 0.- Libertad de usar el programa para cualquier propósito

Libertad 1.- Libertad de estudiar el funcionamiento del programa y adaptarlo a las necesidades. El acceso al código fuente es una condición previa para esto.

Libertad 2.- Libertad de redistribuir copias, y con ello poder ayudar al prójimo

Libertad 3.- La libertad de distribuir copias de sus versiones modificadas a terceros. Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.

De estas libertades y ejemplos prácticos en el quehacer humano hablaremos un poco más adelante.

Así pues, el sistema operativo GNU / LINUX se convierte en uno de los pilares de la comunidad del software libre, una comunidad que defiende y promueve la libertad de usar los programas y adaptarlos para cualquier propósito, de mejorar su programación, de estudiar su funcionamiento y distribuirlo libremente. Puede ser entendido como una alternativa al software de patente y la "privatización" del código, así como una filosofía que no sólo se refiere a programas de computadora, sino que sustenta a un movimiento social, político y de resistencia que defiende ciertas libertades.

### 2.1.3 Las 4 libertades, la ideología detrás del software libre

Ya que mencionamos brevemente cuáles son las libertades del software libre, hablemos específicamente de cada una de ellas y analicemos algunos ejemplos para sustentarlas.

#### Libertad 0

En el caso de la libertad 0, la libertad de usar el programa para cualquier propósito, se refiere a que los usuarios pueden utilizar el programa para cualquier fin sin necesidad de notificarlo a algún programador que haya intervenido en la realización de dicho software, o a

cualquier otra entidad o institución. En esta libertad importa el propósito del usuario y no del programador, de igual suerte que el usuario puede distribuirlo a otras personas y dichos nuevos usuarios utilizarlo para sus propios propósitos.

Ejemplos de cómo algunos programas de patente no permiten esta libertad, encontramos varios, como el caso de Photoshop que impide a los usuarios abrir imágenes detalladas de algunos billetes como el dolar, el euro y la libra esterlina (sucede con otros papel moneda dependiendo de las leyes de cada país). Photoshop ha incluido la tecnología CDS<sup>12</sup> que reconoce los sellos de agua impresos en el billete e imposibilita su edición.<sup>13</sup> Sucede lo mismo con el software del iPod, ya que al comprar una canción en la Apple Store, iTunes, no permite que se copie en un equipo que no sea ese iPod y/o la computadora asociado a su dueño. Otro ejemplo más o menos reciente fue en el 2009 cuando Amazon borró dos libros de George Orwell, "Rebelión en la granja" y "1984" de más de miles de lectores electrónicos Kindle ya vendi-

DRM, Digital Rights Management o gestión de derechos digitales, tecnologías de control de acceso para limitar el uso de medios creativos

dos, gracias al control remoto que poseía la empresa respecto al software que hace funcionar el Kindle. ¿La razón?, habían sido publicados por una empresa que no tenía los derechos para hacerlo en Estados Unidos<sup>14</sup>. Y por último, el reproductor de música de Windows, el Windows Media Player, cuyas nuevas versiones no permiten la reproducción de canciones que tengan dispositivo DRM<sup>15</sup>.

## Libertad 1

Para la libertad 1 que es la libertad de estudiar el funcionamiento del programa y adaptarlo a las necesidades, el acceso al código fuente es una condición previa, ya que para estudiar el funcionamiento del software es necesario tener un libre acceso a sus líneas de código, esto permitirá que posteriormente se modifique según las necesidades del usuario, si así lo requiere. Esta libertad incluye que se puede utilizar esa versión modifi-





Logo de Firefox, navegador de código abierto que actualmente es traducida a diferentes lenguas originarias mexicanas

El estudio del código fuente permite también su modificación, y a su vez, la adaptación de un programa a necesidades muy específicas que de otro modo no hubiesen sido solucionadas, por ejemplo, existen iniciativas para que el navegador libre Firefox sea traducido a lenguas originarias mexicanas gracias al apoyo de algunas asociaciones civiles y a programadores entusiastas mexicanos. Hasta agosto del 2010, se encontraba lista la versión en zapoteco, tarahumara y el maya en un 80%, y se está trabajando en traducirla al náhuatl<sup>16</sup>. Ejemplos similares ocurren en Bolivia y en Perú.<sup>17</sup>

La herramienta Spiro, integrada a las nuevas versiones de algunos programas libres, es otro ejemplo de las ventajas de poder estudiar y modificar un código abierto. En el 2009 Raph Laven, desarrolló una técnica para crear curvas interactivas en el diseño digital para su tesis de doctorado titulada "From Spiral to Spline: Optimal Techniques in Interactive Curve Design"<sup>18</sup>. Dichas técnicas, que son muy útiles para el diseño de tipografías, se integraron en Spiro, herramienta que estuvo disponible en las más recientes versiones de programas libres como Inkscape<sup>19</sup> y FontForge<sup>20</sup>. Gracias a la apertura del código de Inkscape, Laven pudo desarrollar una herramienta para este programa y pudo ser integrada después para el beneficio de todos los usuarios. Laven también liberó el código para que todos puedan ver cómo pro-

cada en lugar de la original. El software deja de ser software libre en tanto el código no sea libre y no pueda ser modificado, y más aún, que no se puedan utilizar dichas modificaciones.

La web es un ejemplo amplio y basto de las aplicaciones que tiene esta libertad, ya que al ser el HTML y el CSS lenguajes descriptivos definidos de manera estándar y abierta, y en lo que muchos sitios están desarrollados, estos pueden ser estudiados para conocer su comportamiento. Es posible ver el código con el que fue hecha una página programada en HTML/CSS y conocer cómo fue realizada, lo cual propicia mejoras futuras. Gran parte del crecimiento tan exponencial que ha tenido la web se debe a que está sustentada bajo un esquema de código abierto.

gramó Spiro<sup>21</sup>. Century Catalogue, Inconsolata, Museum Caps, LeBe, ATF Bodony ATF Frankling Gothic<sup>22</sup>, son fuentes tipográficas diseñadas a partir de Spiro.

## Libertad 2

La libertad 2 que es la de redistribuir copias, debe incluir el código fuente; tanto para las versiones modificadas como para las no lo están.

Esta libertad tiene implícito el “derecho” de distribuir copias de software libre a manera de compartir una herramienta, ya que si estamos en el entendido que el software es una herramienta que ayuda a solucionar problemas, en el caso que nos confiere, una herramienta de creación, ¿debería de impedirse que compartamos dicho software con el fin de ayudar al prójimo?. Prestar o distribuir copias de software libre es una manera de ayudar a la gente a solucionar problemas y que dicha actividad sea completamente legal.

Ejemplos de esto lo mencionamos ya en el capítulo I, donde explicamos la dificultad gracias al alto costo, de conseguir una copia con licencia de software privativo. El S.L. representa una herramienta que puede compartirse sin ningún problema legal.



Diferencia entre un disco de instalación de Windows y uno de la distribución libre, Ubuntu. El primero dice, en inglés: Legalmente libre de copiar, modificar y redistribuir. el segundo dice: No prestar o hacer copias ilegales de este software.

## Libertad 3

### **Libertad 3: distribuir las copias de versiones modificadas a terceros**

Para la libertad 3, la de distribuir copias de sus versiones modificadas a terceros, el acceso al código fuente también es una condición necesaria, si se distribuye, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios.

Esta libertad tiene implícita la idea de colaboración, ya que al compartir los cambios o mejoras ayuda a que todos los usuarios de dicho software avancemos de una manera más eficaz y rápida. Es la metáfora de subirse en hombros de gigantes, ayudarse de lo que ha hecho el prójimo con sus conocimientos y utilizarlo al beneficio propio y a su vez<sup>23</sup>, ser parte de esta comunidad que proporciona y se beneficia de los avances que entre todos, se realizan. Todo esto mediante un acuerdos de partes, es decir, licencias que protegen la distribución del S.L., una de ellas es el GPL, basado en la idea del copyleft, de la cual hablaremos a profundidad más adelante.

Un caso de violación a la GPL que ha sido llevadas a juicio fue en el 2007, donde la compañía vendedora de software Monsoon Multimedia, violó la licencia GPL cuando ofreció el programa libreBusyBox sin permitir el acceso al código del programa a quien lo adquiría. El resultado, fue que Monsoon Multimedia acordó nombrar a un oficial de cumplimiento de código dentro de sus organización para controlar y asegurar el cumplimiento de la GPL, publicó el código fuente para la versión de BusyBox que estaba distribuyendo y notificó a los beneficiarios anteriores de BusyBox sus derechos sobre dicho software. El acuerdo también incluyó una compensación económica a los demandantes<sup>24</sup>.

Así pues, la ideología del software libre plantea que las tecnologías digitales de la información ayudan en muchos sentidos a que ciertas actividades resulten más cómodas y sencillas, y sobre todo que “ayudan al mundo haciendo que sea más fácil copiar y modificar información”<sup>25</sup>, pero que no a todos le resulta conveniente que esto sea tan fácil, como a los grandes corporativos que se dedican a la venta de licencias de software.

