Open Source Software

La libertad de elegir el software

# Introducción

El término software libre refiere el conjunto de software que por elección manifiesta de su autor, puede ser copiado, estudiado, modificado, utilizado libremente con cualquier fin y redistribuido con o sin cambios o mejoras. Su defi­nición está asociada al nacimiento del movimiento de software libre, encabezado por Richard Stallman y la consecuente fundación en 1985 de la Free Software Foundation, que coloca la libertad del usuario informático como pro­pósito ético fundamental.
Proviene del término en inglés free software, que presenta ambi­güedad entre los significados «li­bre» y «gratis» asociados a la pa­labra free.

Un programa informático es software libre si otorga a los usuarios todas estas libertades de manera adecuada. De lo contrario no es libre. Existen diversos esquemas de distribución que no son libres, y si bien podemos distinguirlos en base a cuánto les falta para llegar a ser libres, su uso bien puede ser considerado contrario a la ética en todos los casos por igual.

El software libre suele estar disponible gratuitamente, o al precio de coste de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar «software libre» a «software gratuito» (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente. Análogamente, el software gratis o gratuito incluye en ocasiones el código fuente; no obstante, este tipo de software no es «libre» en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con «software de dominio público». Éste último es aquel que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, consignando su autoría original. Este software sería aquel cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado. Si un **autor condiciona** su uso bajo una licencia, por muy débil que sea, ya no es del dominio público.

|  |
| --- |
| Atributos de carácter |
| Alienación izq. | Aline. Der | Aline. Centr. | Alienación justificada |
| Normal | Negrita | *Cursiva* | Subrayado |
| ***Negr. Cur. Sub.*** | Subr. doble | Tachado |  |
| Sub. solo palabr. | Versales o Versalitas | TODO EN MAYÚSCULAS | A2=B2+C2 |
| Sombra | Colores letras | Colores fondos | H2SO4 |
| Apretado | Separado | Muy separado |  |
|  |  |  |  |

# Historia

Entre los años 1960 y 1970, el software no era considerado un producto sino un añadido que los vendedores de las grandes computadoras de la época (las mainframes) aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los programadores y desarrolladores de software compartieran libremente sus programas unos con otros. Este comportamiento era particularmente habitual en algunos de los mayores grupos de usuarios de la época, como DECUS (grupo de usuarios de computadoras **DEC**). A finales de la década de 1970, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de licencia.

En 1971, cuando la informática todavía no había sufrido su gran auge, las personas que hacían uso de ella, en ámbitos universitarios y empresariales, creaban y compartían el software sin ningún tipo de restricciones.

Con la llegada de los años 1980 la situación empezó a cambiar. Las computadoras más modernas comenzaban a utilizar sistemas operativos privativos, forzando a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones a dicho software.

En caso de que algún usuario o programador encontrase algún error en la aplicación, lo único que podía hacer era darlo a conocer a la empresa desarrolladora para que ésta lo solucionara. Aunque el programador estuviese capacitado para solucionar el problema y lo desease hacer sin pedir nada a cambio, el contrato le impedía que modificase el software.

En 1971, el estadounidense **Richard Matthew Stallman**, estudiante de Física de la Universidad de Harvard, formó parte como hacker del Laboratorio de Inteligencia Artificial del Instituto Tecnológico de Massachusetts. En los años 1980, la cultura hacker que constituía la vida de Stallman empezó a disolverse bajo la presión de la comercialización en la industria del software. Los demás compañeros de Stallman, también hackers del laboratorio, fundaron la compañía Symbolics que intentaba reemplazar el software libre producido en las instalaciones por software no libre de su propiedad. Stallman contó, años después, que por aquellos años, en el laboratorio donde trabajaba, habían recibido una impresora donada por una empresa externa. El dispositivo, que era utilizado en red por todos los trabajadores, parecía no funcionar a la perfección, dado que cada cierto tiempo el papel se atascaba. Como agravante, no se generaba ningún aviso que se enviase por red e informase a los usuarios de la situación.

