

**CONGRESO
IBEROAMERICANO**
DE CIENCIA, TECNOLOGÍA,
INNOVACIÓN Y EDUCACIÓN

BUENOS AIRES, ARGENTINA
12, 13 Y 14 DE NOVIEMBRE 2014

**CONGRESSO
IBERO-AMERICANO**
DE CIÊNCIA, TECNOLOGIA,
INOVAÇÃO E EDUCAÇÃO

BUENOS AIRES, ARGENTINA
12, 13 Y 14 DE NOVIEMBRO 2014

SOFTWARE LIBRE COMO ALTERNATIVA DE CALIDAD EN LA EDUCACIÓN

HERNANDEZ, GONZALEZ, F,J; HUERTA, GARCIA, L;
MENDEZ, GUEVARA, L,C

SOFTWARE LIBRE COMO ALTERNATIVA DE CALIDAD EN LA EDUCACIÓN

Francisco Javier Hernández González¹

Lucero Huerta García²

Laura Cecilia Méndez Guevara³

Centro Universitario UAEM Valle de Teotihuacán.

Cerrada de Nezahualcóyotl S/N, Sto. Domingo Aztacameca, Axapusco, Edo.
Méx. México.

¹ paco_dkz23@hotmail.com

² dulcero.huerta@hotmail.com

³ lceci_l@yahoo.com.mx

INTRODUCCIÓN

En la actualidad uno de los temas de mayor impacto, es hablar de la implementación de las Tecnologías de la Información y las Comunicaciones (TIC' s) en el entorno educativo, priorizando el tema como una problemática de solución pronta, ya que los métodos de enseñanza deben adecuarse para utilizar recursos didácticos, basados en el uso de las computadoras para dar respuesta a las verdaderas necesidades que actualmente la educación enfrenta.

En México no hay una ley que determine el uso de una tecnología en particular, lo que se conoce como neutralidad tecnológica. Al no enfocarse en marcas o plataformas tecnológicas específicas, los usuarios tienen la oportunidad de escoger las soluciones que satisfagan mejor sus necesidades.

Así el 67 % de las empresas en México tiene *software* de forma ilegal o pirata. El 60 % de las empresas en México y Latinoamérica se encuentra en la etapa de evolución, implementación o crecimiento. A pesar de que el uso de *Software* Libre en la educación en México está creciendo, su adopción no es tan rápida como se pregona, debido a factores de conocimiento.

A la hora de incorporar las TIC' s en la Educación, se presentan diversos inconvenientes, uno de los más importantes es el alto costo de adquisición. Los costos asociados a Infraestructura de redes, telecomunicaciones, conectividad, *hardware*, *software*, capacitación de personal, instalación y mantenimiento, pueden ser una de las mayores limitantes. (Jiménez O., Vasquez, Checa H., González G., & Méndez A., 2012)

Con los avances tecnológicos actuales y la trayectoria histórica de la educación en México, se ha marcado la evolución de los recursos informáticos, que las nuevas generaciones pueden acceder y obtener, sin que esto signifique una violación de derechos de autor y pagos de licencia, se han generado sistemas operativos completos, intuitivos, robustos, entre otros, y también un sinnúmero de aplicaciones que pueden ser utilizadas en diferentes áreas.

Por ello, el *software* libre ha pasado de ser un simple modelo de desarrollo de *software* (con todas sus implicaciones técnicas y éticas) a ser un elemento clave en las estrategias de desarrollo de empresas, instituciones, regiones e incluso países enteros. Se visualiza como una alternativa viable para la implementación de recursos informáticos, en el sector educativo, sea cual fuera el nivel para su aplicación.

Esto ha sido la motivación de utilizar herramientas informáticas libres y de calidad, que permitan desempeñar la misma funcionalidad que alguna aplicación de *software* de propietario. Permitiendo al sector el desarrollo y potenciación de una sociedad fuerte, capaz y competitiva en el ámbito tecnológico.

Software Libre

Libertad de copiar y redistribuir un programa.

Ha pasado de ser un simple modelo de desarrollo de software, a ser un elemento clave en las estrategias de desarrollo de empresas.

¿Qué es?

Impacto social

Completamente gratuito

Costo

Ventajas

- Ahorro en la adquisición de licencias.
- Disminuye la copia ilegal de software.

Figura1. *Software Libre*. Fuente: propia.

Principios

El *software* tuvo un origen con características libres, aunque no haya sido definido como lo que hoy se conoce con ese término. En los años sesenta, las computadoras se entregaban con todos los programas necesarios para su funcionamiento, exceptuando las aplicaciones comerciales que debían ser desarrolladas de manera particular. Los programadores eran más artistas que técnicos y se concentraban en el proceso de escribir los programas ya que era común que compartieran el código fuente con otros desarrolladores.

Los años 60's se ven marcados por la presencia de IBM, quien con sus modelos 360 realizó el primer intento de comercializar los programas. Pero esto comenzó a crear ciertas barreras entre los desarrolladores de *software* ya que compartir el código era casi como compartir dinero. Así surge un *software* libre llamado *Tex* desarrollado por *Donald Knuth*, consistiendo en un sistema tipográfico para escribir documentos profesionales.

El sistema operacional *Unix*, creado por *Thompson* y *Ritchie*, en la *Bell Labs* de *AT&T* en 1972, que marca una época de vital importancia para la humanidad, ya que de ahí vendría en desarrollo posterior de lo que hoy es la internet.

El sistema operacional de origen empresarial fue entregado a las universidades estadounidenses, la aplicación fue usada en la área académica incluidos los programas fuentes. Esto estimulo el trabajo en grupo para mejorar *Unix* y crear estándares, especialmente con la participación de la universidad de *Berkeley*, que alentó a los desarrolladores de *software*, lo cual impulso la creación de una de las primeras comunidades llamada *CSRG* o *Computer System Research Group*.

Sin embargo la compañía *AT&T* trato de cerrar el código fuente y se creó un movimiento comercial alrededor del programa, al punto que la compañía demando a la universidad de *Berkeley* por la publicación de código fuente de su *Unix* llamado *BSD Unix*.

Al comienzo del proyecto se incluyó *software* como *Tex*; escribió un compilador de lenguaje C, llamado *GCC* o *GNU C Compiler*; un editor llamado *Emacs* que era competencia con *Vi* el editor más famoso de *Unix*.