Según estas ideas, los programas de computo no debe de tener un “dueño” que se reserve los derechos, y eso no significará que no sea “autor” del mismo<sup>26</sup>. Se basa en la premisa de que el “conocimiento es de todos”, pues al compartir dicho conocimiento ocurre un beneficio, no solo particular, sino a nivel comunitario. Al mejorar un software con la ayuda de una

El abogado estadounidense, Lawrence Lessig, se inspira en la filosofía del software libre y acuña recientemente el concepto de Cultura Libre, la cual define la visión de una cultura basada en la libertad de distribuir y modificar trabajos y obras creativas: "Las culturas libres son culturas que dejan una gran parte abierta a los demás para que se basen en ella, las que no son libres, las culturas del permiso, dejan mucho menos."<sup>27</sup>

Se refiere a basarse en lo que han hecho otros, en lo que se ha hecho en el pasado, en una cultura basada en la compartición y distribución libre del conocimiento.

El conocimiento colaborativo es otro de los pilares del S.L. esto es que entre mas personas colaboren para la creación de un software, se creará un mejor programa a una velocidad mucho mayor y los usuarios se verán beneficiados con ello y parte del principio que "la compartición con los demás constituye la base de la sociedad"<sup>28</sup>

Dentro de la comunidad de usuarios del S.L. se hacen públicas las mejoras que se realicen a algunos programas, esto es accesible para quien este interesado en usarlo e implementarlo y permite que un solo usuario utilice un programa combinado con otro según sus necesidades. También permite que el propio usuario desarrolle su propia herramienta que solucione sus problemas específicos de producción.

Resumiendo las ideas planteadas hasta ahora, el S.L. se basa en la cultura libre, conocimiento colaborativo, compartición de conocimiento, libertad en el conocimiento, copyleft y sobre todo, libertad.



Cultura libre, cómo los grandes medios usan la tecnología y la ley para cerrar la cultura y controlar la creatividad.  
Por el abogado estadounidense Lawrence Lessig

**Una “patente” trata de proteger “ideas” y al protegerlas, imposibilita que los demás las usen y por lo tanto que se generen otras ideas a partir de ellas**

2.14. Software libre y software privativo: licencias que “permiten”, licencias que “privan”

Una vez que hemos hablado en el capítulo I del software patentado y en el inicio de éste, del software libre, es pertinente dedicar un espacio a hablar de las licencias que diferencian a ambos tipos de software.

Por un lado, el software de patente, basa sus licencias en el concepto de copyright, es decir, que sus autores se reservan todos los derechos, entre ellos, el código fuente con el que fue creado, impidiendo que sea visible y por lo tanto que otras personas lo usen, distribuyan, copien o modifiquen de manera legal, sin el permiso por escrito del titular de los derechos. El código se encuentra protegido por una licencia, que es necesario pagar si es que quiere utilizarse de manera legal. Sin embargo, esta licencia sólo da el permiso de “usarlo” e incluye algunas restricciones, como instalarlo en más de una computadora, por ejemplo; el código fuente pertenece a una empresa, compañía, fundación o persona y sus autores se reservan todos los derechos.

Sin embargo, no sólo el término de Copyright está relacionada con este tipo de software, también lo está el término “patente” y es aquí donde comienza a ser engañoso.

Una “patente” trata de proteger “ideas” y al protegerlas, imposibilita que los demás las usen y por lo tanto que se generen otras ideas a partir de ellas: “Toda patente protege alguna idea. Las patentes de software son patentes que protegen ideas que tienen que ver con el software, ideas que podrían usarse para desarrollar software”<sup>29</sup> Además, son pocos los países que aceptan patentar ideas computacionales (como es el caso del software.)

El problema de “patentar” el software valiéndose del copyright, es que los conceptos se confunden a menudo y engañan al usuario diciendo que se están “protegiendo” los derechos de los autores (cuando en realidad protegen el derecho de las empresas, pues en la práctica, los autores seden sus derechos a los editores y/o compañías para las que trabajan). Una diferencia entre el copyright y las patentes, es que la primera regula las condiciones de expresión de una obra, no protege ninguna idea, mientras que la segunda sólo protegen las ideas y el uso de las mismas. Además, el copyright se aplica automáticamente. Las patentes son publicadas por una oficina de patentes como respuesta a una solicitud y son mono-

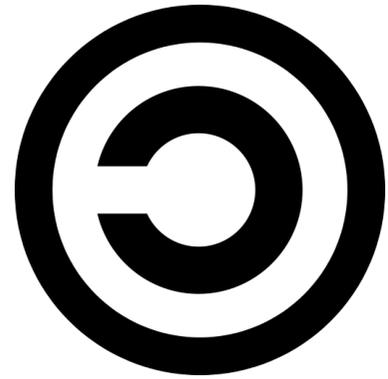
polios absolutos sobre el uso de una idea. Incluso si pudiéramos probar que la idea es nuestra, sería completamente irrelevante si la idea ha sido patentada por otro<sup>30</sup>.

Ejemplos de software patentadoson, en sistemas operativos, los comúnmente usados: Windows en todas sus distribuciones y OSX para MAC, en programas de oficina, Word, Excel, Power Point y todos los pertenecientes a la distribución de Microsoft Office. En programas para edición y creación de imágenes, movimiento y animación que se han convertido en las herramientas de diseñadores y artistas visuales, tenemos a Photoshop, Illustrator, Flash, 3D Max, Maya, Dreamweaver, Indesign, Premier, Pro tools, After Efects, entre otros.

Hace algunos años, entre la comunidad de software libre se usaba el término "propietario" para referirse a este software de patente, sin embargo, Enrique Chaparro, un activista argentino de software libre propuso en el 2003, el uso del término "**privativo**" dado que es un adjetivo que expresa más claramente la noción de que este tipo de software "priva" ciertas libertades de los usuarios. Es por esta razón que este tipo de software, que mantiene su código fuente no abierto, que no permite ser copiado ni reproducido, mucho menos distribuido sin autorización de sus creadores, será llamado a partir de ahora **software privativo**<sup>31</sup>.

Por otro lado, la intención del movimiento del software libre de Stallman es producir código en la medida en que pueda ser transparente y susceptible de modificación haciéndolo libre. El mecanismo para que esto sea posible es un instrumento llamado copyleft que se implementa a través de una licencia llamada GPL (General Public License). Tal como lo menciona Lessig en la introducción de **Software libre para una comunidad libre**: "Usando el poder del copyright, el software libre no sólo asegura que permanecerá abierto y susceptible de modificación, sino también que otro software que incorpore y use software libre y que técnicamente se convierta en obra derivada, debe también, a su vez, ser libre" <sup>32</sup>

El **copyleft** determina que un software (o cualquier otra creación) se distribuya respetando, a su vez, sus características de libre distribución. Esta licencia establece que al redistribuir un programa no puedan añadirse restricciones que falten



Logo de Copyleft

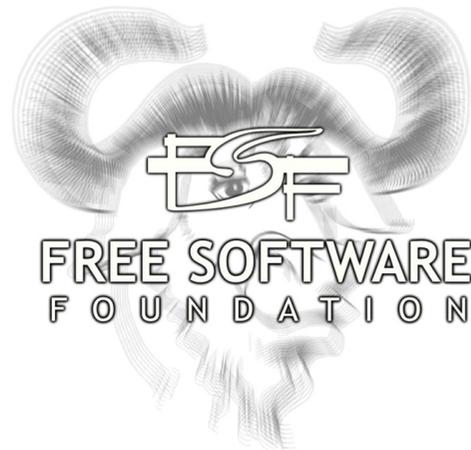
con algunas de las libertades anteriormente mencionadas, de tal forma que estas libertades quedan protegidas: “si uno usa y adapta un programa de software libre y distribuye públicamente esa versión adaptada, la versión distribuida debe ser tan libre como la versión de la que procede. Debe hacerse así, de lo contrario se estará infringiendo el copyright”<sup>33</sup>. Es darle la vuelta al concepto del copyright y en “lugar de privatizar el software ayuda a preservarlo como software libre”.

Así pues, la licencia GNU-GPL protege los derechos por medio de la combinación de dos medidas: 1.- poner el software bajo copyright y 2.- la propia licencia que otorga permiso legal para copiar, distribuir y/o modificar el software.

El preámbulo de esta licencia dice:

“Las licencias que cubren la mayor parte del software están diseñadas para despojarle de la libertad para compartirlo y para modificarlo. Por el contrario, la Licencia Pública General de GNU pretende garantizar la libertad de compartir y modificar software libre para asegurar que el software sea libre para todos sus usuarios. Esta Licencia Pública General se aplica a la mayor parte del software de la Free Software Foundation y a cualquier otro programa, si sus autores se comprometen a utilizarla”<sup>34</sup>

Una de las posturas de este trabajo es la identificación de 2 corrientes en el uso y realización del software. Por un lado el software privativo que “protege” los derechos de las empresas y realizadores y por otro el software libre que “protege” los derechos de los usuarios.



## 2.2. Software Libre en el terreno de la creación visual digital

Una vez que hemos visto de manera detallada las principales diferencias entre los dos tipos de software, ha llegado el momento de hablar de las ventajas y las problemáticas de la adopción del software libre. Se hablará de algunas razones por las cuales podría ser necesario considerar el uso del software libre para crear visualmente por medio de un equipo de computo, sin que eso represente necesariamente una comparación con el software privativo.

