**Software libre**

es la denominación del [**software**](http://es.wikipedia.org/wiki/Software) que respeta la [**libertad**](http://es.wikipedia.org/wiki/Libertad) de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente. Según la [***Free*****Software****Foundation**](http://es.wikipedia.org/wiki/Free_Software_Foundation), el software libre se refiere a la [**libertad**](http://es.wikipedia.org/wiki/Libertad) de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el [**software**](http://es.wikipedia.org/wiki/Software)**;** de modo más preciso, se refiere a [**cuatro libertades de los usuarios del software**](http://es.wikipedia.org/wiki/Software_libre#Libertades_del_software_libre)**:** la libertad de usar el programa, con cualquier propósito; de estudiar el funcionamiento del programa, y adaptarlo a las necesidades; de distribuir copias, con lo cual se puede ayudar a otros y de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (para la segunda y última libertad mencionadas, el acceso al [**código fuente**](http://es.wikipedia.org/wiki/C%C3%B3digo_fuente) es un requisito previo). [

el

el software libre suele estar disponible gratuitamente, o al precio de costo 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" , ya que, conservando su carácter de libre, puede ser distribuido [**comercialmente**](http://es.wikipedia.org/wiki/Software_comercial) ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el [**código fuente**](http://es.wikipedia.org/wiki/C%C3%B3digo_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**](http://es.wikipedia.org/wiki/Dominio_p%C3%BAblico)". Éste último es aquel software que no requiere de [**licencia**](http://es.wikipedia.org/wiki/Licencia_de_software), pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquel cuyo autor lo dona a la humanidad o cuyos [derechos de autor](http://es.wikipedia.org/wiki/Derechos_de_autor) han expirado, tras un plazo contado desde la muerte de este, habitualmente 70 años. Si un [autor](http://es.wikipedia.org/wiki/Autor) condiciona su uso bajo una licencia, por muy débil que sea, ya no es del dominio público.

Entre los [años 60](http://es.wikipedia.org/wiki/A%C3%B1os_1960) y [70](http://es.wikipedia.org/wiki/A%C3%B1os_1970) del [Siglo XX](http://es.wikipedia.org/wiki/Siglo_XX), el software no era considerado un producto sino un añadido que los vendedores de las grandes [computadoras](http://es.wikipedia.org/wiki/Computadora) de la época (las [mainframes](http://es.wikipedia.org/wiki/Mainframe)) aportaban a sus clientes para que éstos pudieran usarlos. En dicha cultura, era común que los [programadores](http://es.wikipedia.org/wiki/Programador) y desarrolladores de software compartieran libremente sus [programas](http://es.wikipedia.org/wiki/Programa_(computaci%C3%B3n)) unos con otros. Este comportamiento era particularmente habitual en algunos de los mayores grupos de usuarios de la época, como [**DECUS**](http://es.wikipedia.org/wiki/DECUS) (grupo de usuarios de computadoras DEC). A finales de los [70](http://es.wikipedia.org/wiki/1970), las compañías iniciaron el hábito de imponer restricciones a los usuarios, con el uso de acuerdos de [licencia](http://es.wikipedia.org/wiki/Licencia_de_software).

Allá por el [1971](http://es.wikipedia.org/wiki/1971), cuando la [informática](http://es.wikipedia.org/wiki/Inform%C3%A1tica) todavía no había sufrido su gran boom, las personas que hacían uso de ella, en ámbitos universitarios y empresariales, creaban y *compartían el* [*software*](http://es.wikipedia.org/wiki/Software) sin ningún tipo de restricciones.

Con la llegada de los años [80](http://es.wikipedia.org/wiki/1980) la situación empezó a cambiar. Las [computadoras](http://es.wikipedia.org/wiki/Computadora) más modernas comenzaban a utilizar [sistemas operativos](http://es.wikipedia.org/wiki/Sistemas_operativos) [privativos](http://es.wikipedia.org/wiki/Software_privativo), forzando a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones a dicho [software](http://es.wikipedia.org/wiki/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 esta 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 mejorase el [software](http://es.wikipedia.org/wiki/Software).