Una de las preocupaciones de este proyecto era definir las libertades que tendría un usuario que usara este software para lo cual era necesario crear una licencia. Y así surge la licencia *GPL* (*General Public Licence*) que daría todas las libertades de uso al usuario denominando estos derechos *copy left* como indicativo de "izquierda de copia" en contraposición del *copy righth* que se refiere a "derecho de copia".

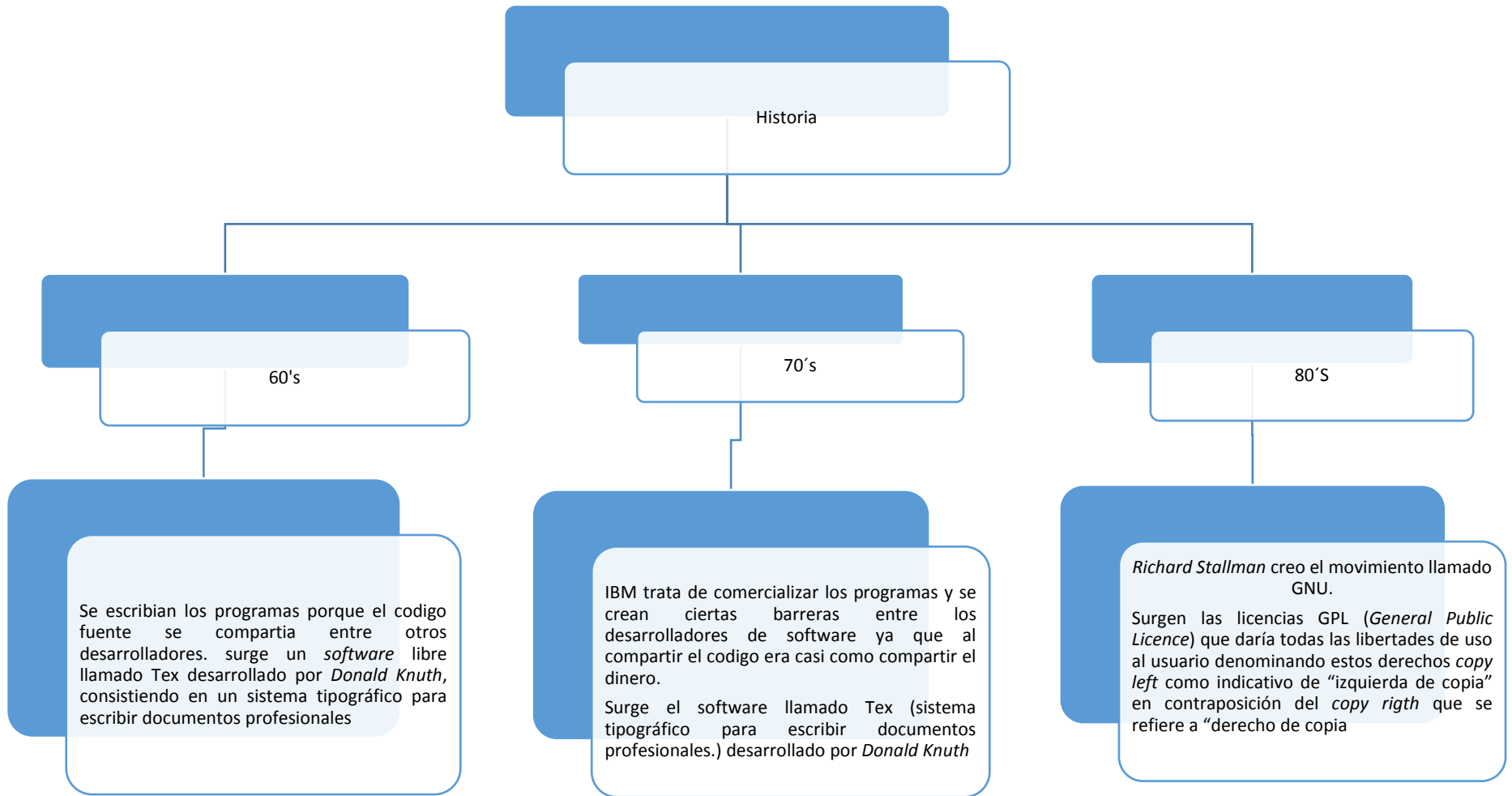


Figura 2. Historia del *Software Libre*. Fuente: basado en (Sguerra, págs. 8,14).

Iniciativas clave para el desarrollo de *software* libre

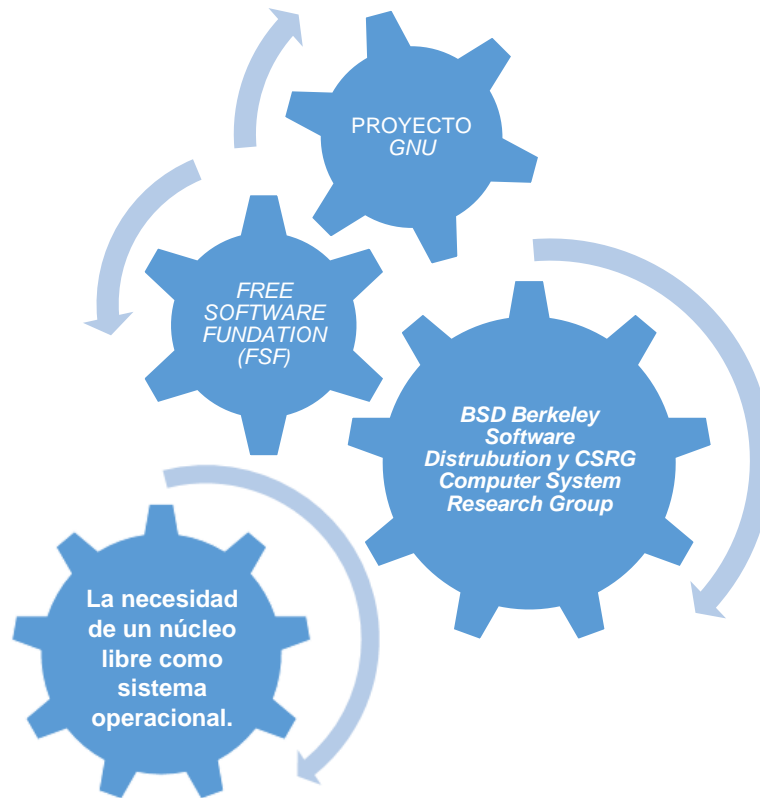


Figura 3. Iniciativas claves para el desarrollo de *software* libre. Fuente: propia.

