

# Licencias en el Software Libre

Jhon James Quintero Osorio<sup>1</sup>

<sup>1</sup>Universidad del Quindío, Programa de Ingeniería Electrónica

Flisol Armenia 2010.

# Contenido

- 1 Introducción
  - El software libre
- 2 Términos relacionados
- 3 El copyleft
- 4 Licencias en el Software Libre
  - Definición de licencia
  - Historia
  - Licencias en el Software Libre

# Contenido

- 1 **Introducción**
  - El software libre
- 2 Términos relacionados
- 3 El copyleft
- 4 Licencias en el Software Libre
  - Definición de licencia
  - Historia
  - Licencias en el Software Libre

# Contenido

- 1 **Introducción**
  - El software libre
- 2 Términos relacionados
- 3 El copyleft
- 4 Licencias en el Software Libre
  - Definición de licencia
  - Historia
  - Licencias en el Software Libre



Para mucha gente, *free software* o *freeware* significa lo mismo que *software libre* (*libre software*), esto es debido a que en inglés existe una ambigüedad peligrosa en la palabra *free*, significa tanto “libre” como “gratis”.

De una vez.

*¡El software libre, no tiene por que ser gratis!*



Para mucha gente, *free software* o *freeware* significa lo mismo que *software libre* (*libre software*), esto es debido a que en inglés existe una ambigüedad peligrosa en la palabra *free*, significa tanto “libre” como “gratis”.

De una vez.

*¡El software libre, no tiene por que ser gratis!*



Para mucha gente, *free software* o *freeware* significa lo mismo que *software libre* (*libre software*), esto es debido a que en inglés existe una ambigüedad peligrosa en la palabra *free*, significa tanto “libre” como “gratis”.

De una vez.

*¡El software libre, no tiene por que ser gratis!*

# Características del Software Libre

El "software libre" es una cuestión de libertad, no de precio. Para entender el concepto, se debería pensar en "libre" como en "libre expresión", no como en "barra libre".

Los usuarios de dicho tipo de software tienen cuatro libertades esenciales:

- Usar el software como deseen, para lo que deseen, y en tantos computadores como deseen, y en cualquier situación técnicamente apropiada (libertad 0).
- La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.



# Características del Software Libre

El "software libre" es una cuestión de libertad, no de precio. Para entender el concepto, se debería pensar en "libre" como en "libre expresión", no como en "barra libre".

Los usuarios de dicho tipo de software tienen cuatro libertades esenciales:

- Usar el software como deseen, para lo que deseen, y en tantos computadores como deseen, y en cualquier situación técnicamente apropiada (libertad 0).
- La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.

# Características del Software Libre

El "software libre" es una cuestión de libertad, no de precio. Para entender el concepto, se debería pensar en "libre" como en "libre expresión", no como en "barra libre".

Los usuarios de dicho tipo de software tienen cuatro libertades esenciales:

- Usar el software como deseen, para lo que deseen, y en tantos computadores como deseen, y en cualquier situación técnicamente apropiada (libertad 0).
- La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.

# Características del Software Libre

- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (la 3ª libertad). 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.

Un programa es software libre si se cumplen las cuatro libertades mencionadas. Entonces, debería ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución, a cualquiera en cualquier parte. El ser libre de hacer estas cosas significa, entre otras cosas, que no tiene que pedir o pagar el permiso.

# Características del Software Libre

- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (la 3ª libertad). 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.

Un programa es software libre si se cumplen las cuatro libertades mencionadas. Entonces, debería ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución, a cualquiera en cualquier parte. El ser libre de hacer estas cosas significa, entre otras cosas, que no tiene que pedir o pagar el permiso.

# Características del Software Libre

- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (la 3ª libertad). 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.

Un programa es software libre si se cumplen las cuatro libertades mencionadas. Entonces, debería ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución, a cualquiera en cualquier parte. El ser libre de hacer estas cosas significa, entre otras cosas, que no tiene que pedir o pagar el permiso.

# Características del Software Libre

También debería tener la *libertad de hacer modificaciones y usarlas en privado*, en su propio trabajo u obra, *sin siquiera mencionar que existen*. Si publica sus cambios, no debería estar obligado a notificarlo a alguien en particular, o de alguna forma en particular.

