## Actividad 1

**1. Ofrece varios ejemplos de licencias para la clasificación según los derechos reservados por el autor del software.**

Licencia de software de código abierto permisivas

Se puede crear una obra derivada sin que ésta tenga obligación de protección alguna. Muchas licencias pertenecen a esta clase, entre otras:

Academic Free License v.1.2.

Apache Software License v.1.1

Artistic License v.2.0

Attribution Assurance license.

BSD License.

MIT License.

University of Illinois/NCSA Open Source License.

W3C Software Notice and License.

Zope Public License v.2.0

Open LDAP License v.2.7

Perl License.

Academic Free License v.3.0

Python License v.2.1

PHP License v.3.0

VMS License.

Licencia de software de código abierto robustas

Estas licencias aplican algunas restricciones a las obras derivadas, haciendo que según el grado de aplicación se puedan dividir a su vez en dos subcategorías:

Licencias de software de código abierto robustas fuertes

Las licencias de software de código abierto robustas fuertes o con copyleft fuerte, contienen una cláusula que obliga a que las obras derivadas o modificaciones que se realicen al software original se deban licenciar bajo los mismos términos y condiciones de la licencia original.

Entre las licencias de esta categoría están:

Common Public License v.1.0.

GNU General Public License v.2.0.

GNU General Public License v.3.0.

Eclipse Public License.

eCos License v.2.0

Sleepycat Software Product License.

Affero License v.1.0

Affero License v.2.0

OpenSSL License.

Licencias de software de código abierto robustas débiles

Las licencias de software de código abierto robustas débiles, con copyleft débil/suave o híbridas, contienen una cláusula que obliga a que las modificaciones que se realicen al software original se deban licenciar bajo los mismos términos y condiciones de la licencia original, pero que las obras derivadas que se puedan realizar de él puedan ser licenciadas bajo otros términos y condiciones distintas.

Entre las licencias de esta categoría están:

| | |
|---|---|
| GNU Lesser General Public License v.2.1. | Apple Source License v.2.0 |
| Mozilla Public License | CDDL. |
| Open Source License. | EUPL. |

Licencia de software de código cerrado

Estas licencias también se conocen con el nombre de «software propietario» o privativo. En ellas los propietarios establecen los derechos de uso, distribución, redistribución, copia, modificación, cesión y en general cualquier otra consideración que se estime necesaria.

Este tipo de licencias, por lo general, no permiten que el software sea modificado, desensamblado, copiado o distribuido de formas no especificadas en la propia licencia (piratería de software), regula el número de copias que pueden ser instaladas e incluso los fines concretos para los cuales puede ser utilizado. La mayoría de estas licencias limitan fuertemente la responsabilidad derivada de fallos en el programa.

Los fabricantes de programas sometidos a este tipo de licencias por lo general ofrecen servicios de soporte técnico y actualizaciones durante el tiempo de vida del producto.

Algunos ejemplos de este tipo de licencias son las llamadas acuerdo de licencia de usuario final (ALUF), también llamado en algunos países contrato de licencia de usuario final (CLUF). En el mundo anglosajón se llama end-user license agreement (EULA), pero los contratos en español deben recoger el término en español.

Software de dominio público (sin licencia)

Se permite uso, copia, modificación o redistribución con o sin fines de lucro.

Según su destinatario

Licencia de distribuidores

En este tipo de contrato, se le asigna derechos restringidos a un mayorista para que venda el producto software a terceros dando una remesa o comisión al fabricante. La misma puede ser por primera venta o licencia de renovación de contrato. No se trata de una licencia de uso en términos jurídicos, sino más bien en un acuerdo comercial en el que no tiene por qué ser cedido el derecho de distribución necesariamente. Puede darse el caso de simple actividad comercial en la que el distribuidor ni siquiera tenga contacto con el software, y éste como elemento y la licencia de uso en sí sea directamente suscrita y puesta a disposición por parte del fabricante. Encargándose el distribuidor del correspondiente cobro al usuario y pago al fabricante menos su comisión.