### 2.2.1. Caracterización, problemáticas y diferencias con el software privativo

#### A) EL ASPECTO TÉCNICO

**El S.L. puede modificarse, lo cual lo hace flexible, permite integrar soluciones:** El acceso al código fuente permitiría al artista y/o diseñador realizar las modificaciones al software que crea convenientes y aunque no se tengan conocimientos de programación, se puede recurrir a la interdisciplina y con la ayuda de un programador, modificar el código y obtener funciones nuevas. Esto es mucho más eficiente en el caso del arte electrónico, por ejemplo, en el que las funciones que debe realizar una obra son muy específicas y es mucho más complicado (y caro) llevarlo a cabo con software privativo. Por ejemplo, en el mes de abril del 2010, el Festival de México FMX presentó en el teatro Julio Castillo la Ópera titulada “Únicamente la verdad, la verdadera historia de Camelia la Texana”, en donde, una de las particularidades de la puesta en escena, consistía en controlar en tiempo real un video a 4 pantallas. El problema fue resuelto con “unas tarjetas de video (modelos viejos), y unos scripts corriendo en Linux, (que) sustituyeron los más de \$100,000 pesos que habría costado la renta de las licencias y computadoras de cualquier programa privativo para resolver el problema.”<sup>35</sup>

Esto se debe a que el carácter modular del S.L. permite la implementación de soluciones, permitiendo “ensamblar” programas libres entre si; tal como lo menciona una de las filosofías del sistema operativo **UNIX** del cual hablamos en el capítulo anterior, “escribir programas que trabajen juntos”. La nobleza de poder controlar, modificar y es-

tudiar un código abierto, permite que puedan ser usados scripts que alguien más ya escribió (o escribirlos uno mismo) y mejorarlo para satisfacer las necesidades específicas de cada usuario.

Otro ejemplo práctico de esto es el programa libre de modelado en 3D, Blender. Con Blender se han desarrollado producciones animadas de muy alta calidad, como Big Buck Bunny, Elephants Dreams y la más reciente Sintel. Con cada una de estas producciones se han implementado nuevas herramientas dentro del propio programa según las necesidades del equipo de producción. Para Sintel (2010, Amsterdam) por ejemplo, se llevaron a cabo modificaciones, entre otros, en la interfaz de usuario, en el sistema de partículas (lluvia, polvo), en el esculpido, el sombreado, el sistema de renderización y en la simulación de humo<sup>36</sup>. "Cómo decía Roosendaal de la Blender Foundation, Blender no se trata de un software, se trata de artistas interconectados comunicándose y generando cosas juntos, se trata de comunidad, no de programas"<sup>37</sup>

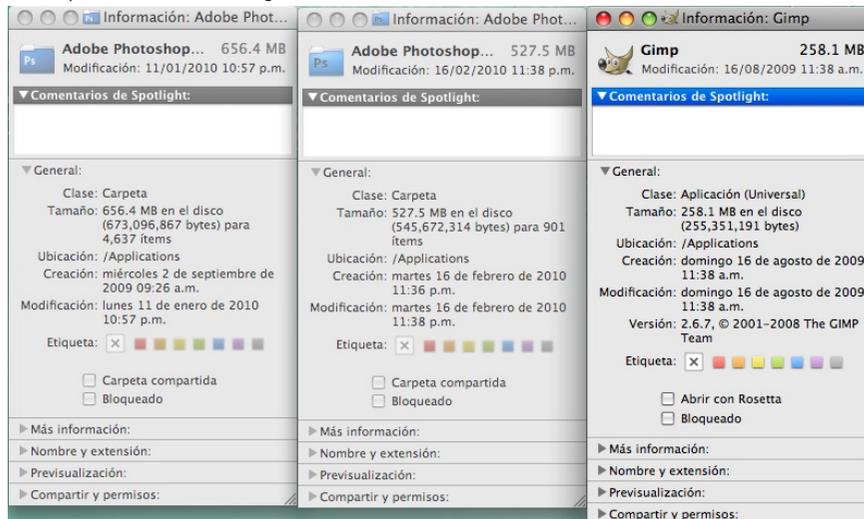
Otro ejemplo más de cómo esto puede llevar a desarrollos importantes de soluciones artísticas, es el caso de PURE DATA, un programa para la manipulación de datos y medios para los cuales, varios desarrolladores escriben el código necesario para resolver diversas necesidades. Cada línea de código se convierte en una línea que hace crecer la librería del código. Recientemente, Hans Christoph Steiner en colaboración con Golan Levin, desarrollaron una fórmula matemática implementada en curvas para animación. Dichas curvas fueron necesarias para la animación de un aliento elegante y bello. Si PURE DATA tuviera un código cerrado, hubiese sido imposible implementar estas curvas y el trabajo artístico probablemente hubiese sido detenido para negociar con alguna compañía de software para que incluyera la solución de dicha necesidad, sin mencionar el costo que esto implicaría. Por el contrario, la implementación que fue posible gracias al software libre, hace que estas herramientas estén disponibles para la comunidad, permitiendo a todos trabajar para alcanzar un nuevo nivel.<sup>38</sup>

**Acceso al proceso de las obras.** El código abierto permite también que sea más fácil saber cómo se realizó una obra determinada, conocer los plug ins, herramientas y cómo se llevó a cabo el proceso de producción digital. Esto es importante al momento de la conservación y difusión de la obra en un futuro donde no sabemos hacia donde y cómo avance la tecnología, con el uso del S.L. no será necesario pedir permiso a alguna compañía para que nos permita el acceso al código de nuestra obra y que futuras generaciones la analicen si es necesario.

Un ejemplo de esto es el programa libre Alchemy, disponible en: <http://al.chemy.org>. Alchemy permite realizar trazos digitales como en cualquier programa de trazo libre, (como paint, por ejemplo), con la diferencia de que dichos trazos pueden ser vectorizados y vueltos a ver en la secuencia en la que fueron creados, pues existe una salida para archivos con formato estándar SVG y editado posteriormente en un archivo de edición de vectores, además de que es posible grabar los movimientos que realizamos en dichos trazos y encontrar patrones. El archivo generado en la

obra puede ser abierto en un editor de texto y poder ver, en líneas de código, la característica de cada uno de nuestros trazos, como las coordenadas de los polígonos, el grosor del trazo, el color, el filtro utilizado, etc. Esto podría permitir estudiar una nueva generación de arte digital.

**El S.L. es multiplataforma y no requiere de excesivas especificaciones de hardware.** Imaginemos que tenemos la oportunidad de realizar un curso de ilustración digital en una comunidad de difícil acceso y sólo se nos ha proporcionado algunos equipos de muy baja capacidad para dichos fines. En una PC pentium 2, por ejemplo, sería muy complicado instalar Illustrator CS5 debido a que las especificaciones de dicho software requiere de una capacidad mayor a 1G de memoria RAM y un Procesador Intel® Pentium® 4 o AMD Athlon® de 64 bits<sup>39</sup>. En el caso de Inkscape por ejemplo, el homólogo de Illustrator en S.L., es un programa muy ligero que no requiere más de 512k de RAM. Una de las razones por las que los programas libres son más ligeros se debe a que comparten bibliotecas entre sí, Inkscape y Gimp (programa libre que realiza funciones similares a Photoshop) comparten GTK<sup>40</sup> que es una biblioteca que se encarga de desarrollar las interfaces gráficas de usuario, como los botones del menú. En lugar de que cada programa tenga su propia biblioteca y sus propios botones y con ello, ocupe más espacio en el disco y en la memoria RAM, Inkscape y Gimp la comparten entre sí para hacerse más ligeros.



Captura de pantalla comparando el peso de Photoshop y de Gimp.

Además de esto, es común que algunos S.P. como Photoshop especifiquen que las tarjetas de video, el RAM o incluso la resolución de las pantallas no son suficientes para algunas tareas. Eso no sucede en el caso del S.L. que en general precisa de pocos requerimientos de hardware para su instalación.

Para usar programas libres no se requiere necesariamente que corran en el sistema operativo GNU/LINUX, algunos programas para creación gráfica digital como Inkscape, Blender, Gimp, My paint, entre otros, corren en el sistema operativo Windows y en OSX.

**Uso de estándares:** Hablamos en el capítulo I de la importancia del uso de estándares y como éstos permiten que una obra digital pueda permanecer en el tiempo. Los programas libres, así como el movimiento del S.L. promueven el uso de estándares en la generación de archivos realizados con esta herramienta, esto es con el fin de que nuestras creaciones, información y contenido que generemos pueda ser visto por todas las computadoras independientemente del sistema operativo que las haga funcionar, es decir, no importando la compañía que nos venda el software, de sus características, deficiencias o virtudes. Además, cuando hablamos de estándar nos referimos a que este sea [41](#):

- 1.- Sujeto a evaluación pública y uso sin restricciones de una manera igual en todas las partes.
- 2.- Carezca de ningún tipo de componentes o extensiones que tengan dependencias sobre formatos o protocolos que no cumplen la definición de un estándar abierto en sí mismos.
- 3.- Libre de cláusulas jurídicas o técnicas que limiten su utilización por cualquiera de las partes o en cualquier modelo de negocio.
- 4.- Gestionado y desarrollado independientemente de cualquier proveedor en un proceso abierto a la igualdad de participación de los competidores y terceros.
- 5.- Disponible en múltiples implementaciones completas compitiendo por proveedores que compitan entre sí, o como una implementación completa igualmente a disposición de todas las partes.