La pérdida de tiempo era constante, ya que en ocasiones, los trabajadores enviaban por red sus trabajos a imprimir y al ir a buscarlos se encontraban la impresora atascada y una cola enorme de trabajos pendientes. Richard Stallman decidió arreglar el problema, e implementar el envío de un aviso por red cuando la impresora se bloqueara. Para ello necesitaba tener acceso al código fuente de los controladores de la impresora. Pidió a la empresa propietaria de la impresora lo que necesitaba, comentando, sin pedir nada a cambio, qué era lo que pretendía realizar. La empresa se negó a entregarle el código fuente. En ese preciso instante, Stallman se vio en una encrucijada: debía elegir entre aceptar el nuevo software propietario firmando acuerdos de no revelación y acabar desarrollando más software propietario con licencias restrictivas, que a su vez deberían ser más adelante aceptadas por sus propios colegas. Stallman renunció a su empleo, viviendo de realizar programas libres y vender las copias en cintas y el día 27 de septiembre de 1983, anunció en varios grupos de noticias de Usenet el inicio del **Proyecto GNU**, que perseguía crear un sistema operativo completamente libre.5 En 1985, Stallman fundó la organización Free Software Foundation (FSF) e introdujo la definición formal de software libre y el concepto de "copyleft", que desarrolló para otorgar libertad a los usuarios y para restringir las posibilidades de apropiación del software.

# Las cuatro libertades del software libre

De acuerdo con la definición establecida por Richard Stallman, un software es "libre" cuando garantiza las siguientes libertades:.

1. la libertad de usar el programa, con cualquier propósitoUso)
2. la libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a las propias necesidades (Estudio).
3. la libertad de distribuir copias del programa, con lo cual se puede ayudar a otros usuarios (Distribución).
4. la libertad de mejorar programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie

Las libertades 1 y 3 requieren acceso al código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.

Ciertos teóricos usan este cuarto punto (libertad 3) para justificar parcialmente las limitaciones impuestas por la licencia GNU GPL frente a otras licencias de software libre (ver Licencias GPL).[cita requerida] Sin embargo el sentido original es más libre, abierto y menos restrictivo que el que le otorga la propia situación de incompatibilidad, que ha sido resuelta a cierto grado en versión 3 de la licencia GNU GPL.

Tanto la **Open Source Initiative** como la **Free Software Foundation**, mantienen en sus webs oficiales, listados de las licencias de software libre que aprueban.

El término software no libre se emplea para referirse al software distribuido bajo una licencia de software más restrictiva que no garantiza estas cuatro libertades. Las leyes de la propiedad intelectual reservan la mayoría de los derechos de modificación, duplicación, y redistribución, para el dueño del copyright; el software dispuesto bajo una licencia de software libre rescinde específicamente la mayoría de estos derechos reservados.

La definición de software libre no contempla la cuestión del precio; un eslogan frecuentemente usado es "libre como en libertad, no como en cerveza gratis" o en inglés "Free as in freedom, not as in free beer" (aludiendo a la ambigüedad del término inglés "free"), y es habitual ver a la venta CD de software libre como distribuciones Linux. Sin embargo, en esta situación, el comprador del CD tiene el derecho de copiarlo y redistribuirlo. El software gratis puede incluir restricciones que no se adaptan a la definición de software libre —por ejemplo, puede no incluir el código fuente, puede prohibir explícitamente a los distribuidores recibir una compensación a cambio, etc—.

Para evitar la confusión, algunas personas utilizan los términos "libre" (software libre) y "gratis" (software gratis) para evitar la ambigüedad de la palabra inglesa "free". Sin embargo, estos términos alternativos son usados únicamente dentro del movimiento del software libre, aunque están extendiéndose lentamente hacia el resto del mundo. Otros defienden el uso del término open source software (software de código abierto). La principal diferencia entre los términos "open source" y "free software" es que éste último tiene en cuenta los aspectos éticos y filosóficos de la libertad, mientras que el "open source" se basa únicamente en los aspectos técnicos.

En un intento por unir los mencionados términos que se refieren a conceptos semejantes, se está extendiendo el uso de la palabra "FLOSS" con el significado de free/libre and open source software e, indirectamente, también a la comunidad que lo produce y apoya.

# Formatos abiertos

Los formatos abiertos permiten al software libre mantener sus cuatro libertades y la libre difusión de todo el código y formatos utilizados, su distribución y estudio, debido a esto, los creadores de software libre desarrollan a la vez de programas libres, formatos libres para estos programas o utilizan formatos libres ya creados anteriormente.