- **El proyecto GNU**

Richard Stallman, investigador del laboratorio de Inteligencia Artificial del MIT (*Massachusetts Institute of Technology*), fue el primero en levantarse contra la posibilidad de convertir el *software* en un producto comercializable. Crea en 1984 el denominado GNU (*GNU is Not Unix*), proyecto para la creación de *Software Libre* y de Código Abierto.

A partir de ese momento el proyecto se abrió a la comunidad con la participación de programadores de todo el mundo, que reprodujeron a una escala mucho mayor el esquema seguido por gran parte de las aplicaciones GNU. (Stallman, 2014)

La posibilidad de que cualquier persona con la capacidad y conocimientos necesarios pudiera aportar su granito de arena, llevaba implícita la necesidad de que el código fuente del sistema operativo fuera público (Código Abierto u *Open Source*).

Código Abierto no solo significa acceso al código fuente. Los términos de distribución del *software* de código abierto deben cumplir los siguientes criterios.

1. Libre Redistribución.

La licencia no restringirá la venta o cesión del *software* como componente de una distribución de *software* que contenga programas de varios códigos distintos.

2. Código Fuente

En el caso de que el producto distribuido no incluya el código fuente debe haber una forma suficientemente clara de obtenerlo, por un precio razonable no mayor del coste de reproducción, preferentemente a través de la descarga de Internet sin coste alguno. El código fuente debe ser la forma preferida en la que un programador pueda modificar el programa. No se permitirá el código fuente deliberadamente al revés, ni tampoco formas intermedias como salidas del preprocesador o traductor.

3. Trabajo derivado

Los trabajos derivados deben estar permitidos en las licencias así como la modificación libre y la distribución en los mismos términos de la licencia del *software* original.

4. Integridad del código fuente del autor

La licencia puede restringir la distribución modificada del código fuente sólo si la licencia autoriza la distribución de *patch files* con el código fuente con la finalidad de modificar el programa en tiempo de compilación. La licencia debe hacer explícito el permiso de distribución de *software* generado a partir de código fuente modificado.

5. Evitar la discriminación contra personas o grupos de personas.

La licencia no puede discriminar a ninguna persona o grupo de personas

6. Disminuir la discriminación contra áreas de trabajo.

La licencia no puede restringir a nadie el uso del programa en un campo específico de trabajo.

7. Distribución de la licencia.

Los derechos vinculados al programa deben ser de aplicación para todos aquellos a los que el programa sea redistribuido, sin necesidad de la ejecución de una nueva licencia para aquellas partes.

8. La licencia no debe ser específica para un producto

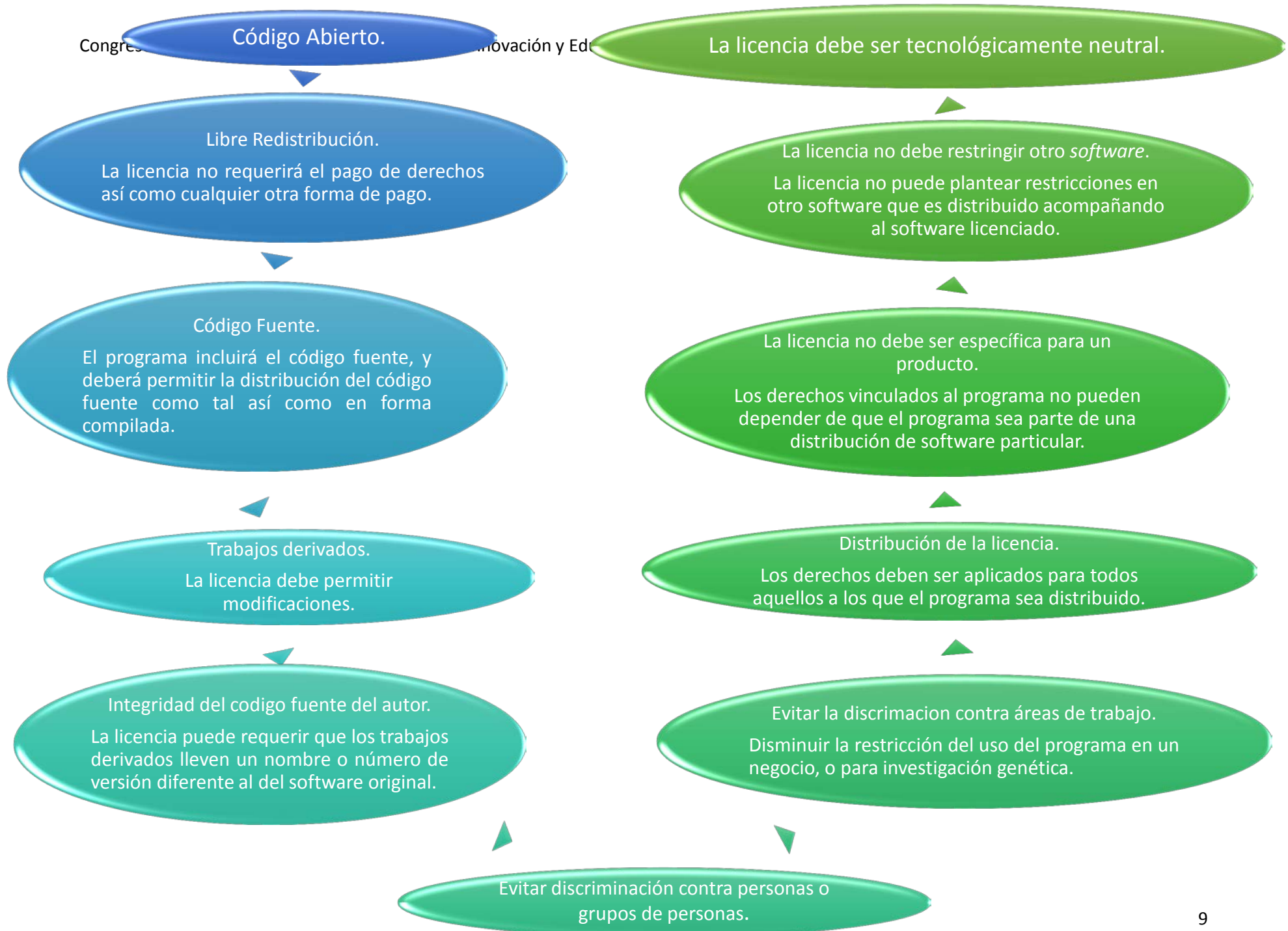
Si el programa se extrae de esa distribución y se usa o distribuye de acuerdo a los términos de licencia del programa, todas las partes a las que se distribuye el programa deben tener los mismos derechos que los que están licenciados en la distribución de *software* original.