El uso de estándares nos permitiría a la larga elegir cualquier sistema operativo o aplicación y que podamos abrir y editar nuestros archivos antiguos, por ejemplo, los realizados en algu-

na versión anterior del software sin que eso represente cambios en dicho archivo y por supuesto, permitiría utilizar cualquier software para generar documentos y que cualquier persona pueda abrirlos.

Lila Pagola, comenta su experiencia al respecto: "Procuro guardar en formato estándar por que es donde tengo menos sorpresas entre lo que yo quiero hacer y lo que después sucede con el archivo cuando lo abro tiempo después o en otro programa"

**Existe menos incidencia de virus:** Hay tantas distribuciones de GNU/LINUX que sería en principio muy complicado que el virus funcionara en todas, y como estas distribuciones de igual forma están en constante actualización, la ejecución de un virus se complica más. Además de que en GNU/LINUX cada usuario tiene ciertos privilegios, si existe un virus, sólo puede estar en la carpeta de dicho usuario y tener sus privilegios, no puede entrar a elementos del sistema que lo haga desvariar, es decir, para poder acceder al interior del sistema es necesario ser un usuario [root<sup>42</sup>](#). Esta cuenta tiene todos los permisos para hacer cualquier cosa en el sistema, incluso, se aconseja no usarla a menos de que se trate de un usuario experimentado o tenga conocimiento del sistema operativo. Para acceder a este control del sistema se requieren ciertos privilegios de superusuario a través de la instrucción [sudo<sup>43</sup>](#), un comando que se teclea en la terminal<sup>44</sup> del sistema operativo seguida de nuestra contraseña. Esto impide que cualquier persona ajena a nosotros tenga control del sistema o programas maliciosos se ejecuten sin nuestro permiso, entre ellos, los llamados virus informáticos. Además de que GNU/LINUX, basado en el sistema operativo UNIX, está hecho de forma "modular". Por otro lado, es importante decir que nos referimos al sistema operativo libre GNU / LINUX, en el caso de programas libres que corran sobre un sistema operativo privativo, es decir de WINDOWS y/o OSX, estos quedarán supeditados a las características y seguridades de dichos sistemas.

**Estable y seguro:** El punto anterior tiene mucho que ver con esta ventaja, ya que el hecho de que no haya virus peligrosos para GNU/LINUX habla de la seguridad que ofrece, no sólo por que se necesitan privilegios para entrar a modificar el sistema, sino también por que este sistema operativo cuenta con aproximadamente 5000 desarrolladores de más de 200 compañías que respaldan sus líneas de código<sup>45</sup>, además de que, miles de ojos de programadores de todo el mundo están puestos en las 2.7 millones de líneas de código abierto del kernel de linux y en los errores que éste pudiera tener, en mejorar el rendimiento y reforzar la

**Para adquirir una licencia de la Adobe Creative Suite Design, un diseñador tendría que invertir unos 384 salarios mínimos en México**

seguridad, haciendo que las mejoras sean realizadas en un periodo de tiempo mucho más corto y los problemas detectados con mayor prontitud. Una nueva versión de kernel es liberada al cabo de 8 a 12 semanas, y está lista para que los usuarios la descarguen e implementen. En estados Unidos, el departamento de defensa, la flota de submarinos de la fuerza naval, la administración federal del aviación, la corte federal, entre otros, así como el gobierno de la ciudad de Munich, el parlamento francés y el banco industrial y de comercio estatal de China, han migrado a sistemas operativos libres, entre otras razones por la seguridad que ofrece<sup>46</sup>.

## B) ASPECTOS ÉTICOS Y SOCIALES:

**Bajo costo.** Utilizar software libre puede representar un muy bajo costo de producción al momento de buscar programas que nos ayuden a generar imágenes. En el caso de las Organizaciones No Gubernamentales (ONG) o Asociaciones Civiles (A.C.) en las que no siempre se cuenta con un presupuesto dedicado a la inversión de licencias de software y que requieran generar material gráfico y de difusión para su promoción, el S.L. puede ser una buena opción sin necesidad de recurrir a la copia no autorizada del software, de igual forma que puede serlo en la industria cultural. Cabe recordar que el costo del software libre se ve reflejado en la eventual capacitación para su uso, el cobro de honorarios de las personas que realicen la migración y en el soporte técnico del mismo.

Como mencionamos ya en el capítulo anterior, la Adobe Creative Suite Design Premium tiene un costo de U\$1 899<sup>47</sup> dólares, eso es poco más de \$23 000 pesos mexicanos, lo cual indica que un diseñador tendría que invertir unos 384 salarios mínimos en México para adquirir la licencia. Además que este costo no sería la única inversión, pues las licencias deben ser renovadas cada que se lleve a cabo una actualización, esto es, aproximadamente cada año. Con el S.L. tampoco es necesario adquirir una nueva versión, existen actualizaciones que pueden realizarse conectándose a la red, sin costo y completamente legales.

**Legal:** Como ya dijimos en este capítulo, todos los programas libres ofrecen una licencia denominada GPL<sup>48</sup> GNU, que en sus siglas en ingles significa Licencia General Pública de GNU<sup>49</sup>. Dicha licencia pretende asegurar que el software no represente una limitante para quien lo use, para ello, debe de respetarse las 4 libertades del S.L. que también ya mencio-

namos. Este contrato creado por la Free Software Foundation, hace que el S.L también contenga una licencia pero que ésta, más que restringir su uso, copia en otras computadoras, distribución, cambio o mejora, permite que esas libertades se respeten en todas y cada una de

las versiones que se realicen de dicho software, esto es, como ya dijimos, basado en el copyleft. El copyleft básicamente surgió de la necesidad de que el S.L no se convirtiera a lo largo el software privativo, así pues:

“La idea fundamental del copyleft es que se autoriza la ejecución del programa, su copia, modificación y distribución de versiones modificadas, siempre que no se añada ninguna clase de restricción a *posteriori*. De

este modo, las libertades cruciales que definen el software libre quedan garantizadas para cualquiera que posea una copia; estas libertades se convierten en derechos inalienables”.<sup>50</sup>

Esto ofrece una ventaja importante en el terreno de la creación pues al usar software libre podemos estar seguros de que nuestra herramienta de trabajo está dentro de los marcos legales, lo cual es muy útil al momento de hacer una distribución digital masiva, por ejemplo.

**Fácil de conseguir:** Para poder acceder a un programa de edición de imágenes libre es necesario sólo una co-

nexión a internet, bajarlo, instalarlo y comenzar a utilizarlo, de igual forma el disco de la distribución de Ubuntu que por lo general se distribuye gratuitamente, contiene no sólo el sistema operativo GNU/LINUX, sino la paquetería de oficina y de gráficos, lo que nos

**“La idea fundamental del copyleft es que se autoriza la ejecución del programa, su copia, modificación y distribución de versiones modificadas, siempre que no se añada ninguna clase de restricción a posteriori. De este modo, las libertades cruciales que definen el S.L. quedan garantizadas para cualquiera que posea una copia”**

permitiría, de tener dicho disco, operar en cualquier lugar que nos ofrezca un equipo de computo para trabajar. Existen además distribuciones de GNU/LINUX portables, es decir que caben en un usb de poco mas de 1 giga y que pueden correr en cualquier computadora si le indicamos que cargue el sistema operativo desde dicho usb, (hacer un booteo) sería como tener una navaja suiza o nuestra propia distribución portátil para

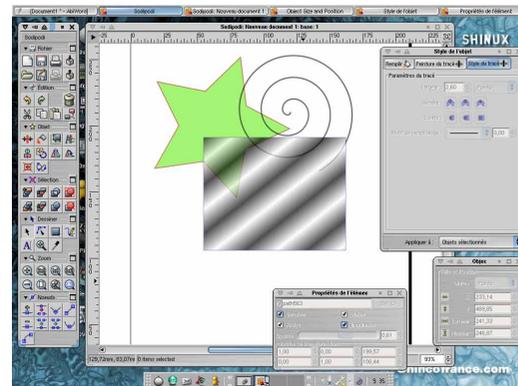
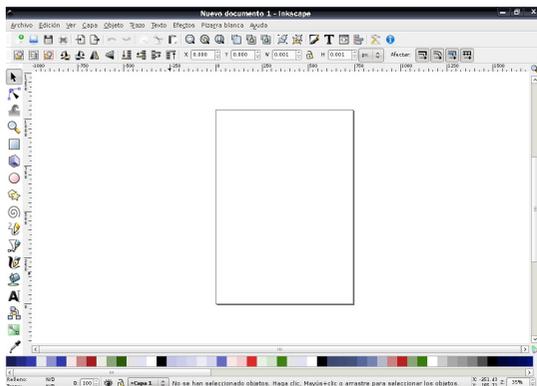
usar en cualquier equipo, sin necesidad de recurrir a las copias ilegales con algún distribuidor no autorizado lejos de nuestro lugar de trabajo o la compra de una licencia que exceda nuestras posibilidades económicas.