# Características del Software Libre

La libertad de ejecutar el programa significa la libertad para cualquier tipo de persona u organización de usarlo en cualquier tipo de sistema de computación, *para cualquier tipo de trabajo y propósito*, sin estar obligado a comunicarlo a su programador, o alguna otra entidad específica. En esta libertad, *el propósito de los usuarios es el que importa*, no el propósito de los programadores.

# Características del Software Libre

Como el usuario es libre de ejecutar un programa para sus propósitos; y si lo distribuye a otra persona, ésta también es libre para ejecutarlo para sus propósitos, el programador no tiene derecho a imponerle sus propios propósitos.



# Características del Software Libre

"Software libre" no significa "que no sea comercial". Un programa libre debe estar disponible para el uso comercial, la programación comercial y la distribución comercial.

Puede haber pagado dinero para obtener copias de software libre, o puede haber obtenido copias sin costo. Pero sin tener en cuenta cómo obtuvo sus copias, siempre tiene la libertad de copiar y modificar el software, incluso de *vender copias*.

# Características del Software Libre

Cuando se habla de software libre, es mejor evitar usar términos como "regalar" o "gratis", porque dichos términos implican que el asunto pasa por el precio, no la libertad.

# Contenido

- 1 Introducción
  - El software libre
- 2 Términos relacionados
- 3 El copyleft
- 4 Licencias en el Software Libre
  - Definición de licencia
  - Historia
  - Licencias en el Software Libre

# El OpenSource

- Equivalente a software libre es el término Open Source Software ('programas de fuente abierto'), promovido por Eric Raymond y la Open Source Initiative.
- Filosóficamente, el término es muy distinto, ya que hace énfasis en la disponibilidad de código fuente, no en la libertad.
- Fuertemente criticado por Richard Stallman y la Free Software Foundation, ha encontrado mucho más eco en la literatura comercial y en las estrategias de las empresas que de una manera u otra apoyan el modelo.

# Freeware

- Programas gratuitos. Normalmente se ceden en binario y con derechos de redistribución.
- Sin embargo, a veces sólo se pueden obtener de un sitio oficial, normalmente para promocionar otros programas o servicios.

# Shareware

- No es siquiera software gratis, sino un método de distribución, ya que los programas, generalmente sin fuentes, se pueden copiar libremente, pero no usar continuamente sin pagarlos.
- La exigencia de pago puede estar incentivada por funcionalidad limitada o mensajes molestos, o una simple apelación a la moral del usuario.
- Las estipulaciones legales de la licencia podrían utilizarse en contra del infractor.

# Charityware, Careware

- Generalmente shareware, pero cuyo pago se exige para una organización caritativa patrocinada.
- En muchos casos, el pago no se exige, pero se solicita una contribución voluntaria.
- Algún software libre, como vim solicita contribuciones voluntarias de este tipo.

# Dominio público

- El autor renuncia absolutamente a todos sus derechos, en favor del común, lo cual tiene que estar declarado explícitamente en el programa, ya que si no se dice nada, el programa es propietario y no se puede hacer nada con él.
- Si además se proporcionan los códigos fuentes, el programa es libre.



# Propietario, cerrado, no libre

- Términos usados para denominar al software que no es libre ni de fuente abierta.

# Contenido

- 1 Introducción
  - El software libre
- 2 Términos relacionados
- 3 El copyleft**
- 4 Licencias en el Software Libre
  - Definición de licencia
  - Historia
  - Licencias en el Software Libre

# El copyleft

- El copyleft es un método general para hacer un programa (u otro tipo de trabajo) libre, exigiendo que todas las versiones modificadas y extendidas del mismo sean también libres.
- La forma más simple de hacer que un programa sea libre es ponerlo bajo dominio público, sin derechos de autor.

# El copyleft

- Con copyleft cualquiera que redistribuya el software, con o sin cambios, deberá de otorgar al usuario la libertad de copiarlo y modificarlo, garantizando que se mantendrán estas libertades para todos los usuarios.
- Cuándo se publica una versión modificada, esta deberá estar exactamente bajo la misma licencia.
- El copyleft es un concepto general y, por lo tanto, no puede usarse de forma directa; solamente es posible utilizar una implementación específica del concepto.