El mismo [**Richard Stallman**](http://es.wikipedia.org/wiki/Richard_Stallman) cuenta que por aquellos años, en el laboratorio donde trabajaba, habían recibido una [impresora](http://es.wikipedia.org/wiki/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](http://es.wikipedia.org/wiki/Richard_Stallman) decidió arreglar el problema, e implementar el envío de un aviso por red cuando la [impresora](http://es.wikipedia.org/wiki/Impresora) se bloqueara. Para ello necesitaba tener acceso al [código fuente](http://es.wikipedia.org/wiki/C%C3%B3digo_fuente) de los controladores de la [impresora](http://es.wikipedia.org/wiki/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](http://es.wikipedia.org/wiki/C%C3%B3digo_fuente).

En ese preciso instante, [Richard Stallman](http://es.wikipedia.org/wiki/Richard_Stallman) se vio en una encrucijada: debía elegir entre aceptar el nuevo [**software privativo**](http://es.wikipedia.org/wiki/Software_privativo) firmando acuerdos de no revelación y acabar desarrollando más [software privativo](http://es.wikipedia.org/wiki/Software_privativo) con licencias restrictivas, que a su vez deberían ser más adelante aceptadas por sus propios colegas.

Con este antecedente, en [1984](http://es.wikipedia.org/wiki/1984), [Richard Stallman](http://es.wikipedia.org/wiki/Richard_Stallman) comenzó a trabajar en el proyecto [**GNU**](http://es.wikipedia.org/wiki/GNU), y un año más tarde fundó la [Free Software Foundacion](http://es.wikipedia.org/wiki/Free_Software_Foundation). [Stallman](http://es.wikipedia.org/wiki/Richard_Stallman) introdujo la definición de free software y el concepto de "[copyleft](http://es.wikipedia.org/wiki/Copyleft)", que desarrolló para otorgar libertad a los usuarios y para restringir las posibilidades de apropiación del [software](http://es.wikipedia.org/wiki/Software).[]

**Libertades del software libre**

|  |  |
| --- | --- |
| **Libertad** | **Descripción** |
| **0** | La libertad de usar el programa, con cualquier propósito. |
| **1** | La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades. |
| **2** | La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo. |
| **3** | La libertad de mejorar el 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**](http://es.wikipedia.org/wiki/C%C3%B3digo_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](http://es.wikipedia.org/wiki/Licencia_p%C3%BAblica_general_de_GNU) frente a otras licencias de software libre (ver [Licencias GPL](http://es.wikipedia.org/wiki/Software_libre#Licencias_GPL)). Sin embargo el sentido original es más libre, abierto y menos restrictivo que el que le otorga la propia situación de incompatibilidad, que podría ser resuelta en la próxima versión 3.0 de la licencia [GNU GPL](http://es.wikipedia.org/wiki/Licencia_p%C3%BAblica_general_de_GNU), causa en estos momentos graves perjuicios a la comunidad de [programadores](http://es.wikipedia.org/wiki/Programador) de software libre, que muchas veces no pueden reutilizar o mezclar códigos de dos licencias distintas, pese a que las libertades teóricamente lo deberían permitir.

En el sitio web oficial de [Open Source Initiative](http://es.wikipedia.org/wiki/Open_Source_Initiative) está la lista completa de las [licencias de software libre](http://es.wikipedia.org/wiki/Licencias_de_software_libre) actualmente aprobadas y tenidas como tales.

El término [software no libre](http://es.wikipedia.org/wiki/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](http://es.wikipedia.org/wiki/Propiedad_intelectual) reservan la mayoría de los derechos de modificación, duplicación y redistribución para el dueño del [copyright](http://es.wikipedia.org/wiki/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 el asunto del precio; un eslogan frecuentemente usado es o en inglés es habitual ver a la venta [CD](http://es.wikipedia.org/wiki/Disco_compacto) de software libre como [distribuciones Linux](http://es.wikipedia.org/wiki/Distribuci%C3%B3n_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.

