

Tarea 1

Clasificación del software

Parte 1

Familiarizarnos con el software libre, de código abierto y propietario.

1. ¿Qué libertades del software libre crees que debe cumplir el software open source? Razona tu respuesta

Debería cumplir como mínimo las libertades de estudio y modificación y de mejora, excluyendo las demás en la mayoría de casos por la necesidad de pasar por un filtro para que una mala persona cause mucho daño.

2. ¿Dónde crees que radica la principal diferencia entre ambos?

En que el software libre es mucho menos limitado que el open source debido a que el open source surge a partir del software libre pero siendo gestionado por una entidad (i.e. Libre office),

3. ¿Cuál es más estricto? ¿Por qué?

Es más estricto el open source debido a que no es obligatorio que permita modificación a pesar de dejar ver el código, y aunque permita la modificación lo normal es que no se permita la distribución fuera de canales oficiales.

4. ¿Qué modalidades de pago nos podemos encontrar en un software propietario? ¿Y en uno de software libre? ¿Y en uno de código abierto?

Para el software libre tenemos las modalidades gratuita, de donativos o de servicios y soporte.

Para el software propietario tenemos las modalidades de pago único, de suscripción y de licencia por usuario/equipo.

Para el software de código abierto tenemos las modalidades gratuita o de servicios y soporte.

5. ¿Puede ser propietario un software libre? ¿Y uno de código abierto?

Razona tus respuestas

El software propietario no puede ser libre debido a que nos estaríamos comiendo las libertades del software libre, las cuales son en su mayoría incompatibles con las limitaciones del software propietario.

El software propietario no puede ser de código abierto debido a que el código abierto incumple las limitaciones de distribución y de acceso al código fuente entre otras.

6. ¿Es todo software libre también de código abierto? ¿Por qué?

No, porque a pesar de que suelen ir de la mano, hay veces que el software libre utiliza licencias más restrictivas que no coinciden del todo con el software de código abierto. En definitiva, el software libre se centra en las libertades de los usuarios mientras que el software de código abierto se centra en la accesibilidad y visibilidad del código fuente.

7. ¿Es todo software de código abierto también software libre? ¿Por qué?

Si se podría considerar así debido a que el software de código abierto cumple con los principios del software libre.

Parte 2

Identificar las licencias Open Source más utilizadas

8. Ve a la página web de la Open Source Initiative y recopila las licencias de código abierto más reconocidas y utilizadas. ¿Cuántas son?

Apache 2.0, Common Development and Distribution License 1.0, Eclipse Public License version 2.0, GNU General Public License version 2, GNU General Public License version 3, GNU Lesser General Public License version 2.1, GNU Lesser General Public License version 3, GNU Library General Public License version 2, Mozilla Public License 2.0, The 2-Clause BSD License, The 3-Clause BSD License, The MIT License.
Son 12 en total.

9. ¿Es Apache una licencia copyleft? ¿Por qué?

No, la licencia de Apache no es un copyleft debido a que estas licencias requieren que las obras modificadas o derivadas también se distribuyan con los mismos términos que la licencia original.

10. ¿Cómo añadirías una licencia Apache 2.0 a tu trabajo?

Asegúrate de que tienes los derechos necesarios: Antes de aplicar cualquier licencia a tu trabajo, debes asegurarte de tener los derechos o permisos necesarios para hacerlo. Si estás utilizando contenido o componentes de terceros en tu trabajo, verifica si tienen licencias que sean compatibles con la Licencia Apache 2.0.

Agrega un archivo LICENSE: Crea un archivo de texto llamado "LICENSE" en la raíz de tu proyecto. Este archivo contendrá el texto completo de la Licencia Apache 2.0. Puedes obtener el texto oficial de la licencia desde el sitio web de la Apache Software Foundation.

Incluye los avisos de copyright: En cada archivo de tu proyecto, puedes añadir un aviso de copyright que indique los derechos de autor y la licencia aplicable. Puedes utilizar una estructura similar a esta:

Copyright [Año] [Nombre del titular del copyright]

Licensed under the Apache License, Version 2.0 (la "Licencia");
no puedes utilizar este archivo excepto en conformidad con la Licencia.
Puedes obtener una copia de la Licencia en

<http://www.apache.org/licenses/LICENSE-2.0>

A menos que lo requiera la ley aplicable o se acuerde por escrito, el software distribuido bajo la Licencia se distribuye "TAL CUAL",
SIN GARANTÍAS NI CONDICIONES DE NINGÚN TIPO, ya sean expresas o implícitas.
Consulta la Licencia para obtener el texto específico que rige los permisos y limitaciones bajo la Licencia.

Actualiza los metadatos: Si tu proyecto tiene metadatos o archivos de configuración que indican la licencia, asegúrate de actualizarlos para reflejar que estás utilizando la Licencia Apache 2.0.