9. La licencia no debe restringir otro *software*.

La licencia no puede insistir en que todos los restantes programas distribuidos en el mismo medio deben ser de Código Abierto.

10. La licencia debe ser tecnológicamente neutral

Ninguna previsión de la licencia puede estar dedicada a una tecnología individual o estilo de interfaz. (Initiative, 2007) Ver figura 4.



ISBN: 978-84-7666-210-9 Artículo 564 **Figura 4. Criterios a cumplir para la distribución del software de código abierto, Fuente: propia.**

- **Free Software Foundation (FSF)**

Un año más tarde (1985), Stallman junto con otros programadores crearían la Fundación para el Software Libre (FSF) que daría lugar al nacimiento de la licencia GPL (*General Public License*) para proteger al Software Libre y para conseguir fondos para el desarrollo de más *software*. La creación de la fundación se identifica con el manifiesto por medio del cual fundamenta las bases éticas del *software* libre, documento denominado *The GNU Manifiesto* y *Why Software Should Not Have Owners*. En estas épocas el internet no tenía el impacto de nuestros días y eran comunes las cintas magnéticas para la distribución del *software*, entonces el beneficio era por el material entregado y algo más por el servicio de copiado. En esos años Stallman quiso desarrollar el núcleo de un sistema operacional propio llamado *Hurd*, que aún sigue en desarrollo. (Foundation, 2014) Ver figura 5.

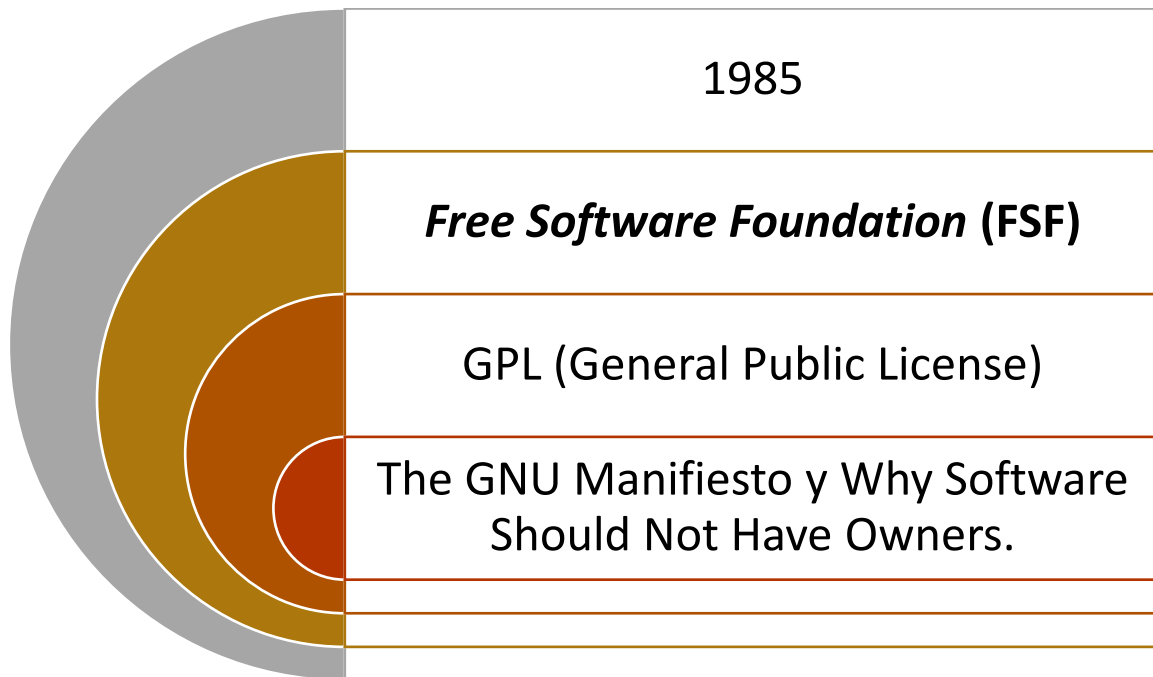


Figura 5. Nacimiento de la FSF. Fuente: propia.

- **BSD Berkeley Software Distribution y CSRG Computer System Research Group**

El CSRG (*Computer System Research Group*) de la Universidad de California en *Berkeley*, fue el que en realidad desarrollo a *Unix*, entre 1973-1980, muy reforzado por el compromiso del departamento de defensa de EE.UU. para crear un protocolo de comunicaciones entre computadoras dentro del proyecto DARPA que fue el precursor del protocolo TCP/IP el alma de internet. (Distribution)

El sistema *Unix* salió al mundo académico con el nombre de *BSD Unix*, unido a una licencia llamada *BSD*, denominada minimalista, que además de permitir la redistribución libre, también permitía su incorporación a productos propietarios, con un método de distribución parecido al de la FSF de cintas magnéticas a bajos precios. Así surgen una serie de familias de *Unix* como *BSD*, *NetBSD*, *FreeBSD*, *OpenBSD*, *BSDI*, entre otros. Ver figura 6.