Los formatos libres permiten a los usuarios poder trabajar con programas libres aunque al ser libres pueden ser implementados y utilizados en cualquier programa sea cerrado o no. Algunas compañías, como Microsoft, no suelen utilizar formatos libres en sus programas, no por impedimento si no por falta de voluntad de implementar formatos abiertos en sus programas, aún así los usuarios pueden instalar software libre en sus sistemas para trabajar con estos formatos.

# Tipos de licencias

El software no se vende, se licencia. Una licencia es aquella autorización formal con carácter contractual que un autor de un software da a un interesado para ejercer "actos de explotación legales". Es decir, el software no se compra, sino que se adquieren una serie de derechos sobre el uso que se le puede dar. En las licencias de software libre esos derechos son muy abiertos y permisivos, apenas hay restricciones al uso de los programas. De ahí que ayude al desarrollo de la cultura . Pueden existir tantas licencias como acuerdos concretos se den entre el autor y el licenciatario. Desde el punto de vista del software libre, existen distintas variantes del concepto o grupos de licencias:

## Licencias GPL

Una de las más utilizadas es la Licencia Pública General de GNU (GNU GPL). El autor conserva los derechos de autor (copyright), y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL.

Es decir, la licencia **GNU GPL** posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia. Y añade que si se reutiliza en un mismo programa código "A" licenciado bajo licencia GNU GPL y código "B" licenciado bajo otro tipo de licencia libre, el código final "C", independientemente de la cantidad y calidad de cada uno de los códigos "A" y "B", debe estar bajo la licencia GNU GPL.

En la práctica esto hace que las licencias de software libre se dividan en dos grandes grupos, aquellas que pueden ser mezcladas con código licenciado bajo GNU GPL (y que inevitablemente desaparecerán en el proceso, al ser el código resultante licenciado bajo GNU GPL) y las que no lo permiten al incluir mayores u otros requisitos que no contemplan ni admiten la GNU GPL y que por lo tanto no pueden ser enlazadas ni mezcladas con código gobernado por la licencia GNU GPL.

En el sitio web oficial de GNU hay una lista de licencias que cumplen las condiciones impuestas por la GNU GPL y otras que no.

Aproximadamente el 60% del software licenciado como software libre emplea una licencia GPL o de manejo.

## Licencias AGPL

La Licencia Pública General de Affero (en inglés Affero General Public License, también Affero GPL o AGPL) es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red.

La Affero GPL es íntegramente una GNU GPL con una cláusula nueva que añade la obligación de distribuir el software si éste se ejecuta para ofrecer servicios a través de una red de ordenadores.

La Free Software Foundation recomienda que el uso de la GNU AGPLv3 sea considerado para cualquier software que usualmente corra sobre una red.

## Licencias estilo BSD

Llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles. Puede argumentarse que esta licencia asegura “verdadero” software libre, en el sentido que el usuario tiene libertad ilimitada con respecto al software, y que puede decidir incluso redistribuirlo como no libre. Otras opiniones están orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre (normalmente utilizando la siguiente analogía: "una licencia BSD es más libre que una GPL si y sólo si se opina también que un país que permita la esclavitud es más libre que otro que no la permite").

## Licencias estilo MPL y derivadas

Esta licencia es de Software Libre y tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y empezar ese proyecto tan importante para el mundo del Software Libre: Mozilla. Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos. La MPL es Software Libre y promueve eficazmente la colaboración evitando el efecto "viral" de la GPL (si usas código licenciado GPL, tu desarrollo final tiene que estar licenciado GPL). Desde un punto de vista del desarrollador la GPL presenta un inconveniente en este punto, y lamentablemente mucha gente se cierra en banda ante el uso de dicho código. No obstante la MPL no es tan excesivamente permisiva como las licencias tipo BSD. Estas licencias son denominadas de copyleft débil. La NPL (luego la MPL) fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tomados en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

## Copyleft

Hay que hacer constar que el titular de los derechos de autor (copyright) de un software bajo licencia copyleft puede también realizar una versión modificada bajo su copyright original, y venderla bajo cualquier licencia que desee, además de distribuir la versión original como software libre. Esta técnica ha sido usada como un modelo de negocio por una serie de empresas que realizan software libre (por ejemplo MySQL); esta práctica no restringe ninguno de los derechos otorgados a los usuarios de la versión copyleft.