# El copyleft

El copyleft se implementa por ejemplo en:

- Licencia Pública GNU
- Licencia Pública General Reducida de GNU (LGPL)
- La Licencia de Documentación Libre de GNU (FDL)

# Contenido

- 1 Introducción
  - El software libre
- 2 Términos relacionados
- 3 El copyleft
- 4 Licencias en el Software Libre**
  - Definición de licencia
  - Historia
  - Licencias en el Software Libre

# Contenido

- 1 Introducción
  - El software libre
- 2 Términos relacionados
- 3 El copyleft
- 4 **Licencias en el Software Libre**
  - **Definición de licencia**
  - Historia
  - Licencias en el Software Libre

# Definición de licencia

- Defiende exactamente los derechos que los usuarios tienen sobre él
- En la mayoría de los programas propietarios, la licencia limita (o anula) los derechos de copia, modificación, donación, alquiler, uso en varias máquinas, etc.
- De hecho en forma general las licencias especifican que el propietario del software es la compañía que los publica, que solamente vende derechos limitados de uso.



# Definición de licencia

- Defiende exactamente los derechos que los usuarios tienen sobre él
- En la mayoría de los programas propietarios, la licencia limita (o anula) los derechos de copia, modificación, donación, alquiler, uso en varias máquinas, etc.
- De hecho en forma general las licencias especifican que el propietario del software es la compañía que los publica, que solamente vende derechos limitados de uso.

# Definición de licencia

- Defiende exactamente los derechos que los usuarios tienen sobre él
- En la mayoría de los programas propietarios, la licencia limita (o anula) los derechos de copia, modificación, donación, alquiler, uso en varias máquinas, etc.
- De hecho en forma general las licencias especifican que el propietario del software es la compañía que los publica, que solamente vende derechos limitados de uso.

# Contenido

- 1 Introducción
  - El software libre
- 2 Términos relacionados
- 3 El copyleft
- 4 Licencias en el Software Libre**
  - Definición de licencia
  - Historia**
  - Licencias en el Software Libre

# Historia

- Durante los años 1960 el panorama de la informática estaba dominado por los grandes ordenadores, instalados fundamentalmente en empresas y centros gubernamentales.
- IBM era el principal fabricante, con gran diferencia sobre sus competidores. En esta época, cuando se adquiría un ordenador (el hardware), el software venía como un acompañante.
- Mientras se pagase el contrato de mantenimiento, se tenía acceso al catálogo de software que ofrecía el fabricante.

# Historia

En palabras de Richard Stallman:

- “No denominábamos software libre a nuestro software porque este término no existía, pero eso es lo que era. Cuando alguien de otra universidad o de una empresa deseaba portar y usar un programa, se lo permitíamos con gusto. Si veías a alguien usando un programa interesante y poco conocido, siempre podías pedir el código fuente para verlo, de manera que podías leerlo, cambiarlo, o canibalizar ciertas partes del mismo para hacer un nuevo programa.”

# Historia

- El 30 de junio de 1969 IBM anunció que a comienzos de 1970 iba a empezar a vender parte de su software por separado.
- Esto supuso que sus clientes ya no podían obtener, incluido en el precio del hardware, los programas que necesitaban.
- El software se comenzó a percibir como algo con valor intrínseco y, como consecuencia, se hizo cada vez más habitual restringir escrupulosamente el acceso a los programas.

# Historia

- A mediados de la década de 1970 era ya absolutamente habitual, en cualquier ámbito informático, encontrarse con software propietario.
- En los 70's y primeros 80's había iniciativas que mostraban algunas características de lo que luego se consideraría software libre.
- Entre ellas, caben destacar Spice y T<sub>E</sub>X, además del caso mucho más complejo de Unix.

# Historia

- Durante el final de la década de 1970, y sobre todo durante la de 1980, AT&T cambió su política, y el acceso a nuevas versiones de Unix se convirtió en algo difícil y caro.
- La filosofía de los primeros años, que hizo tan popular a Unix entre los desarrolladores, cambió radicalmente, hasta el punto de que en 1991 AT&T puso una demanda a la Universidad de Berkeley por publicar el código de Unix BSD que ellos (el CSRG de Berkeley) habían creado.