Para ejemplificar esto, hablaré de un caso anecdótico. Durante casi toda mi carrera universitaria viajé constantemente a Tulancingo Hidalgo donde viven mis padres, más de una ocasión necesité hacerle modificaciones a un archivo realizado en Photoshop o Illustrator, cosa que me fue imposible debido a que no contaba con equipo portátil en aquel entonces y en

aquella región de Hidalgo las computadoras de los café Internet apenas y tenían Microsoft Office instalado. Lograr comprar una copia no autorizada del software en aquella pequeña ciudad, era también imposible, pues el uso de estos programas aún no era generalizado en aquella región; pedirlo prestado menos, por que ni siquiera los diseñadores tulancinguenses utilizaban Illustrator, sino Corel en su versión 10. En aquella época, mas de una ocasión fue complicada la tarea de modificar mis archivos a distancia. El software libre, eventualmente solucionaría éstos y otros problemas de accesibilidad a una herramienta de trabajo.

**Se trata con personas que desarrollan y no con compañías que crean un producto, no fomenta el monopolio:** El software libre es de todos, no le pertenece a ninguna compañía o individuo, es posible que cualquier persona con conocimientos de programación pueda llegar al fondo de algún problema y resolverlo sin tener que recurrir a las soluciones de algún proveedor del programa o al llamado “soporte técnico” que lo provee una sola empresa, pues se esta tratando con “personas” que desarrollan el software. La característica de que sea libre y abierto y por lo tanto, accesible a todos, podría darle un giro humanístico a la tecnología, pues se trata con personas (aunque también pueden ser corporaciones y/u organizaciones) que lo desarrollan o implementan soluciones y no con compañías que venden productos. Esto nos hace incluso dejar de contribuir al crecimiento de los monopolios, acercarnos a lo que podríamos llamar “consumo responsable” en el rubro de herramientas tecnológicas, ya que contrario al S.P. no se beneficia a una compañía o a un monopolio en particular, sino a una comunidad. La posibilidad de generar un software que puede ser modificado por todos da la oportunidad de que se generen los llamados FORKS<sup>51</sup> que son los rumbos distintos que se generan de un programa que ya no se sigue desarrollando, o bien, que no cumple con las expectativas de una comunidad. Por ejemplo, el caso de Inkscape es un fork de un programa anterior llamado SODIPO-

A la izquierda, captura de pantalla del programa vectorial libre Inkscape que es n “fork” de Sodipodi, a la derecha.



DI que tomó una bifurcación en manos de una comunidad con necesidades específicas de diseño.

Respecto a esto, Christian Oyarzún comenta:

“(el software libre) permite también bajar una figura de autoridad que pasa con las tecnologías propietarias, es decir, “la marca”. Desaparece la marca y aparecen otros colectivos o investigadores que desarrollan problemas específicos y comienzas a ver muchas formas de resolver problemas o problemas que nunca antes habías visto”<sup>52</sup>

En este mismo sentido, cuando se usa S.L., invariablemente el rol de usuario-cliente que se tiene con el S.P. se transforma en un rol mucho más amplio y participativo, alejado de la pasividad, pues es posible conformar la comunidad que es parte del proceso del desarrollo del software, entendiendo de cerca la relación entre código, soluciones, herramientas y medios, software y resultados, tal como se menciona en FLOSS + ART:

“La diferencia principal entre usuarios agrupados como clientes y usuarios que forman parte de una comunidad de S.L., es que estos últimos tienen la capacidad de ser pro-activos en el proceso de desarrollo de software a través de la escritura de documentación, apoyo, información y en algunos casos participación en la escritura del software.”<sup>53</sup>

En entrevista, Guillermo Espertino, diseñador argentino dice:

“A mi me parece bastante importante el tema de la libertad y de poder involucrarse con el desarrollo. Tener un error en el programa y poder hablar personalmente con quien puede arreglarlo, eso me parece insuperable y nunca habrá esa posibilidad en el software privativo”<sup>54</sup>

**Todos para uno y uno para todos:** Por último y no menos importante, al usar S.L. desde la perspectiva de un diseñador o un artista visual, contribuiría a que las siguientes versiones de programas se mejoren de acuerdo a nuestras necesidades o a la deficiencia de las anteriores versiones, algo que nadie más que un usuario especializado en imagen podría realizar; estaríamos contribuyendo a que se realicen cambios que beneficien a todos y cada uno de los usuarios de S.L. en el mundo, sin ninguna excepción o limitación. Como lo mencioné en el rubro anterior, se beneficia a una comunidad y es donde entra el dicho de que si uno avanza, los demás también lo hacen.

Guillermo Espertino, diseñador argentino usuario de software libre, en entrevista. Rosario Argentina, mayo 2011



Al respecto, Guillermo Espertino comenta que algunos de sus diseños fueron incluidos en las recientes versiones de Inkscape, más específicamente, los gráficos de los punteros del mouse al cambiar de herramienta: “Como diseñador aparte tienes la posibilidad de aportar tus diseños a un programa que tiene distribución mundial y son tuyos”. Existen también repositorios en línea de plug ins para Gimp, el programa libre de manipulación de mapa de bits, creados por la comunidad de usuarios y no por una instancia en particular o una empresa privada: <http://registry.Gimp.org> [A].

Aunque también es cierto que el S.L. cuenta con una retroalimentación por parte de los usuarios, dichos “feedbacks” difícilmente se reflejarían en cambios al programa que a la propia compañía no le convengan o le representen algún tipo de pérdida, como hacerlos más interoperables con las versiones anteriores, o abrir archivos nativos de software libre o traducir el programa en alguna lengua minoritaria.

Sobre esto, Martín Eschoyes, artista digital y diseñador gráfico argentino, comenta:

“Con el software libre hay un feedback constante entre diseñadores y programadores, intercambio de ideas, construir sobre otras cosas que ya han construido otros, eso es un potencial que no tiene ningún otro software”<sup>55</sup>



Martín Eschoyez, diseñador y artista argentino, en entrevista. Córdoba Argentina, Junio 2011

La oportunidad de no lidiar con dueños que restrinjan los derechos del software como obra intelectual permite que la participación se canalice y se generen una cantidad más grande de programas derivados de otras iniciativas. Un ejemplo es el Libre Office, una bifurcación de Oppen Office, la opción de paquetería ofimática libre, o el Gimpshop, una bifurcación de Gimp que pretende tener más similitudes con el software privativo Photoshop. Todos los ejemplos de forks en el S.L. son una muestra de como una comunidad hace virar un proyecto en una dirección distinta a la de los programadores principales.

Lila Pagola comenta al respecto: "La comunidad del software libre me representa una influencia impactante. Sobre todo ha cambiado mucho mi modo de trabajar desde que la conozco. Una comunidad internacional organizada, unas formas de hacer posible esas colaboraciones. Me parece un modelo interesante y potente para replicar"<sup>56</sup>

Nuevamente, Martín Eschoyez da su opinión respecto a este tema:

"Me parece que fomenta la creatividad (el software libre) por que te sientes con la necesidad de compartir todas esas cosas y descubrir que eso es mucho más poderoso que quedar-te las cosas y no compartirlas con nadie, esta idea de que el diseñador y/o el artista es súper elitista e individualista; en la filosofía del S.L. no es así."<sup>57</sup>

## 2.2.2 Problemáticas de la adopción del software libre en la creación gráfica digital

### A) GENERALIDADES:

**Se necesita de asesoría al empezar:** Para explicar este punto hablaré de mi experiencia personal como diseñadora y mis nulos conocimientos con respecto a programación. Cuando decidí iniciarme en el S.L. pedí me fuera instalado una partición de Windows con GNU/LINUX, desde entonces he necesitado eventual asesoría para instalación de programas, fuentes y solución de problemas en cuanto a tratamiento de imágenes se refiere, y aunque muchas cosas he podido resolverlas sola con ayuda de los foros en línea y la experiencia adquirida a lo largo de un par de años, también es cierto que aún no conozco todas las posibilidades de usar la terminal de mi ya no tan nuevo sistema operativo, recurso utilizado mucho más que en otros sistemas y que permite resolver problemas de manera más rápida y eficaz.

Sin embargo, esto no es ajeno a otros sistemas operativos como Windows, ya que es muy probable que cuando nos iniciamos en su uso, hayamos pedido ayuda al menos para instalar y comenzar a utilizarlo o incluso, aún lo hagamos. Adaptarnos al programa y aprender a manipularlo también nos llevó tiempo, años y posiblemente, ayuda de terceros.

**Curva de aprendizaje:** Una vez que aprendemos a usar un sistema operativo y nos acostumbramos a él, es difícil al principio reconocer y acoplarnos a las características de otro, esto no pasa sólo al momento de migrar a GNU/LINUX, sino cuando alguien acostumbrado a usar MAC, pasa al sistema operativo de Microsoft e incluso es probable que le escuchemos decir más de una vez: "odio Windows".