**2. Define GPL, LGPL y BSD enumerando diferencias así como ventajas e inconvenientes de unas y Licencia GPL**

Lo que importa de las licencias GPL, es el destino “final” que tendrán los programas con este tipo de licencia, ya que la decisión inicial es la decisión final. Si decides publicar tu programa

con los términos de la licencia GPL, este será su destino, ya que no habrá modo alguno de querer cambiar de licencia, ni mucho menos de querer volverlo privativo (cerrar el código del programa).

1. Se puede copiar, regalar o vender a terceros el software, sin tener la "obligación" de pagar por ello.

2. El software modificado no debe tener costo por la licencia.

3. Tiene que incluir el código fuente.

4. Un programa con licencia GPL que ha sido modificado automáticamente es publicado con licencia GPL.

Sobre el primer punto, no confundamos libertad con gratuidad, el hecho de que se llame software libre se refiere a la libertad de liberar el programa con el código fuente. Esta confusión radica en que la palabra inglesa "free" tiene doble significado: libre y gratis, la FSF hace hincapié en ese sentido. Sobre el último punto, en algún momento creo haber escuchado o leído por algún medio que la única forma de poder renunciar a la licencia GPL es que absolutamente todos los desarrolladores de aquel programa estén de acuerdo, pues creo que es un absurdo, tal vez es la única libertad que el software libre deja de lado, pero es para el bien de la comunidad del software libre, imagínense como sería el desarrollo de ciertos programas si llega un momento en el que decide no ser más GPL y cerrar su código, o si es que el mismo diablo personificado decide comprar sus derechos de GPL, para continuar su desarrollo secretamente, sería una verdadera pérdida de tiempo haber estado realizando un programa con licencia GPL para luego iniciar un proyecto desde cero teniendo el código libre para mejorarlo. No creen?. En mi humilde opinión diría que gracias al software libre podemos compartir conocimientos para lograr desarrollos rápidos y de calidad, y también la importancia radica en evitar la formación de grandes monopolios de software privativos, aunque hay un caso en particular que se ha cuajado de lleno en este rango.

Licencia LGPL

LGPL es una licencia que es prácticamente igual a la GPL, pero permite que software con esta licencia estén integrado en software privativos. La biblioteca C de Linux posee este tipo de licencia, imagínense por un momento que pasaría si solo fuera GPL?, inevitablemente solo se podrían crear aplicaciones para Linux u otros sistemas que manejen la filosofía de software libre, pero como es LGPL, está adaptada para poder crear también aplicaciones privativas.

Licencia BSD

A diferencia de la licencia GPL que obliga a incluir el código fuente en sus liberaciones siendo imposible cambiarla, la licencia BSD respeta las libertades del software libre excepto la de poder modificar el tipo de licencia, por lo tanto, no pertenece al rango del software libre, pero mantiene una estrecha relación con la GPL. El punto más controversial de la licencia BSD, es que tú tienes la libertad de poder cambiar tu licenciamiento cuando quieras, si quieres puedes convertir en privativo tus programas bajo la modalidad BSD. Por lo demás tiene las mismas libertades que la GPL, Se puede copiar, regalar o vender a terceros el software, puedes cobrarlo o no, es decir, eres absolutamente libre para elegir qué quieres hacer o no.

Por último, quiero mostrarles una licencia que es netamente privativa y comparen las grandes diferencias con el software libre o con el open source. otras.

**3. Define el concepto copyleft y evalúa posible su conveniencia frente a las licencias con derechos reservados.**

El copyleft es una práctica legal que consiste en el ejercicio del derecho de autor con el objetivo de propiciar el libre uso y distribución de una obra, exigiendo que los concesionarios preserven las mismas libertades al distribuir sus copias y derivados. Los autores pueden aplicar una licencia con copyleft a programas informáticos, obras de arte, textos o cualquier tipo de trabajo creativo que sea regido por el derecho de autor.