Para evitar la confusión, algunas personas utilizan los términos "libre” y "gratis" para evitar la ambigüedad de la palabra inglesa "free". Sin embargo, estos términos alternativos son usados únicamente dentro del [movimiento del software libre](http://es.wikipedia.org/wiki/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](http://es.wikipedia.org/wiki/Software_libre_y_de_c%C3%B3digo_abierto)" con el significado de free/libre and open source software e, indirectamente, también a la comunidad que lo produce y apoya.

**Tipos de licencias**

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". 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](http://es.wikipedia.org/wiki/GNU_General_Public_License)). 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](http://es.wikipedia.org/wiki/Licencia_p%C3%BAblica_general_de_GNU). 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.[[4]](http://es.wikipedia.org/wiki/Software_libre#cite_note-3)

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

**Licencias estilo BSD**

Llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos [**BSD**](http://es.wikipedia.org/wiki/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**](http://es.wikipedia.org/wiki/Licencia_p%C3%BAblica_general_de_GNU) 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 Comunicaciones Corp. para liberar su Netscape Comunicador 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 tenidos 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](http://es.wikipedia.org/wiki/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**](http://es.wikipedia.org/wiki/MySQL); esta práctica *no* restringe ninguno de los derechos otorgados a los usuarios de la versión [copyleft](http://es.wikipedia.org/wiki/Copyleft). También podría retirar todas las licencias de software libre anteriormente otorgadas, pero esto obligaría a una indemnización a los titulares de las licencias en uso. 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**

|  |  |
| --- | --- |
|  |  |

en la práctica el software de [código abierto](http://es.wikipedia.org/wiki/Software_de_c%C3%B3digo_abierto) y el software libre comparten muchas de sus licencias, la FSF opina que el movimiento de [código abierto](http://es.wikipedia.org/wiki/C%C3%B3digo_abierto) es filosóficamente diferente del movimiento del software libre. Apareció en [1998](http://es.wikipedia.org/wiki/1998) con un grupo de personas, entre los que cabe destacar a [Eric S. Raymond](http://es.wikipedia.org/wiki/Eric_S._Raymond) y [Bruce Perens](http://es.wikipedia.org/wiki/Bruce_Perens), que formaron la [Open Source Initiative](http://es.wikipedia.org/wiki/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. Mientras que la [FSF](http://es.wikipedia.org/wiki/FSF) y [Richard Stallman](http://es.wikipedia.org/wiki/Richard_Stallman), prefieren plantear el asunto en términos éticos empleando el término Software Libre.

Estos defensores ven que el término "código abierto". El término "código abierto" fue acuñado por [Christine Peterson](http://es.wikipedia.org/w/index.php?title=Christine_Peterson&action=edit&redlink=1) del [Foresight Institute](http://es.wikipedia.org/w/index.php?title=Foresight_Institute&action=edit&redlink=1), y se registró para actuar como [marca registrada](http://es.wikipedia.org/wiki/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 deseable 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 FSF se distancia tanto del movimiento de código abierto como del término "Código Abierto".

Puesto que la OSI sólo aprueba las licencias que se ajustan, la mayoría de la gente lo interpreta como un esquema de distribución, e intercambia libremente "código abierto" con "software libre". Aun 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, 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 códigoabierto, puesto que lo pueden manipular (un ejemplo de este tipo de software sería el popular paquete de software gratuito [Graphviz](http://es.wikipedia.org/w/index.php?title=Graphviz&action=edit&redlink=1), inicialmente no libre pero que incluía el código fuente, aunque luego [AT&T](http://es.wikipedia.org/wiki/AT%26T) 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 del usuario.