# Historia

- A principios de 1984, Richard Stallman, en aquella época empleado en el AI Lab del MIT, abandonó su trabajo para comenzar el proyecto GNU.
- Su idea al abandonar el MIT era construir un sistema de software completo, de propósito general, pero completamente libre.
- El sistema (y el proyecto que se encargaría de hacerlo realidad) se llamó GNU (acrónimo recursivo, GNU's Not Unix).

# Historia

- Richard Stallman comenzó por escribir un compilador de C (GCC) y un editor (Emacs), ambos aún en uso (y muy populares) hoy día.
- Desde el principio del proyecto GNU, Richard Stallman estaba preocupado por las libertades que tendrían los usuarios de su software.
- Para ello, escribió la licencia GPL, probablemente la primera licencia de software diseñada específicamente para garantizar que un programa fuera libre.

# Historia

- Richard Stallman también fundó la Free Software Foundation (FSF) con el fin de conseguir fondos para el desarrollo y la protección del software libre.
- El CSRG (Computer Science Research Group) de la Universidad de California en Berkeley fue, desde 1973, uno de los centros donde más se desarrolló todo lo relacionado con Unix.
- Para poder utilizar todo el código que producía el CSRG, hacía falta la licencia de Unix de AT&T, que cada vez era más difícil (y más cara) de conseguir, sobre todo si se quería el acceso al código fuente del sistema.

# Historia

- Tratando de evitar en parte este problema, en junio de 1989 el CSRG liberó la parte de Unix relacionada con TCP/IP.
- Fue la llamada Networking Release 1 (Net-1).
- En junio de 1991, y después de conseguir el permiso de la Administración de la Universidad de Berkeley, se distribuyó la Networking Release 2 (Net-2), con casi todo el código del kernel y todas las utilidades de un sistema Unix completo.

# Historia

- Sólo seis meses después de la liberación de Net-2, Bill Jolitz escribió el código que faltaba en el kernel para que funcionase sobre arquitectura i386, liberando 386BSD.
- Sucesión de las familias \*BSD.
- Varios años más tarde se formó el proyecto OpenBSD, con énfasis en la seguridad.

# Historia

- En julio de 1991 Linus Torvalds (estudiante finés de 21 años) pone el primer mensaje donde menciona su (por entonces) proyecto de hacer un sistema libre similar a Minix.
- En marzo de 1994 apareció la versión 1.0, la primera que fue denominada estable.
- Durante este periodo, literalmente cientos de desarrolladores se vuelcan sobre Linux, integrando a su alrededor todo el software de GNU, XFree, y muchos otros programas libres.
- A diferencia de los \*BSD, Linux (el kernel) y gran parte de los componentes que se integran alrededor de él se distribuyen con la licencia GPL.

# Contenido

- 1 Introducción
  - El software libre
- 2 Términos relacionados
- 3 El copyleft
- 4 **Licencias en el Software Libre**
  - Definición de licencia
  - Historia
  - **Licencias en el Software Libre**

# Licencias en el Software Libre

Las condiciones que se especifican en las licencias del software libre son el resultado de un compromiso entre varios objetivos que son en parte contradictorios:

- Garantizar las libertades básicas.
- Asegurar algunas condiciones impuestas por los autores (por ejemplo, cita del autor en trabajos derivados).
- Garantizar que los trabajos derivados sean también software libre.