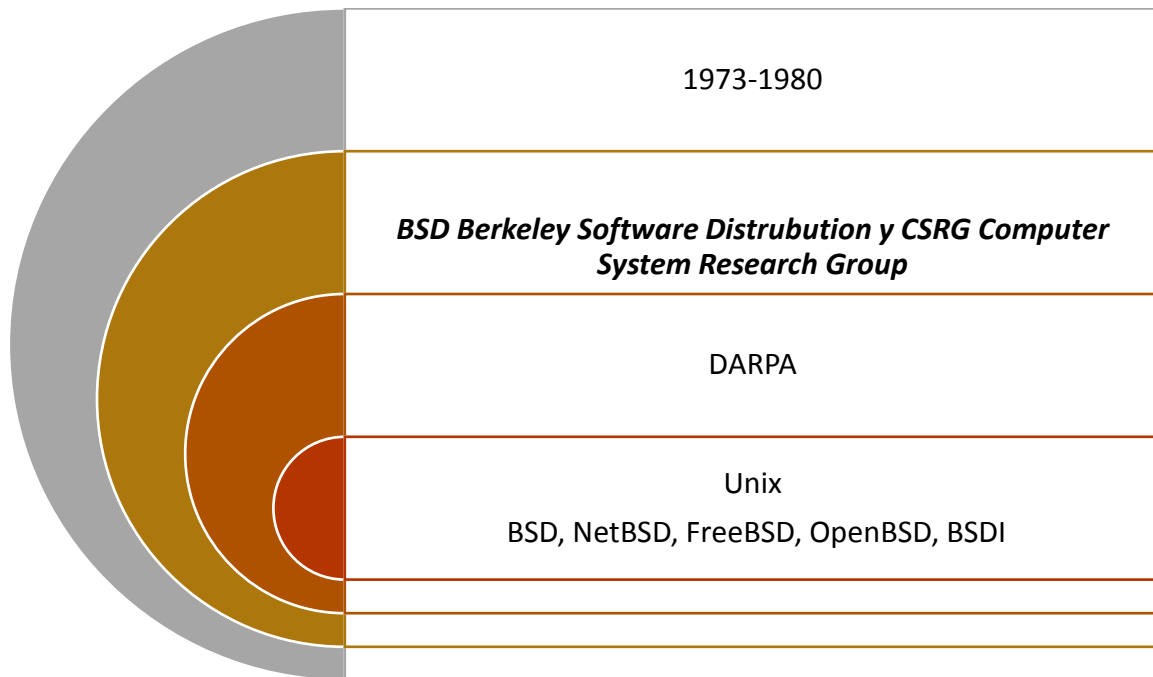


Figura 6. *BSD* y *CSRG*, fuente propia.

- **La necesidad de un núcleo libre como sistema operacional**

Al pensar Stallman desarrollar su propio núcleo llamado *Hurd*. En 1990 entre *BSD*, *GNU*, el sistema de ventanas de *Unix Xwindow* y otros programas se había construido un ambiente de desarrollo muy poderoso, pero no existía un núcleo o kernel bajo la *GPL*.

Un estudiante Finlandés llamado Linus Torvalds, seguidor de Andrew Tanenbaum – el autor del primer sistema operacional académico abierto llamado *Minix*, decidió el Julio 1991, crear su propio núcleo bajo la licencia *GPL* que vino a aparecer en su primera versión estable en Marzo de 1994, naturalmente publicada en Internet. Así cientos de desarrolladores se le unieron para integrar sobre el núcleo todo el *software* existente y mucho más. Así surge el sistema operativo *Linux* y con el tiempo aparecen algunas distribuciones que hoy en día conocemos como: *Debian*, *RedHat*, *SuSE*, *Mandriva*, con esto se pensara que son *Linux* diferentes por no *Linux* solo es uno. (Sguerra). Ver figura 7.

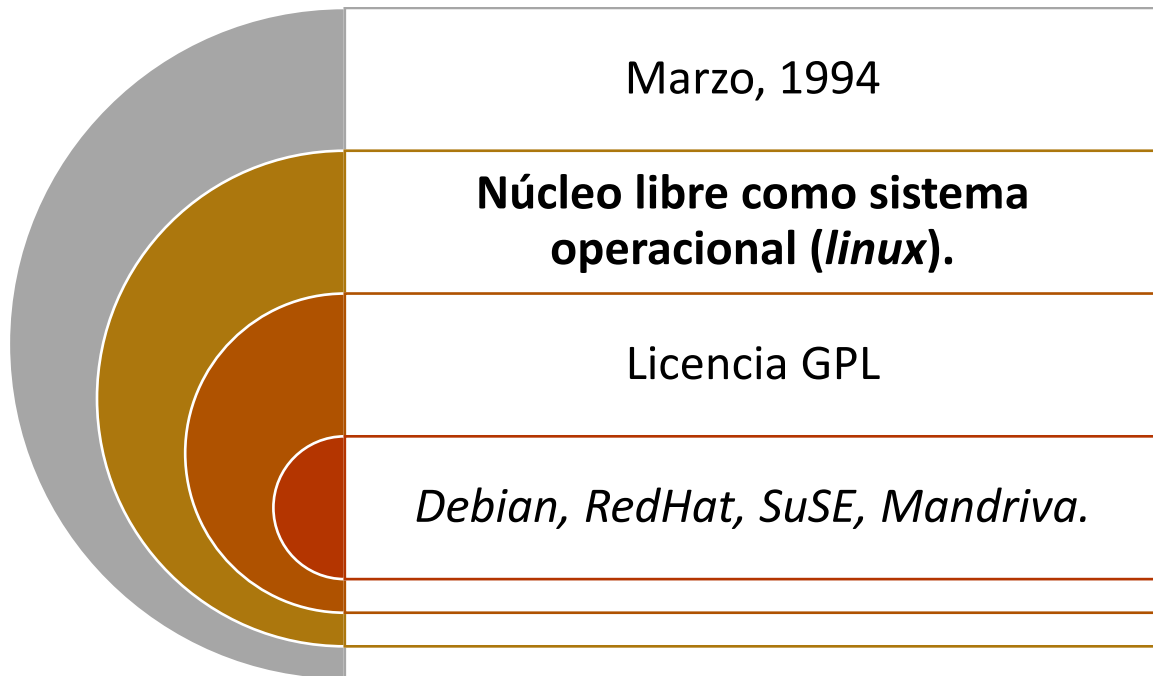


Figura 7. Surge *Linux*, fuente propia.