11. ¿Cuántas versiones tiene la licencia GPL?

GPL versión 1: Fue la primera versión de la GPL y se publicó en 1989. Estableció los principios fundamentales de la licencia y brindó protección a los derechos de los usuarios y desarrolladores de software libre.

GPL versión 2: Fue lanzada en 1991 y ha sido una de las versiones más utilizadas y reconocidas. Introdujo cambios y mejoras significativas en comparación con la versión anterior, como la adición de la cláusula de copyleft y la expansión de los derechos de los usuarios.

GPL versión 3: Fue publicada en 2007 y es la versión más reciente de la GPL. Introduce cambios adicionales para adaptarse a los nuevos desafíos y avances tecnológicos. Incluye disposiciones relacionadas con la protección de la libertad de los usuarios frente a las restricciones tecnológicas y la promoción de la compatibilidad con otras licencias de software libre.

12. ¿Puedes añadir un software con licencia GPL dentro de un proyecto propietario? ¿Qué tendrías que hacer?

Sí, puedes incorporar software con licencia GPL dentro de un proyecto propietario, pero hay ciertas consideraciones y restricciones que debes tener en cuenta. Aquí hay algunas pautas generales:

Comprende las restricciones de la GPL: La Licencia Pública General de GNU (GPL) es una licencia copyleft, lo que significa que si incorporas código GPL en tu proyecto, el proyecto en su conjunto debe ser distribuido bajo los términos de la GPL. Esto implica que el código fuente de tu proyecto propietario también debe estar disponible para los usuarios y que no puedes imponer restricciones adicionales a quienes utilicen o modifiquen tu software.

Utiliza la comunicación a través de procesos: Una forma de incluir software con licencia GPL en un proyecto propietario es establecer una comunicación clara y separada entre los componentes GPL y el resto de tu proyecto. Puedes establecer una interfaz o API que permita la interacción entre los componentes, pero asegúrate de que sean entidades independientes, de modo que el código GPL no se mezcle directamente con el código propietario.

Utiliza enlaces dinámicos: Si deseas incluir una biblioteca con licencia GPL en tu proyecto propietario, puedes enlazarla dinámicamente en lugar de enlazarla estáticamente. Esto significa que el código propietario se conecta con la biblioteca GPL en tiempo de ejecución en lugar de estar vinculado directamente en el momento de la compilación. Esto puede ayudar a mantener una separación clara entre los componentes y a cumplir con los requisitos de la GPL.

Recuerda que estas son solo pautas generales, y cada situación puede ser única. Es importante consultar a un especialista legal para obtener asesoramiento específico sobre cómo incorporar software con licencia GPL en un proyecto propietario y asegurarse de cumplir con las obligaciones de la licencia.

13. ¿Qué significa LGPL? ¿Cuál es su versión más actual? ¿Dirías que es más estricta o menos que la GPL? Cuidado, entendemos por estricta o fuerte a la menos permisiva (o la más enemiga del código propietario).

LGPL significa Licencia Pública General Menor de GNU (Lesser General Public License, en inglés). Es una licencia de software libre creada por la Free Software Foundation (FSF) que permite la combinación de código LGPL con software propietario.

La versión más actual de la LGPL es la LGPL versión 3. Esta versión fue publicada en 2007 y está diseñada para abordar problemas específicos relacionados con bibliotecas de software. Proporciona protección a los derechos de los usuarios y desarrolladores de software libre, al tiempo que ofrece cierta flexibilidad para la compatibilidad con software propietario.

En términos de restricciones y permisividad en relación con el código propietario, se podría considerar que la LGPL es menos estricta que la GPL. Mientras que la GPL requiere que el software derivado también sea distribuido bajo los términos de la GPL (copyleft fuerte), la LGPL permite que las bibliotecas bajo esta licencia se utilicen en proyectos propietarios sin imponer la obligación de que el software propietario se distribuya bajo los términos de la LGPL.

Sin embargo, es importante tener en cuenta que la LGPL tiene ciertas condiciones y requisitos que deben cumplirse cuando se utiliza código LGPL en un proyecto propietario. Por ejemplo, se debe permitir la modificación y reemplazo de la biblioteca LGPL, y se debe proporcionar acceso al código fuente de la biblioteca LGPL a los usuarios que lo soliciten.

Siempre es recomendable revisar detalladamente los términos y condiciones de la LGPL y, si tienes dudas específicas sobre tu caso en particular, es aconsejable consultar con un experto legal para obtener asesoramiento adecuado.

14. ¿Cuál es la principal diferencia entre GPL y LGPL?

La principal diferencia entre la GPL (Licencia Pública General de GNU) y la LGPL (Licencia Pública General Menor de GNU) radica en cómo se aplican a los proyectos de software.