# Licencias en el Software Libre

- Los autores pueden escoger proteger su software con diferentes licencias de acuerdo al grado con el que deseen cumplir sus objetivos y los detalles que quieran asegurar.
- El autor puede, si lo así lo desea, distribuir su software con diferentes licencias a través de diferentes canales, y a distintos precios. Ejemplo conocidos son, las librerías Qt y MySQL

# Licencias en el Software Libre

Aunque existen muchos tipos de licencias libre que protegen las cuatro libertades mencionadas, se pueden clasificar en dos matices:

- **Licencias Robustas:** exigen que los trabajos derivados se redistribuyan con la misma licencia, sin añadir ninguna restricción adicional. Son conocidas como licencias copyleft. Ejemplos de licencias robustas: GPL: GNU General Public License, LGPL: Lesser GPL, MPL: Mozilla Public License.
- **Licencias Permisivas:** permiten la creación de trabajos derivados cambiando las condiciones originales de la licencia. Esto, por ejemplo, permite crear a partir de un código software libre un producto que no lo sea. Ejemplos de licencias permisivas: BSD: Berkeley Software Distribution, X11/MIT License, Apache.

# La Licencia Pública General de GNU (GNU GPL)

- Es con diferencia la licencia más popular y conocida de todas las licencias del mundo del software libre creada por la FSF.
- Permite la redistribución binaria y la de las fuentes, aunque, en el caso de que redistribuya de manera binaria, obliga a que también se pueda acceder a las fuentes.
- Está permitido realizar modificaciones sin restricciones, aunque sólo se pueda integrar código licenciado bajo GPL con otro código que se encuentre bajo una licencia idéntica o compatible, lo que ha venido a llamarse el *efecto viral* de la GPL, ya que el código publicado una vez con esas condiciones nunca puede cambiar de condiciones.

# La Licencia Pública General de GNU (GNU GPL)

- Los partidarios de las licencias tipo BSD ven en esta cláusula un recorte de la libertad, mientras que sus seguidores ven en ello una forma de asegurarse que ese software siempre va a ser libre.
- Por otro lado, se puede considerar que la licencia GPL maximiza las libertades de los usuarios, mientras que las de tipo BSD lo hacen para los desarrolladores.
- La GPL contempla también a las patentes de software, exigiendo que si el código lleva algoritmos patentados o se concede licencia de uso de la patente libre de tasas, o no se puede distribuir bajo la GPL.

# La Licencia Pública General Menor de GNU (GNU LGPL)

- Pensada en sus inicios para su uso en bibliotecas (la L en sus comienzos venía de library, 'biblioteca'), fue modificada recientemente para ser considerada la hermana menor (lesser, 'menor') de la GPL.
- La LGPL permite el uso de programas libres con software propietario.

# Licencia de Apache

- Licencia bajo la que se distribuyen la mayor parte de los programas producidos por el proyecto Apache.
- Hay algunos programas libres que no se distribuyen con una licencia específica, sino que su autor los declara explícitamente *public domain*, el autor renuncia a todos sus derechos sobre el programa, y por lo tanto puede modificarse, redistribuirse, usarse, etc. de cualquier manera, a efectos prácticos, esta situación es muy similar a que el programa esté bajo una licencia tipo BSD.

# Affero General Public License

- Igual que la GPL, pero teniendo en cuenta el caso de "Software as a Service".
- La licencia GPL no obliga a compartir el código hasta que éste sea distribuido con alguien.
- ¿Pero qué ocurre cuando se usan servicios externos que corren en otros servidores? La Affero obliga a éstos servicios a dar la posibilidad de descargar el código desde la web.

# Compatibilidad

- Suponga que tenemos dos o tres programas y cada uno tiene su propia licencia y usted quiere combinarlos o enlazarlos. ¿Puede hacerlo? Bien, cada una de estas licencias puede incluir condiciones para el licenciamiento de combinaciones. Entonces, la pregunta sería: ¿Existe algún modo de respetar los términos de ambas licencias a la vez? Si lo hay, las licencias son compatibles, y se puede hacer la combinación porque existe una forma de licenciar la combinación que satisfaga ambas licencias.



# Compatibilidad

- Pero si no hay manera de licenciar la combinación de modo que se cumplan los requisitos de ambas, la combinación no podrá hacerse. Eso significa que las licencias son incompatibles.
- En general dos diferentes licencias con copyleft, son generalmente incompatibles.

# Resumen

- El Software libre se distribuye sobre muchas licencias que pueden ser incompatibles entre sí.
- Es importante informarse sobre los tipos de licencia para publicar desarrollo.
- Todavía falta un estudio serio sobre las licencias de software libre y compatibilidad con la legislación Colombiana.