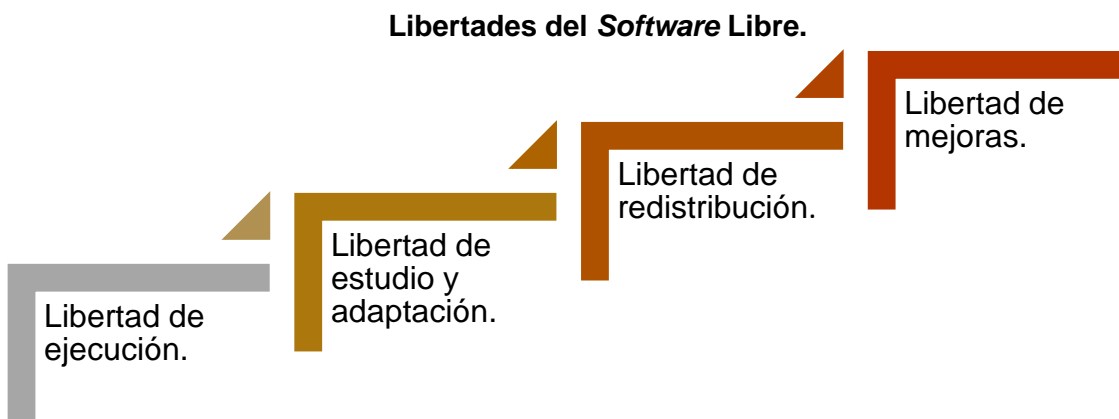


Figura 8, Libertades del *software* libre, Fuente: propia.

El término *software* libre (o programas libres) se refiere a la libertad. En concreto se refiere a cuatro libertades:

1. Libertad para ejecutar el programa en cualquier sitio, con cualquier propósito y para siempre.
2. Libertad para estudiarlo y adaptarlo a nuestras necesidades. Esto exige el acceso al código fuente.
3. Libertad de redistribución.
4. Libertad para mejorar el programa y publicar las mejoras. También exige el código fuente.

El *Software Libre* no tiene restricciones de uso de ningún tipo, sean políticas, geográficas, ideológicas. Cualquier persona interesada puede usarlo, mejorarlos y compartirlos. Igualmente puede distribuir cuantas copias desee sin problemas legales, ya que está contemplado por licencias que se cuenta con el permiso. (Arenas M., Fernández de Viana, & Abad Herrera, 2010)

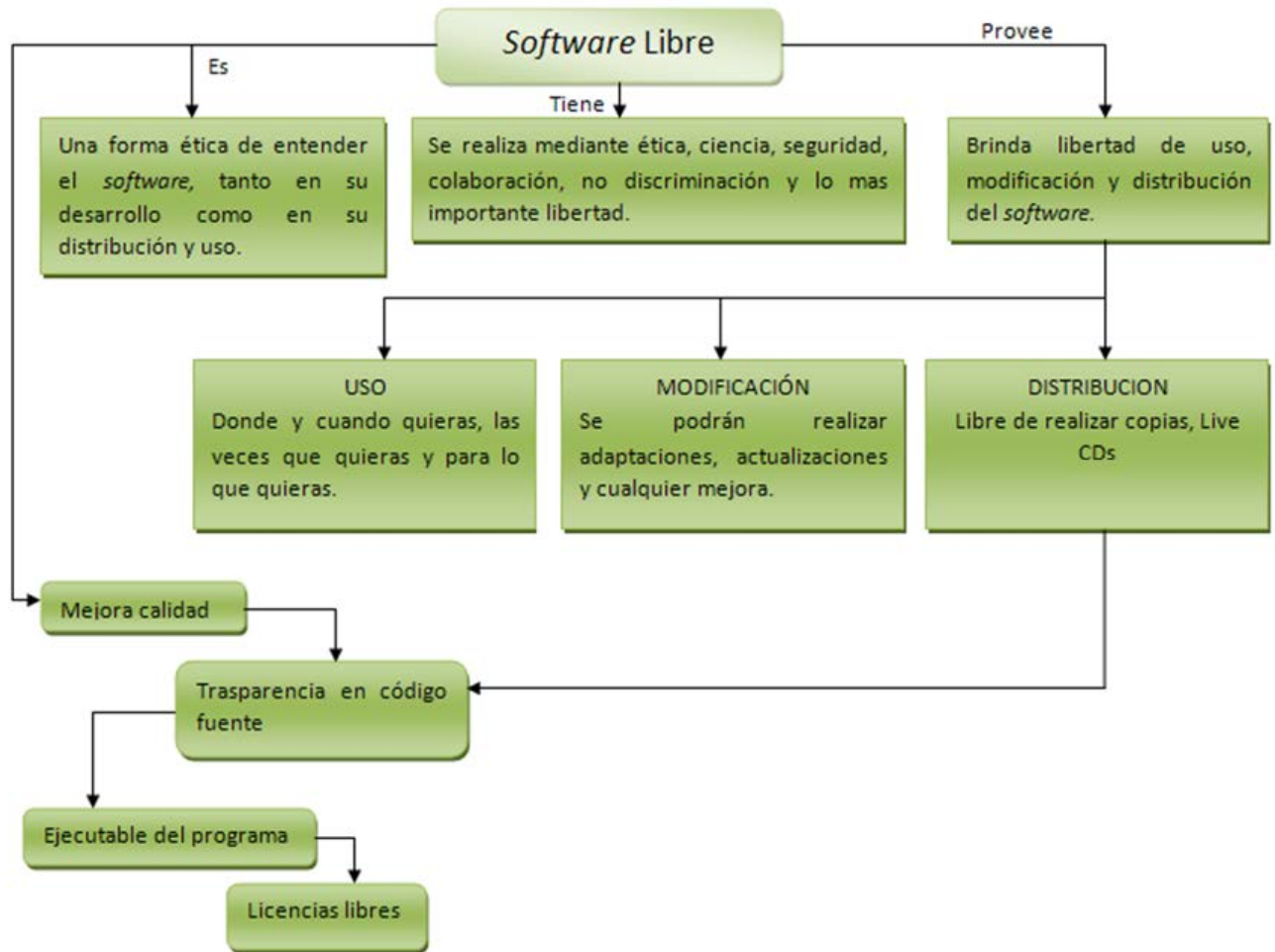


Figura 9. *Software libre* ventajas y usos. Fuente: propia.

Software Libre como alternativa de calidad en la educación

En el entorno educativo, además de las ventajas de tipo económico, el uso e implementación de *software* libre trae consigo muchos beneficios. Ya que el *software* libre es desarrollado por miles de personas geográficamente dispersas, con costumbres, ideologías y pensamientos diferentes, así el paradigma del *software* libre invita a que se genere cooperación, colaboración y reconocimiento de las diferencias como una forma de enriquecimiento y fortalecimiento mutuo, valores que deben ser impartidos desde la escuela a los estudiantes de manera que se generen estilos de vida beneficiosos para la sociedad en conjunto.