En el caso del S.L., algunas personas opinan que la curva de aprendizaje es mayor, esto puede ser debido a muchas cosas, una de ellas que los programas libres, debido a la poca difusión que tienen en el "mercado", las usan mucho menos personas que el S.P. Incluso, hay personas que asocian al sistema operativo Windows con todo lo que tenga que ver con computadoras. Es el sistema que se usa en el gobierno mexicano y para el cual los empleados reciben capacitación y el que además eventualmente se enseña en las escuelas. Es por eso que cuando se pretende incursionar en cualquier otro sistema operativo o cualquier otro pro-

grama, es preciso desaprender lo anterior. El software privativo definitivamente es más popular y más usado en computadoras de escritorio que el libre y el cambio implica siempre un nuevo aprendizaje, un nuevo entendimiento de una manera diferente de resolver cosas, un nueva forma de manejo de usuarios, de archivos, de instalar programas, etc.

José Santorcuato, artista electrónico y académico menciona: "Uno de los impedimentos para que el artista se acerque al software libre es la curva de aprendizaje, sin embargo poder replicar su obra en cualquier situación y cualquier entorno, es una recompensa"<sup>58</sup>

Sin embargo, Fabricio Caiazza, artista plástico argentino que además da clases de iniciación artística con software libre a niños, comenta:

"El mayor conflicto lo tienen las personas que ya tienen una experiencia previa con la manipulación de software privativo, como Photoshop, encuentran que en Gimp las herramientas están ubicadas en otro sitio y se desconciertan. Esto no ocurre con los niños (a los que les enseña) que su primer acercamiento con un software para manipular imágenes es justamente el Gimp, ellos trabajan con naturalidad"<sup>59</sup>

**No lo conocen en imprentas ni casas de impresión digital:** A pesar de que en muchos países europeos como Alemania, Francia y España y algunos de América como Brasil y Argentina, cuya población está familiarizada con el software libre y su uso, en el caso de México no pasa igual. En el mapa mostrado a continuación, realizado por el Georgia Tech/Red Hat Open Source

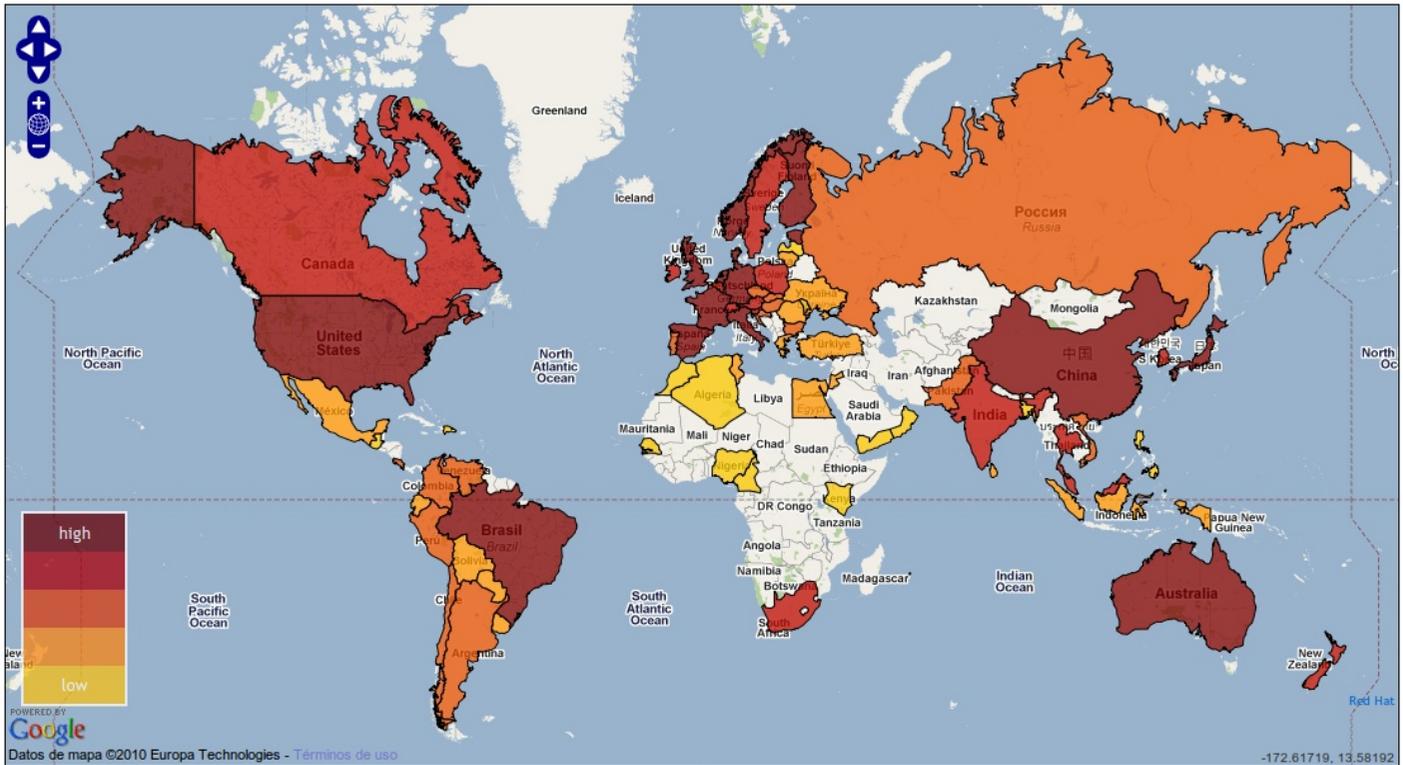
Software Potential Index project <sup>60</sup>se puede ver claramente la actividad relacionada con el uso del software libre en el mundo.

México mantiene una actividad aún baja con respecto a otros países, por lo que su popularidad es igualmente baja, más en ciertos sectores, como el diseño y el arte, lo cual hace suponer que somos todavía una minoría de diseñadores y artistas los que trabajamos con estas herramientas libres y por lo tanto, las casas de diseño y las imprentas no cuentan con estos programas a pesar de la facilidad de descarga y de su instalación, además de que les son desconocidos. Eso, aunado a que por confundirlo con software "gratis", la gente piensa que es de baja calidad o que es imposible desarrollar diseños de calidad con él.

Sin embargo, esto nos obliga a pensar nuevamente en la utilización de estándares al momento de realizar nuestros archivos. Mientras las casas de impresión requieren tener más de un programa privativo (en su mayoría en versión no autorizada) para abrir los archivos guardados en la extensión nativa de los diferentes programas existentes y en los que decida guardarlo el cliente, (generalmente, diseños pedidos en formato .AI, .PSD o .CDR), en su lugar podemos usar formatos PDF o SVG para que los abra cualquier programa.

**Instalación de Hardware:** En algunos casos, se requiere de más tiempo para instalar ciertos dispositivos de hardware, como tabletas de dibujo, cámaras, scanners o impresoras. Esto es por que algunos de estos dispositivos están fabricados para que funcio-

## Open Source Activity Map



Mapa de actividad del software de código abierto en 2010. Recuperado en Junio del 2011, en: <http://www.redhat.com/about/where-is-open-source/activity/>



nen en los sistemas operativos más comunes que utilizan controladores de código cerrado, lo cual complica el desarrollo para la compatibilidad con el S.L.. Sin embargo, siempre existe la manera de "parchar" estos problemas, sólo que requiere de más tiempo y paciencia por parte del usuario. Por ejemplo, un músico que recién compra un piano digital requiere que su tarjeta de audio que corre dentro de un sistema operativo libre, reconozca su nuevo piano. La compañía de fabricación del piano digital ha decidido no implementar controladores para GNU/LINUX pues no les representa una comunidad a la cual atender. Por tal motivo, el músico tendrá que visitar foros, pedir asesoría y hacer un esfuerzo extra para que su piano reconozca la tarjeta de sonido. Sin embargo esto resulta muy relativo, ya que, cuando yo compré la impresora en la cual imprimí esta tesis, me aseguré antes de comprarla, que fuera compatible con GNU/LINUX. Cuando la conecté a mi equipo, no tuve que instalar absolutamente nada, sólo la conecté por USB y comencé a imprimir, sin instalar discos, ni bajar controladores. Pero esto no siempre es así. Mucho de esto de debe a lo que algunos llaman *trusted computing* o computación de confianza, en el que supuestamente podremos confiar en las compañías que generan tecnología pues nos aseguramos que nos proveen de equipos confiables. Sin embargo en el video *Trusted Computing*<sup>61</sup>, sus autores se cuestionan si esto es realmente cierto, ya que, en teoría, una computación de confianza se basa en una reciprocidad, donde compañía y cliente confían entre sí. Los usuarios, deberían de ser capaces de decidir qué es bueno para ellos y sus necesidades y la compañía respetar estas ideas y protegerlos. Esto no ocurre así pues la compañía por lo general decide no confiar en el usuario y que no podrá resolver ciertas cosas por él mismo, o bien, decide no dar soporte cuando se trata de ciertos sistemas operativos. Entonces, si la compañía no confía en los usuarios, ¿los usuarios deberían confiar en la compañía que decide por ellos?