El copyleft es propuesto como alternativa y defensa contra las restricciones al público en las que normalmente incurren los editores y la industria del entretenimiento. Se pretende así que quienes poseen los derechos patrimoniales de una obra la ofrezcan mediante una licencia libre; al mismo tiempo que una cláusula adicional protege los derechos ofrecidos en la licencia de intentos subsecuentes de privatización por parte del público (mientras la obra no pase al dominio público). Las licencias con copyleft son entonces una de las categorías principales de licencia libre; en contraste con las llamadas licencias permisivas o sin copyleft, y en contraste con el dominio público.

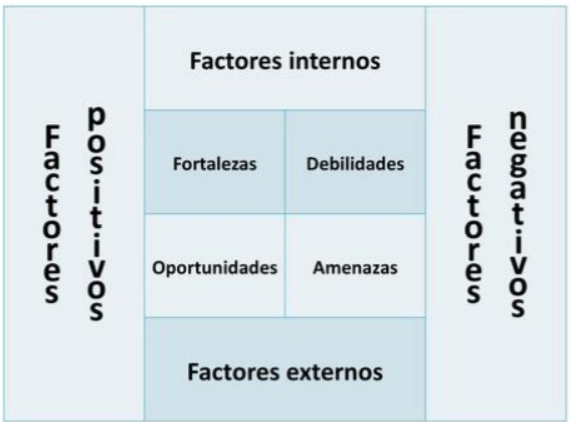
El término surge en la comunidad del software libre como un juego de palabras en torno a copyright: «derecho de autor» en inglés (literalmente «derecho de copia») con otro sentido, el de left: pretérito y participio del verbo «dejar» o «permitir» (literalmente «dejada-copia»). Left también corresponde al concepto de «izquierda», en contraste con right, que es derecho.

## Actividad 2

**El análisis DAFO es una herramienta muy sencilla y a la vez muy útil como mecanismo de análisis de la realidad y de soporte para la toma de decisiones. Su nombre es un acrónimo de**

**las cuatro ideas que centran el análisis: Debilidades, Amenazas, Fortalezas y Oportunidades. En inglés el término utilizado es SWOT (Strengths, Weaknesses, Opportunities y Threats).**


**Puedes ampliar esta información en el artículo de Wikipedia sobre Análisis DAFO**




**El objetivo de esta actividad es la realización de dos análisis DAFO, uno para el software libre y otro para el software privativo. Como veréis, se trata tanto de una ampliación como de un análisis previo para la Actividad 1.**

**Una vez realizados los dos análisis, intenta argumentar en qué entornos operativos resulta más conveniente la implementación de sistemas informáticos con software bajo cada tipo de licencia. O dicho de otra forma, ¿basándonos en qué nos decidimos por utilizar software libre o software privativo en nuestras empresas?**

SOFTWARE LIBRE

Fuerzas

I+D realizada por voluntarios

La labor realizada de manera voluntaria por muchas personas significaría miles de millones de euros para las empresas del sector. Los desarrolladores libres a menudo modifican el código en su interés propio, pero por consiguiente se convierte en una mejora para los demás.

A veces las empresas del sector de software libre ofrecen recompensas a estos programadores para la mejora de versiones, para mantener los desarrollos activos e interesantes para los clientes.

Lanzamientos de versiones frecuentes: Generalmente las versiones de los programas de código abierto son liberadas con un menor lapso de tiempo, debido a las frecuentes integraciones que se realizan por una heterogénea masa de programadores.

Desarrollo y pruebas paralelas: El modelo de desarrollo que utiliza el software libre permite a grupos o personas trabajar de manera independiente, cada uno trabajando en partes distintas del software.

El desarrollo paralelo de las funcionalidades permite escoger posteriormente la mejor implementación de las funcionalidades. Y el paralelismo en las pruebas aumenta el rendimiento de las mismas, que redunda en un menor tiempo de corrección de fallos.

Cultura de comunidad, solidaridad: La comunidad permite la realización de un trabajo una única vez, en vez de que cada empresa implemente lo mismo varias veces, con la consiguiente pérdida de trabajo. Desde una vista económica global, se elimina la pérdida asociada al trabajo duplicado.