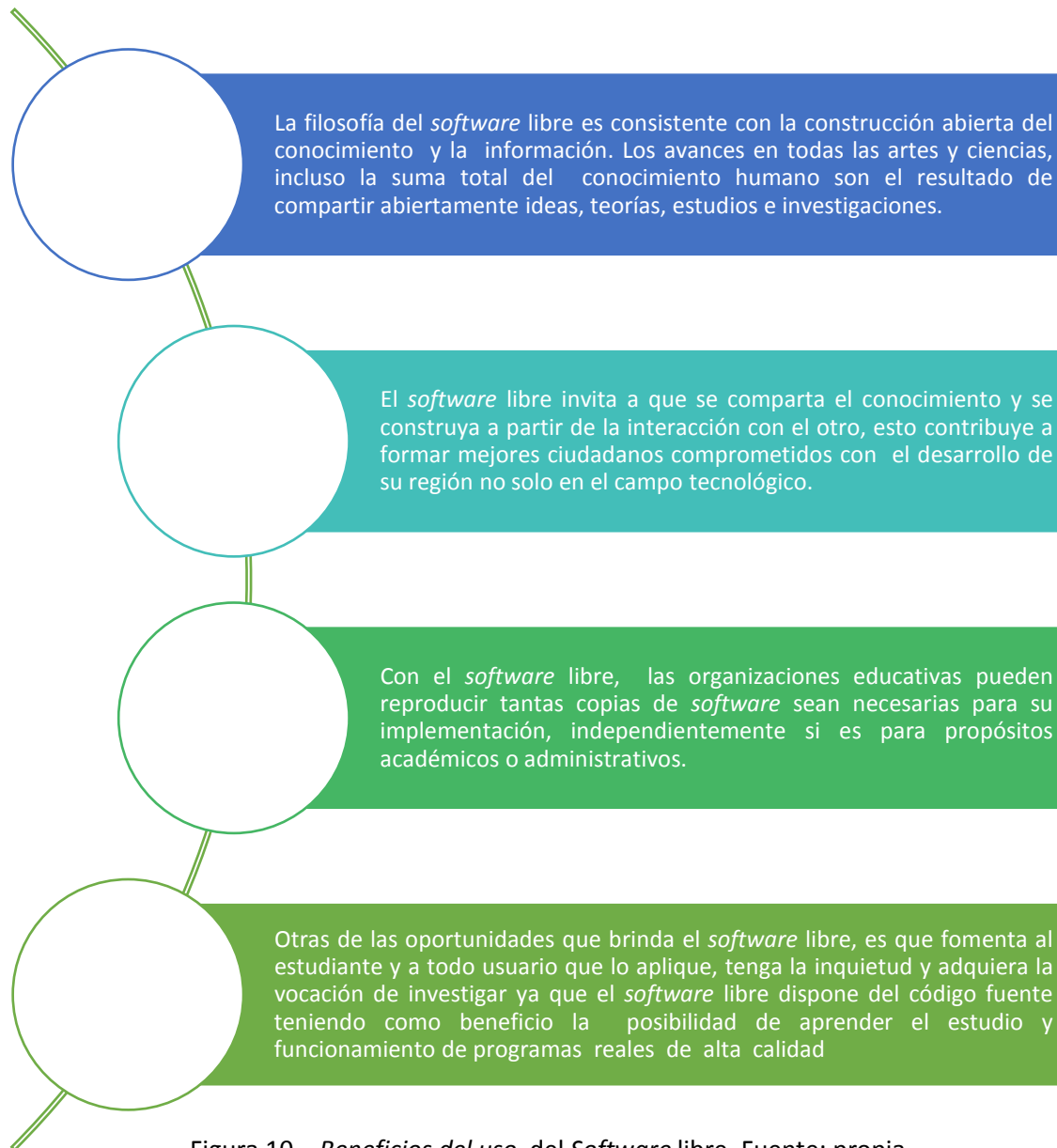


Figura 10. *Beneficios del uso del Software libre.* Fuente: propia.

Por otra parte existen algunas dependencias educativas que no pueden pagar las altas licencias de *software* de propietario por su elevado costo de adquisición y suelen recurrir al uso de copias ilegales de este tipo de *software*, cayendo por necesidad en un delito que puede ser sancionado.

Los estudiantes y docentes pueden copiar y compartir programas, incluso fuera de la institución a la que pertenezcan, sin estar incurriendo en un acto de piratería

Adicionalmente al tener la oportunidad de modificar el programa es posible adaptarlo al contexto específico que se requiera, con el fin de optimizar el uso e implementación del mismo.

Sin duda uno de los problemas que comúnmente se presenta en las escuelas de educación, del nivel que sean, es que cuentan con recursos *hardware* de muy bajas características, lo que incide a que se tenga *software* desactualizado, sin embargo existen herramientas *software* que optimizan y potencializan dichos recursos, de manera que, a pesar de no ser de muy buenas especificaciones se puede implementar con sistemas operativos y aplicaciones que se adapten a las necesidades y exigencias pedagógicas actuales de algún sistema educativo.

Todos estos beneficios contribuyen al ideal de generar una sociedad más incluyente, equitativa y solidaria, donde la tecnología logre estar al alcance de todos independientemente de las condiciones socio-económicas y socio-culturales del entorno en el que se desarrollen, donde cada individuo tenga la posibilidad de participación en la dinámica de la construcción cooperativa del conocimiento y así su aporte sea vital para el fortalecimiento de los lazos de comunidad y desarrollo social.

Es por eso que el sector educativo tendrá que apostar más al uso del *software* libre como una alternativa de calidad que permita gestionar un aprendizaje significativo a los estudiantes, logrando así una sociedad estudiantil autónoma, dependiente de sí misma, y contribuyendo al desarrollo sustentable de la sociedad en general.