**El software libre posee muchas distribuciones las cuales representan variadas opciones según las necesidades de cada usuario.**

## B) PARTICULARIDADES:

**Muchas distribuciones que generan confusión:** El software libre posee muchas distribuciones<sup>62</sup> las cuales representan variadas opciones según las necesidades de cada usuario que además se actualizan constantemente, y aunque son adaptables para cualquier tipo de usuario y necesidad, esto para un usuario novato, puede causar cierta confusión al no saber que "distro" le conviene más. Para ello se necesita de la asesoría de algún usuario experimentado, de la búsqueda en Internet de experiencias de usuarios y/o desarrolladores o bien,

preguntar en los múltiples foros que se dedican al tema. Cada distribución tiene características muy específicas que las hace útiles para algunos casos, hay versiones mas estables que otras y otras más amigables. A un usuario novato se le recomienda el uso de UBUNTU, que hasta el 2011 liberó su versión 11.04 Beta, la cual tiene una interfaz gráfica de usuario bastante amigable. También existe UBUNTU STUDIO enfocado a diseñadores y artistas que trabajan con edición de sonido, video y gráficos 2D y 3D y Artix3D enfocada a la producción multimedia.

Aunque las múltiples opciones de distros del software libre son una ventaja en cuanto a satisfacer las necesidades a diversos números de usuarios, también es cierto que puede existir confusiones en usuarios que inician, así como puede resultar un inconveniente la solución de problemas específicas de cada distribución.

**Desarrollo de los programas:** Inkscape tiene aproximadamente 7 años de desarrollo, y ha logrado ser una herramienta poderosa en cuanto a edición de vectores se refiere, sin embargo, es evidente que no tiene el grado de desarrollo que posee la actual versión CS5 de Adobe. En muchas ocasiones, el poco desarrollo que tenga algún programa libre se debe a que no existe una comunidad que demande dicha mejora, esto es, entre menos diseñadores haya utilizando Inkscape, menos posibilidades existen de que se reconozcan los errores o problemas que represente para la creación, ya que sólo un especialista en esto, puede detectarlo en el uso cotidiano. Por otro lado, aunque no es postura de esta tesis "comparar" programas libres versus privativos, es cierto que al migrar de un software a otro, las comparaciones no siempre se pueden dejar de lado.

**En Gimp no existe aún una manera eficaz y rápida para deformar una imagen de tal forma que ésta se adecuó a una superficie ya deformada, el "WARP"**

En mi uso cotidiano del S.L. para el diseño, he encontrado que existen herramientas para las cuales resulta más complicado la resolución de algunos problemas, por ejemplo, en el caso de Gimp, no existe aún una manera eficaz y rápida para deformar una imagen de tal forma que ésta se adecuó a una superficie ya deformada, el llamado WARP en Photoshop. Para ello, se tiene que recurrir a deformarlo en perspectiva y luego el filtro de deformar por curva, pero resulta muy tardado pues no se tiene un control suficiente a dicha deformación.

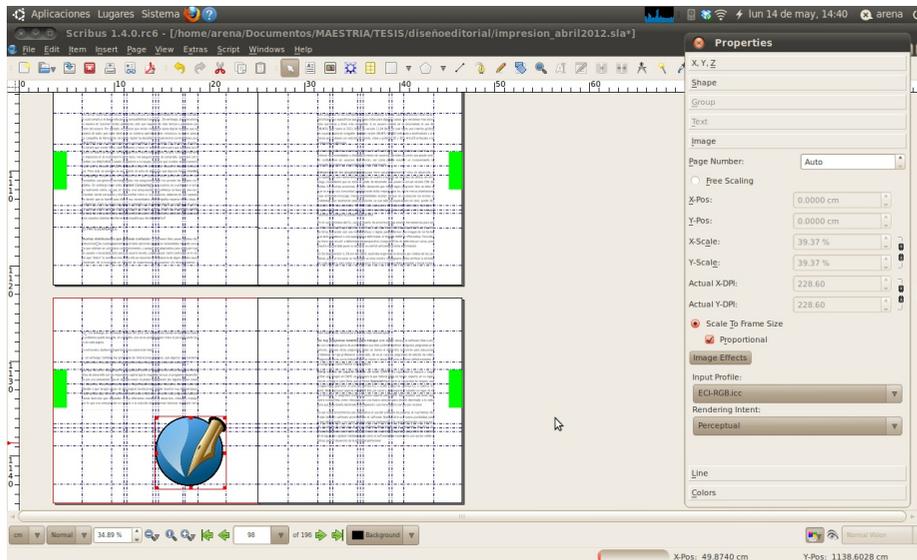
En Scribus<sup>63</sup> versión 1.3.6.svn del 2009, resultaba engorroso la división por sílabas de las palabras, pues al incrustar un texto que ya esta escrito, el programa pedía verificar la división de cada palabra, para saber si es correcta. Esto a la larga resultaba cansado y hasta moles-

to<sup>64</sup>. Sin embargo, en la versión Scribus NG 14.0, con la que está siendo armada esta tesis, el problema quedó resuelto, sin embargo, aún no es posible poner notas al pie en la parte baja de cada página.

En entrevista, Guillermo Espertino opina sobre este tema:

“...sin embargo, también hay problemas de limitaciones técnicas, que algunos son bastante importantes y hay que hacer un gran esfuerzo para superarlos y otros son menores.”<sup>65</sup>

Hay que tomar en cuenta que la comparación entre un programa comercial que lleva varios años de desarrollo con un importante capital que lo respalda, versus un programa desarrollado por una comunidad, que en algunos casos no posee financiación (en algunos otros casos si, donde las industrias invierten en un mejor desarrollo que las favorezca, como el caso de Blender o que tengan apoyo de sus propias fundaciones) puede resultar muy desventajosa. Cabe señalar que tanto el software libre como el software privativo son programas radicalmente distintos que responden a muy diferentes maneras de desarrollo, difusión y trabajo, por lo que una comparación en cuanto a la solución de problemas técnicos resultaría también insuficiente, inconclusa y una vez más, desventajosa.



Captura de pantalla de Scribus NG 14.0

**No hay programas estables para trabajar con video:** Aunque el software libre cuenta con un amplia gama de posibilidades que bien podrían sustituir a algunos programas privados, algunos otros programas libres no tienen el desarrollo suficiente para solucionar problemas de tipo profesional o avanzado, tal es el caso de programas de edición de video. Programas libres como Kdenlive aún no tienen el desarrollo para sustituir definitivamente a un programas como Premier o After Effects.

**Gimp carece de soporte modelo de color CMYK:** La capacidad de separar y luego editar una imagen en CMYK, es algo para lo que todavía Gimp no tienen soporte, en su lugar, existe un plug in para Gimp que se llama *Separate+* que tiene la capacidad de separar una imagen RGB, gestionar el color en perfiles ICC y LCMS y adjuntar perfiles ICC a nuestro diseño. Gimp tampoco soporta más de 8 bits por canal y solo soporta 3 canales en lugar de 4. Sin embargo, el programa Scribus sí posee soporte para CMYK. Hay usuarios que dicen que tanto como Gimp como Inkscape son una buena solución para diseño destinado a la web, pero que para diseño destinado a la impresión, aún tiene mucho camino por recorrer.

Es así como encontramos una alternativa al uso del software de patente, al cual hemos llamado también software privado. Con el software libre se abre una nueva posibilidad para crear digitalmente y un nuevo terreno para la exploración y la experimentación; por supuesto, sin perder de vista que, al ser un software desarrollado por la comunidad, requiere e incluso depende de la estrecha interacción de los usuarios para generar programas de calidad. En el siguiente capítulo hablaremos de cómo el software libre representa una opción viable y eficaz para el desarrollo de la actividad profesional.

1Richard Stallman es considerado el padre del Software Libre por ser el que acuñó el concepto y se lo otorgó a los sistemas operativos y programas que cumplieran con las 4 libertades que aseguran la libertad para los usuarios.

2Sam Williams Libre como en libertad, (Madrid, O'Reilly & Associates, Inc, 2002) 6.

3Williams, "Libre como..." 6.

4Williams, "Libre como..." 9.

5Williams, "Libre como.." 10.

6 "Visión general del proyecto GNU", Historia del proyecto GNU, (Enero 2008, [citada el 6 de junio del 2009]), GNU Opening Systems. Disponible en <http://www.gnu.org/gnu/gnu-history.es.html>

7A lo largo de esta tesis, nos referiremos al Software Libre, ya sea con su nombre completo o con las siglas SL.

8 Richard Stallman Software libre para una sociedad libre, (Madrid, Traficantes de sueños, 2004) 47.

9 Neil McAllister "Devices provide a fertile new ground of Linux", Computerworld. (18-Sep-2006 [citado el 7 de Julio del 2009]) disponible en: <http://computerworld.co.nz/news.nsf/tech/2E968937ADA351D8CC2571EA0013F01B>

10Erick Raymond La catedral y el bazar (Argentina, Openbiz,1997) 16.

11 Stallman, "Software Libre," 59.

12CDS cuyas siglas en ingles significan Counterfeit Deterrence System, que podría traducirse como un Sistema de Disuasión de Falsificaciones.

13Adobe, "Photoshop and CDS" Adobe Photoshop CS5 (Julio 2009 [citado el 8 de agosto del 2011]) disponible en: <http://www.adobe.com/special/products/Photoshop/cds.html> [2].