En España, toda obra derivada está tan protegida como una original, siempre que la obra derivada parta de una autorización contractual con el autor. En el caso genérico de que el autor retire las licencias "copyleft", no afectaría de ningún modo a los productos derivados anteriores a esa retirada, ya que no tiene efecto retroactivo. En términos legales, el autor no tiene derecho a retirar el permiso de una licencia en vigencia. Si así sucediera, el conflicto entre las partes se resolvería en un pleito convencional.

# Comparación con el software de código abierto

Este artículo o sección necesita referencias que aparezcan en una publicación acreditada, como revistas especializadas, monografías, prensa diaria o páginas de Internet fidedignas. Este aviso fue puesto el 22 de noviembre de 2010.

Aunque en la práctica el software de código abierto y el software libre comparten muchas de sus licencias, la Free Software Foundation opina que el movimiento del software de código abierto es filosóficamente diferente del movimiento del software libre. Apareció en 1998 con un grupo de personas, entre los que cabe destacar a Eric S. Raymond y Bruce Perens, que formaron la Open Source Initiative (OSI). Ellos buscaban darle mayor relevancia a los beneficios prácticos del compartir el código fuente, e interesar a las principales casas de software y otras empresas de la industria de la alta tecnología en el concepto. Por otro lado, la Free Software Foundation y Richard Stallman prefieren plantear el asunto en términos éticos empleando el término "software libre".

Los defensores del término "código abierto" afirman que éste evita la ambigüedad del término en ese idioma que es free en free software. El término "código abierto" fue acuñado por Christine Peterson del instituto de investigación Foresight Institute, y se registró para actuar como marca registrada el término en inglés, para los productos de software libre.

Mucha gente reconoce el beneficio cualitativo del proceso de desarrollo de software cuando los desarrolladores pueden usar, modificar y redistribuir el código fuente de un programa. El movimiento del software libre hace especial énfasis en los aspectos morales o éticos del software, viendo la excelencia técnica como un producto secundario de su estándar ético. El movimiento de código abierto ve la excelencia técnica como el objetivo prioritario, siendo la compartición del código fuente un medio para dicho fin. Por dicho motivo, la Fundación del Software Libre se distancia tanto del movimiento de código abierto como del término "Código Abierto".

Puesto que la OSI sólo aprueba las licencias que se ajustan a la Open Source Definition (definición de código abierto), la mayoría de la gente lo interpreta como un esquema de distribución, e intercambia libremente los términos "código abierto" y "software libre". Aún cuando existen importantes diferencias filosóficas entre ambos términos, especialmente en términos de las motivaciones para el desarrollo y el uso de tal software, raramente suelen tener impacto en el proceso de colaboración.

Aunque el término "código abierto" elimina la ambigüedad de libertad frente a precio (en el caso del inglés), introduce una nueva: entre los programas que se ajustan a la definición de código abierto, que dan a los usuarios la libertad de mejorarlos, y los programas que simplemente tiene el código fuente disponible, posiblemente con fuertes restricciones sobre el uso de dicho código fuente. Mucha gente cree que cualquier software que tenga el código fuente disponible es de código abierto, puesto que lo pueden manipular (un ejemplo de este tipo de software sería el popular paquete de software gratuito Graphviz, inicialmente no libre pero que incluía el código fuente, aunque luego AT&T le cambió la licencia). Sin embargo, mucho de este software no da a sus usuarios la libertad de distribuir sus modificaciones, restringe el uso comercial, o en general restringe los derechos de los usuarios.

# Implicaciones económico-políticas

Una vez que un producto de software libre ha empezado a circular, rápidamente está disponible a un costo muy bajo. Al mismo tiempo, su utilidad no decrece. El software, en general, podría ser considerado un bien de uso inagotable, tomando en cuenta que su costo marginal es pequeñísimo y que no es un bien sujeto a rivalidad (la posesión del bien por un agente económico no impide que otro lo posea).

Puesto que el software libre permite el libre uso, modificación y redistribución, a menudo encuentra un hogar entre usuarios para los cuales el coste del software no libre es a veces prohibitivo, o como alternativa a la piratería. También es sencillo modificarlo localmente, lo que permite que sean posibles los esfuerzos de traducción a idiomas que no son necesariamente rentables comercialmente.

La mayoría del software libre se produce por equipos internacionales que cooperan a través de la libre asociación. Los equipos están típicamente compuestos por individuos con una amplia variedad de motivaciones, y pueden provenir tanto del sector privado, del sector voluntario o del sector público. Existen muchas posturas acerca de la relación entre el software libre y el actual sistema político-económico:

 Algunos consideran el software libre como un competidor contra el centralismo en empresas y gobiernos, una forma de orden espontáneo o de anarquismo práctico.