De tal manera que viendo el *software* libre como una alternativa se empieza a justificar y a demostrar que es competitivo con cualquier otro *software* de propietario, identificando que cumple con una estructura similar. (Jiménez O., Vasquez, Checa H., González G., & Méndez A., 2012)

El *software* libre al igual que el software de propietario tiende a clasificarse internamente de la siguiente manera:

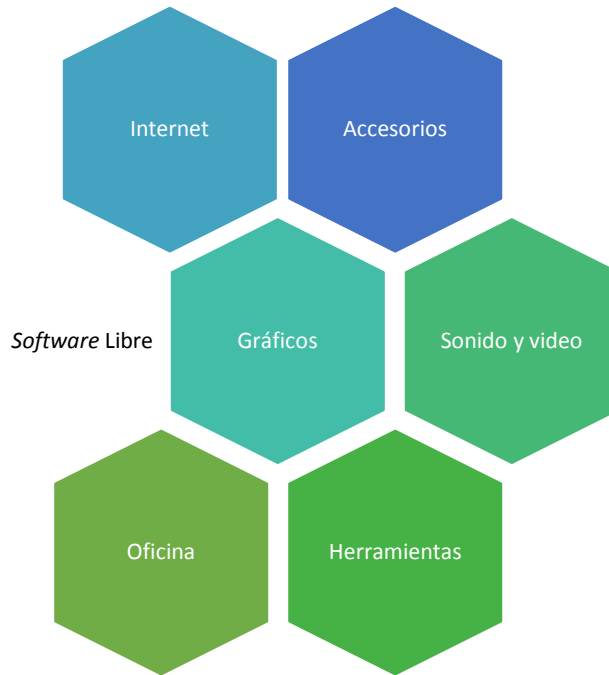


Figura 11. Clasificación interna del *Software* libre. Fuente: propia.

- Accesorios: se incluye todo aquel software que permite mejorar la experiencia del usuario; en diversos ámbitos.
- Gráficos: programas relacionados con la edición y visualización de todo tipo de formatos gráficos.
- Herramientas del sistema: aplicaciones encaminadas a realizar labores de mantenimiento interno.
- Oficina: todas aquellas herramientas que se utilizan dentro de una oficina (procesadores de texto, hojas de cálculo, etc.)
- Internet: aplicaciones que permiten la comunicación entre usuarios mediante la red.
- Sonido y vídeo: todo tipo de programas que permiten tanto editar como generar contenido multimedia. (Arenas M., Fernández de Viana, & Abad Herrera, 2010)

Teniendo la disponibilidad que para cada una de estas clasificaciones hay no solo un *software* libre sino varios que encajan perfectamente en alguno de estos puntos, adaptándose a las necesidades que se requieran cubrir en alguna determinada organización.

Cuadro comparativo.

Tabla 1 Comparación entre software libre y propietario de Sistemas Operativos. Fuente: propia.

	SOFTWARE PROPIETARIO	SOFTWARE LIBRE
Software Base	Sistemas operativos (para computadoras clientes y servidores)	
	Microsoft Windows 2012	Debian
	Microsoft Windows 7	CentOS
	Microsoft Windows 8	Ubuntu
	Servicios de comunicación (redes, correos, mensajería)	
	Wamp: Servidor web par ambientes Microsoft Windows	LAMP: Servidor web para ambientes GNU/Linux
	Microsoft Outlook	PHP
	MSN Messenger	Postfix
	Servicios de data (bases de datos, centralización de archivos)	
	MS SQL Server	MySQL Derby
Oracle	PostgreSQL SAMBA	

Cuadro comparativo.

Tabla 2 Comparación entre software libre y propietario Ofimática. Fuente: propia.

	SOFTWARE PROPIETARIO	SOFTWARE LIBRE
Software de Aplicación	Ofimática	
	Microsoft Office	OpenOffice.org
	Bloc de notas	LibreOffice
	Microsoft Visio	Dia
		Gedit
	Gestión de contenidos	
	HTML Y PHP, Páginas nativas desde código	Joomla
		Drupal
		Wordpress
	Clientes de comunicaciones (chat, correo, navegadores)	
MSN Messenger	Evolution	
Internet Explorer	Mozilla Thunderbird	
	Mozilla Firefox	

Tabla 3
Comparación entre software libre y propietario de diseño multimedia, fuente propia.

Software de Aplicación	Graficadores, herramientas para desarrollo de software, herramientas de CAD, multimedia	
	<i>Dreamweaver</i>	<i>Bluefish</i>
	<i>Adobe Photoshop</i>	<i>GIMP</i>
	<i>CorelDraw</i>	<i>Inkscape</i>
	<i>Microsoft Project</i>	<i>Planner</i>
	<i>Microsoft Visual Studio .NET</i>	<i>Open Atrium</i>
	<i>JCreator</i>	<i>Eclipse</i>
	<i>JDeveloper</i>	<i>Netbeans</i>

Con esto se presenta la incógnita de decidir qué tipo de *software* utilizar, pero si hablamos de *software* libre, se habla de un ahorro económico para las escuelas, pero este es un beneficio secundario. El ahorro es posible porque el *software* libre le da a las escuelas, igual que a cualquier otro usuario, la libertad de copiar y redistribuir el *software*. Así, el sistema educativo puede entregar una copia del programa a todas las escuelas, y cada una de ellas puede instalarlo en todos sus equipos que posea sin estar obligada a pagar por ello. La única necesidad que se tuviera, sería la de capacitar al personal estudiantil y administrativo para que logren familiarizarse con el nuevo sistema de *software* libre y puedan desempeñar sus labores más rápidamente y con una nueva visión futurista.

Bibliografía

- Arenas M., I., Fernández de Viana, I., & Abad Herrera, P. J. (2010). *Dialnet*. Obtenido de <http://dialnet.unirioja.es/servlet/articulo?codigo=3348294>
- Distribution, B. S. (s.f.). *BSD*. Obtenido de <http://bsd.org/>
- Foundation, F. S. (2014). *FSF*. Obtenido de <http://www.fsf.org/>
- Initiative, O. S. (2007). *Open Source Initiative*. Obtenido de <http://www.opensource.org/docs/osd.pdf>
- Jiménez O., J. E., Vasquez, K., Checa H., A. C., González G., G., & Méndez A., Y. A. (23 de Enero de 2012). *Eduka*. Obtenido de <http://www.gnu.org/education/edu-schools.es.html>
- Sguerra, M. D. (s.f.). *GNU/Linux y el Software libre y sus multiples aplicaciones*. México: AlfaOmega.
- Stallman, R. (2014). *GNU*. Obtenido de <http://www.gnu.org/>