Accesibilidad del producto asegurada

Un producto de software libre con el suficiente interés en la comunidad, no podría ser eliminado del negocio en un corto plazo de tiempo. Una empresa comercial podría decidir que no le interesa para su estrategia seguir una determinada línea de productos y discontinuarla bruscamente.

Además, el hecho de la posesión del código fuente por parte de los clientes permitiría un mantenimiento interno, y se podría continuar la vida del producto indefinidamente.

Oportunidades

Conectividad por Internet: La comunidad se comunica a través de Internet. Las listas de correo, chats, grupos de noticias y sitios Web permiten el crecimiento de esas comunidades, permitiendo que exista un desarrollo y mantenimiento las 24 horas del día los siete días de la semana.

Estructura de soporte competitiva: El software comercial depende de un monopolio, en cuanto a que el código fuente no está disponible. Cuando un producto de software libre supera el umbral crítico de usuarios, el soporte que puede dar cualquier empresa es el mismo, y por tanto la competencia aumenta, lo que beneficiará al mercado.

Debilidades

Falta de propietario del producto:

Limita la posibilidad de determinar con garantías cual será el futuro estratégico del producto. Una compañía comercial podría comprometerse a una determinada compatibilidad, y se le podría demandar si no la cumple, pero en un proyecto de código abierto no se puede garantizar nada.

Muchos directores de los departamentos de tecnología confían más en las empresas, que en un movimiento dirigido por una comunidad, puesto que pueden poner nombre a las responsabilidades ante los problemas.

Difícil de originar un proyecto: Se debe lograr una masa crítica de buenos desarrolladores interesados en el proyecto para que este pueda "despegar". Los inicios de muchos proyectos son debidos a una motivación personal de un programador, y debe suscitar suficiente interés para poder crear una comunidad que de respaldo al software.

Amenazas

Riesgo de fragmentación: Ocurre cuando se rompe el código fuente en varias líneas de desarrollo. Esto puede ocurrir porque distintas personas tomen un camino distinto en el desarrollo, de manera que cada una crea su propia versión del código base compartido. Esto puede ocurrir por que los desarrolladores no alcanzan un acuerdo sobre el camino a seguir, o porque los intereses de las partes implicadas son muy distintos.

Falta de aplicaciones compatibles: Los proyectos de código abierto tienen limitaciones cuando aún no son aplicaciones extendidas en la comunidad informática. Hasta que un programa no se reconoce como exitoso, pocas personas crean herramientas para él.

Necesidad de control de versiones: El código base debe ser puesto en un repositorio común, y cada cambio debe ser integrado y probado antes de lanzar cada versión.

**SOFTWARE PRIVADO y conclusión para cuando utilizar software libre o no.**

Fuerzas

Para desarrollar un software solo dispondremos de nuestras capacidades para desarrollarlo. Con lo cual también tendremos que invertir más dinero en I+D para el desarrollo del software para nuestras necesidades o las necesidades del usuario final. Ya que en este apartado estaremos solos por decirlo de alguna forma. No tendremos las ayudas de terceros para realizar mejoras o corregir posibles errores que puedan surgir.

Oportunidades

Las oportunidades serán más limitadas que en el software libre, si necesitas una mejora o lo dices a tu programador que te añada una característica o contratas a una persona para que la implemente. Si quieres mejorar o tener más oportunidades necesitaras más inversión.

Para los distribuidores de software para esto es una ventaja ya que los clientes tendrán una dependencia total de su software a la hora de trabajar y realizar posibles cambios. Con lo cual se aseguran una clientela.

Debilidades

Menos seguro, ya que si tienes algún problema dependerás de ti mismo 100% y si surge algún problema de seguridad dependerás de ti mismo o de la empresa que te distribuya el software para corregir las posibles amenazas externas.

Amenazas

A más debilidades mayores amenazas, también si dispones de algo exclusivo correrás el riesgo de que alguien lo quiera sin pagar como puede ser la piratería.