14Brad Stone, "Amazon Erases Orwell Books From Kindle" The New York Times, (17 de Julio 2009 [citado el 8 de agosto del 2011]) Sección: Companies, Technology, disponible en: <http://www.nytimes.com/2009/07/18/technology/companies/18amazon.html> [3]

15DRM Digital Rights Management, termino para designar a las tecnologías de control de acceso para limitar el

uso de contenidos digitales y que impide el uso del contenido digital que no es deseado o pretendido por el proveedor de contenido.

16 José Flores, "Mozilla Firefox, traducido a lenguas prehispánicas" Alt1040 (16 de Agosto de 2010 [citada en Julio del 2011]) ed. Hipertextual, disponible en: <http://alt1040.com/2010/08/mozilla-firefox-traducido-a-lenguas-prehispanicas>

17 En iniciativas como Runasimitpi Qespisqa Software <http://runasimipi.org> [4]

18 Ralph Levien, "Ralph Levien's Thesis" Sitio web personal de Ralph Levien (Septiembre 2009 [citada en Julio del 2011]) ed. Ralph Levien, disponible en: <http://www.levien.com/phd/phd.html>

19 Inkscape es un software libre para la creación y edición de vectores, un homólogo a Illustrator o al Corel Draw.

20 FontForge es un programa libre para editar fuentes tipográficas.

21 Ralph Levien, "Spiro" Sitio web personal de Ralph Levien (Septiembre 2009 [citada en Julio del 2011]) ed. Ralph Levien, disponible en: <http://www.levien.com/spiro/>

22 Ralph Levien, "Fonts in progress" Sitio web personal de Ralph Levien (Septiembre 2009 [citada en Julio del 2011]) ed. Ralph Levien, disponible en: <http://www.levien.com/type/myfonts/>

23 Lawrence Lessing Cultura Libre, Trad. Antonio Córdoba/Elástico ( E.U. Penguin Books, 2004)

24 Linux Devices, "SFLC, BusyBox, Monsoon dismiss GPL lawsuit" Eweek Linux Devices (30 de octubre del 2007 [citado en Agosto del 2011]) ed. Linux Devices, disponible en: <http://www.linuxfordevices.com/c/a/News/SFLC-Busy-Box-Monsoon-dismiss-GPL-lawsuit/> [5] recuperada en agosto del 2011.

25 Stallman, "Software libre," 63.

26 El concepto de "autor" separado del de "dueño". Donde el autor no pierde el crédito por la obra pero tampoco se "adueña" de ella para restringir su posterior uso, mejora, estudio y distribución.

27 Lessing, Cultura Libre, 34.

28 Stallman, "Software Libre," 22.

29 Stallman, "Software Libre," 133.

[30](#) Stallman, "Software Libre," 135.

[31](#) A partir de ahora, para referirnos al él, usaremos las palabras software privativo o bien sus siglas S.P.

[32](#) Stallman "Software libre para," 13.

[33](#) Stallman "Software libre para," 13.

[34](#) Stallman "Software libre para," 283.

[35](#) José Serralde, "Opera, norteros y software libre (I)" Sonoblog, música, tecnología, cultura libre y sonidos por dentro, (10 de marzo del 2010 [citado en septiembre 2010]) ed. José Serralde, disponible en: <http://blog.joseserralde.org/opera-video-norteros-y-software-libre>

[36](#) Ton Roosendaal, "About" Sintel, the Durian Open Movie Project (citada en Agosto del 2011) Sintel ed. Blender Open Movie Project, disponible en: <http://www.sintel.org/about/>

[37](#) Eschoyez Martín, extracto de entrevista (Córdoba Argentina, Junio 2011)

[38](#) Mansoux y Valk Floss and Art. 13.

[39](#) "Illustrator CS5, especificaciones técnicas" Adobe (citada en Julio del 2011) disponible en: <http://www.adobe.com/es/products/illustrator/tech-specs.html>

[40](#) GTK+ o The Gimp Toolkit es un conjunto de subprogramas que contienen código y datos y proporcionan servicios a programas independientes, es decir, pasan a formar parte de éstos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular. Este subprograma o biblioteca multiplataforma en particular sirve para desarrollar interfaces gráficas de usuario principalmente para los entornos gráficos GNOME, XFCE y ROX, aunque también se puede usar en el escritorio de Windows, MacOS y otros.

[41](#) "Open Standards" Document Freedom Day (Primavera 2010 [citado en septiembre 2010]) ed. FSFE, disponible en: <http://documentfreedom.org/2011/os.en.html>

[42](#) Root es el archivo raíz y tiene todos los derechos sobre el sistema. Todo lo que esta accesible dentro de Linux se encuentra en una subcarpeta de la raíz, se representa mediante /

[43](#) De user do, en español, el usuario hace.

44Una terminal en términos de computo es la abstracción generalizada, que comprende tanto una [interfaz de programación de aplicaciones](#) [6] para programas y un conjunto de expectativas de comportamiento para los usuarios,

45Greg Kroah-Hartman , Jonathan Corbet y Amanda McPherson , "Linux Kernel Development, How Fast it is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It" (August 2009 [citado en agosto del 2010] The Linux Foundation) pdf disponible en: [www. \[7\]linux \[8\]foundation.org/sites/main/files/publications/whowrites \[9\]linux \[10\].pdf \[11\]](http://www.linuxfoundation.org/sites/main/files/publications/whowrites-linux.pdf)

46"50 Places Linux is Running That You Might Not Expect" Focus (2011 [citado en diciembre 2011]) ed. Focus Editors, disponible en: <http://www.focus.com/fyi/50-places-linux-running-you-might-not-expect/>

47El precio de la Creative Suite CS5.5 Design Premium en 2011, del sitio oficial de Adobe para latinoamérica: <http://www.adobe.com/mx/products/creativesuite/design.html>

48General Public Licence

49Gnu is Not Unix, refiriéndose a la base del sistema operativo linux que no tiene que ver con su antecesor Unix que después se volvió privativo.

50Stallman, "Software libre," 28.

51Un Fork, traducido al español significa bifurcación. Se trata de la creación de un proyecto en una dirección distinta de la principal u oficial tomando el código fuente del proyecto ya existente. Comúnmente se utiliza el término inglés.

52Oyarzún Christian, extracto de entrevista (Santiago de Chile, Mayo del 2011).

53Aymeric Mansoux y Marloes de Valk , "Prefacio" en Floss and art ed. Aymeric Mansoux y Marloes de Valk (Francia, GOTO10 y OpenMute, 2008) 10.

54Espertino Guillermo, extracto de entrevista (Rosario, Argentina en Mayo 2011).

55Eschoyez Martín, extracto de entrevista (Córdoba, Argentina en Junio 2011).

56Pagola Lila, extracto de entrevista (Córdoba, Argentina, Mayo del 2011)

57Eschoyez Martín, extracto de entrevista (Córdoba Argentina en Junio del 2011).

58Santorcuato José, extracto de entrevista Santiago de Chile, Mayo 2011).

59Caiazza Fabricio, extracto de entrevista, (Rosario, Argentina, Mayo 2011).

60Red Hat, "Open Source Activity Map – 2008" (2009 [citada en octubre2009] Red Hat) ed. Red Hat Enterprise Linux disponible en: <http://www.redhat.com/about/where-is-open-source/activity/> [12]

61Benjamin Stephan y Lutz Vogel "Trusted Computing" MPG4. Dirigido por: Benjamin Stephan y Lutz Vogel (E.U.:LAFKON, 2005). Disponible en: <http://www.lafkon.net/tc/>

62Las llamadas "distro" entre la comunidad del Software Libre que son las diferentes versiones de distemas operativos, lo que en caso de Microsoft sería los conocidos Windows xp o Windows vista y en Mac, el Leopard o el OS X

63El programa libre para la diagramación y diseño de paginas de libros y revistas, pertenecientes a los llamados desktop-publisher, o bien, la versión libre de Indesign.

64Para términos del diseño editorial de la presente tesis, comencé a experimentar con Scribus y comienzo con su descubrimiento, hasta el momento he encontrado esa deficiencia, aunque sigo trabajando en ello.

65Espertino Guillermo, extracto de entrevista Rosario, Argentina, mayo 2011.

#### Links externos

[1] <http://registry.gimp.org/>

[2] <http://www.adobe.com/special/products/photoshop/cds.html>

[3] <http://www.nytimes.com/2009/07/18/technology/companies/18amazon.html>

[4] <http://runasimipi.org/>

[5] <http://www.linuxfordevices.com/c/a/News/SFLC-BusyBox-Monsoon-dismiss-GPL-lawsuit/>

[6] [http://translate.googleusercontent.com/translate\\_c?rurl=translate.google.com&sl=auto&tl=es&twu=1&u=http://en.wikipedia.org/wiki/Application\\_Programming\\_Interface&usg=ALkJrhgXMgJ-LIt0IMmm8unImMchedBCJg](http://translate.googleusercontent.com/translate_c?rurl=translate.google.com&sl=auto&tl=es&twu=1&u=http://en.wikipedia.org/wiki/Application_Programming_Interface&usg=ALkJrhgXMgJ-LIt0IMmm8unImMchedBCJg)