 Algunos consideran el software libre como una forma de trabajo colaborativo en un modelo de mercado, tal como se había planteado el cooperativismo.

 Algunos comparan el software libre a una economía del regalo, donde el valor de una persona está basado en lo que ésta da a los demás, sin que incurra valor monetario formal de por medio.

 Grupos como Oekonux e Hipatia consideran que todo debería producirse de esta forma y que este modelo de producción no se limita a reemplazar el modelo no libre de desarrollo del software. La cooperación basada en la libre asociación puede usarse y se usa para otros propósitos (tales como escribir enciclopedias, por ejemplo).

 Hay proyectos de desarrollo con impulso gubernamental que utilizan software libre, así como en proyectos de voluntariado en países en vías de desarrollo.

Las implicaciones políticas y económicas del software libre, o su afinidad con el antiautoritarismo, es discutida. Mientras para unos estas implicaciones son notorias y representan un factor importante a tomarse en cuenta, para otros si bien podría existir una leve relación, no tiene suficiente relevancia.

# Modelo de negocio

El negocio detrás del software libre se caracteriza por la oferta de servicios adicionales al software como: la personalización y/o instalación del mismo, soporte técnico, donaciones, patrocinios o como un elemento de responsabilidad social corporativa; en contraposición al modelo de negocio basado en licencias predominante en el software de código cerrado.

# Seguridad relativa

Existe una cierta controversia sobre la seguridad del software libre frente al software no libre (siendo uno de los mayores asuntos la seguridad por oscuridad). Un método usado de forma habitual para determinar la seguridad relativa de los productos es determinar cuántos fallos de seguridad no resueltos existen en cada uno de los productos involucrados. Por lo general, los usuarios de este método recomiendan que no sean usados productos que no suministren un método de solucionar los fallos de seguridad, al menos hasta que no esté disponible un arreglo.

Existen países en los cuales, sus administraciones públicas, han mostrado apoyo al software libre, sea migrando total o parcialmente sus servidores y sistemas de escritorio, sea subvencionándolo. Este es el caso de Alemania, Brasil, Cuba, Chile, China, Ecuador, España, Francia, México, República Dominicana, y Venezuela.

Además de lo anterior, la Administración Pública tiene cierta función de guía de la industria que la hace tener un gran impacto, que se considera que debe dirigirse a la creación de un ámbito tecnológico generador de riqueza nacional. Ésta puede crearse fomentando empresas, cuyo negocio sea en parte el desarrollo de nuevo software libre para la Administración, el mantenimiento y la adaptación del software existente.

En España en el año 2009, el Centro Nacional de Referencia de Aplicación de las TIC basadas en Fuentes Abiertas (CENATIC), elaboró un informe junto a la Universidad Rey Juan Carlos (Grupo GsyC/LibreSoft) y Telefónica I+D, con el fin de analizar el estado en que se encuentra el proceso de implantación del software de fuentes abiertas en la Administración Pública española.

En México, el Software Libre fue impulsado en las universidades y los centros de investigación. Es por eso que, desde hace tres décadas, los estudiantes y los profesores usan software libre para fines didácticos y de investigación. Ejemplo de ello es la Delta Metropolitana, red de supercomputadoras que están en varios puntos de la Ciudad de México, en el CINESTAV, el IPN, la UAM, y la UNAM la cual utiliza software libre para consolidar sus recursos, hacer investigación, y generar conocimiento.

# Ventajas del Software libre en la educación

Distribución de algunas aplicaciones de software libre en el proceso de educación a distancia.

Según sus impulsores, el software libre ayuda a conocer y participar de forma equitativa el proceso educativo, por lo cual parece tener beneficios en el campo educativo como los siguientes:

* Libertad para elegir el programa que cubra de mejor manera las necesidades educativas.
* Beneficio económico: Las licencias del software libre son completamente gratuitas, por lo que puede ser eliminado el pago en licencia de programas no libres e invertir el dinero en otra área.
* Uso por parte de los estudiantes de los mismos programas en casa y escuela, de forma legal.

# Existencia de comunidades de intercambio educativo.

 Desarrollo independiente de aplicaciones: Se favorece el crecimiento de desarrolladores “pequeños” de software creando así una amplia oferta de programas educativos libres.

 Conocimiento ampliado sobre la informática: El software libre permite que los estudiantes conozcan cómo trabajan los programas y fomenta el interés, la investigación y el aprendizaje de nuevos conceptos en el campo de la informática.

 Diversidad de herramientas educativas en diversas áreas del conocimiento.

# Motivaciones del software libre

 La motivación ética, esgrimida la Free Software Foundation ente que argumenta que el software es conocimiento y debe poderse difundir sin trabas. Su ocultación es una actitud antisocial y la posibilidad de modificar programas es una forma de libertad de expresión, aunque sin olvidar una estructura jerarquizada por la meritocracia

 La motivación pragmática, defendida por la Open Source Initiative, que argumenta ventajas técnicas y económicas, con respecto a evitar una tragedia de los anticomunes mejorando los incentivos.

Aparte de estas motivaciones, quienes trabajan con software libre suelen hacerlo por muchas otras razones, que van desde la diversión a la mera retribución económica, que es posible debido a modelos de negocio sustentables.

## Regulación

### Argentina

En la Argentina, en la Provincia de Río Negro, el 08/03/2012 el Parlamento aprobó la Ley Ley 4747/12 que establece el empleo obligatorio del sistema de Software Libre en los tres Poderes del Estado, entes descentralizados y empresas con participación estatal.

En la Provincia de Santa Fe, fue publicada la Ley No. 13139 en el 2010 como una actualización de la Ley No. 12360 del año 2004, la primera de este tipo en el país. El espítiru de la Ley 13139 radica en la implementación de políticas de incorporación de software que garanticen soberanía, seguridad de los datos.

### Bolivia

El presidente Evo Morales Ayma, el día lunes 8 de agosto del 2011, reglamentó la Ley Nº 164 de Telecomunicaciones y TIC's para el Desarrollo de Tecnologías de Información y Comunicación.

### Ecuador

El Presidente Rafael Correa Delgado, el día jueves 10 de abril del 2008, firmó el Decreto N° 1014; en el cual ordena, que el software usado por las administraciones públicas del país sea software libre e implícitamente basado en estándares abiertos.

### España

La Orden EDU/2341/2009, de 27 de agosto, por la que se crea el Centro Nacional de Desarrollo Curricular en Sistemas no Propietarios, tiene como finalidad el diseño, el desarrollo y la promoción de contenidos educativos digitales para colectivos educativos específicos, en el ámbito de las Tecnologías de la Información y la Comunicación, que se centra en promocionar y aplicar estrategias dirigidas a poner a disposición de los centros escolares recursos y contenidos digitales de calidad, desarrollados en software libre.

### Uruguay

En Uruguay, a partir de 2003 (Comisión de Constitución, Códigos, Legislación General y Administración Carpeta Nº 3565 de 2003 Repartido Nº 1510 de noviembre de 2003) se estudió una "Ley de Software Libre y Formatos Abiertos en el Estado". La primera versión planteaba el uso de formatos abiertos en todo el Estado y Software Libre en la educación, con una segunda versión presentada en el 2006, que ya planteaba dar preferencia al uso de Software Libre en todos los organismos del Estado. Luego dicha versión con modificaciones, fue la que recibió media sanción en la Cámara de Diputados el 19 de diciembre de 2012, la cual fue apoyada por la comunidad de software libre uruguaya. Finalmente, la ley fue aprobada en diciembre de 2013 estableciéndose en ella que:

El Estado deberá preferir la inversión y desarrollo en software libre sobre el que no lo es, salvo cuando éste no cumpla las necesidades técnicas requeridas.

* En caso de que el Estado decida invertir en software privativo, deberá justificar las razones del gasto y argumentar su elección.
* El Estado deberá distribuir y aceptar toda información en al menos un formato abierto, estándar y libre.
* El intercambio de información a través de Internet deberá ser posible en al menos un programa con licencia libre.

### Venezuela

El Decreto presidencial 3390 de fecha 23 de diciembre de 2004 y publicado en la Gaceta Oficial de Venezuela Nº 38095 del 28 de diciembre de 2004, establece textualmente en su artículo 1 que los entes públicos emplearán prioritariamente Software Libre, en sus sistemas, proyectos y servicios informáticos, por lo que dichas organizaciones deben iniciar procesos de migración gradual y progresiva de éstos hacia el Software Libre