La GPL es una licencia copyleft que requiere que cualquier software derivado o combinado con código GPL también se distribuya bajo los términos de la GPL. Esto significa que si utilizas código GPL en tu proyecto, todo el proyecto debe ser distribuido como software libre y poner a disposición el código fuente correspondiente.

En contraste, la LGPL es una licencia copyleft modificada que se enfoca en bibliotecas de software. Permite que estas bibliotecas se utilicen en proyectos propietarios sin imponer la obligación de que el software propietario se distribuya bajo los términos de la LGPL. Sin embargo, la LGPL tiene ciertas condiciones y requisitos que deben cumplirse, como permitir la modificación y reemplazo de la biblioteca LGPL y proporcionar acceso al código fuente de la biblioteca a los usuarios que lo soliciten.

En resumen, la principal diferencia entre la GPL y la LGPL es que la GPL se aplica a todo el proyecto en el que se utiliza código GPL, mientras que la LGPL se enfoca en bibliotecas y permite que se utilicen en proyectos propietarios sin imponer las mismas restricciones en la distribución del software propietario.

15. Ordena de la más permisiva a la más restrictiva: GPL, MIT, BSD. ¿Por qué?

Licencia MIT: Es considerada la licencia más permisiva de las tres mencionadas. Permite a los usuarios hacer casi cualquier cosa con el software, incluyendo la modificación, distribución y uso tanto en proyectos de código abierto como en proyectos propietarios. La licencia MIT exige solo la inclusión del aviso de copyright y la exención de responsabilidad en el software distribuido.

Licencia BSD: La licencia BSD también es bastante permisiva, aunque presenta algunas diferencias respecto a la licencia MIT. Esta licencia permite la modificación, distribución y uso del software, tanto en proyectos de código abierto como en proyectos propietarios. Al igual que la licencia MIT, la licencia BSD requiere la inclusión del aviso de copyright y la exención de responsabilidad, pero puede tener cláusulas adicionales dependiendo de la variante específica de la licencia BSD.

Licencia GPL: La Licencia Pública General de GNU (GPL) es considerada más restrictiva en comparación con las licencias MIT y BSD. La GPL establece que cualquier software derivado o combinado con código GPL debe distribuirse bajo los mismos términos de la GPL, lo que implica que el software derivado también debe ser software libre y poner a disposición su código fuente. No permite la combinación con software propietario sin que este último deba ser distribuido bajo los términos de la GPL.

Parte 3

Reconocer las licencias de las aplicaciones

16. Busca en internet dos ejemplos de software freeware y otros dos de shareware.

Software freeware:

Everything: Es un programa de búsqueda en el escritorio de archivos que transforma la forma en que utilizas tu PC.

Launchy: Es un lanzador de aplicaciones que te permite acceder rápidamente a tus programas favoritos.

Software shareware:

WinRAR: Es un programa que permite comprimir y descomprimir archivos.

SoftPerfect Network Protocol Analyzer: Es una herramienta que permite analizar, depurar, mantener y supervisar las redes locales y conexiones a Internet.

17. ¿Qué es la licencia MPL? Indica un ejemplo de software que la utilice

La licencia MPL es una licencia de código abierto que permite a los desarrolladores modificar, distribuir y utilizar el software de acuerdo con los términos establecidos en la licencia. Proporciona una combinación de características de licencias de código abierto y licencias de software propietario, lo que permite a los desarrolladores combinar el software MPL con software propietario en ciertos casos.

Un ejemplo de software que utiliza la licencia MPL es el navegador web Firefox. Firefox es un navegador popular y ampliamente utilizado que se basa en el código fuente abierto y está disponible de forma gratuita para su descarga y uso. Otro ejemplo es el cliente de correo electrónico Thunderbird, que también utiliza la licencia MPL y ofrece una alternativa de código abierto para la gestión de correos electrónicos.

18. ¿Qué es un contrato EULA o CLUF? Localiza en internet el contrato eula de Windows 10. Lee detenidamente los apartados 2 y 3. ¿A qué se refieren? Destaca alguna característica que te resulte interesante y que esté relacionada con el tema de estudio.

19. Las licencias de distribución de Microsoft son: OEM, Retail y de volumen. Explica resumidamente sus características y su propósito.

Es un acuerdo legal entre el licenciante y el usuario final del software.

El apartado 2 habla sobre la concesión de Licencia ya que es el que da la licencia al usuario, y el apartado 3 habla sobre la propiedad Intelectual establece la propiedad del software por parte del titular de los derechos de autor.

20. ¿Qué licencia (Retail u OEM) tiene tu Windows 10? Explica cómo lo has averiguado.

Tengo licencia OEM, se averigua ejecutando CMD como administrador e introduciendo el comando "slmgr /dli"

21. ¿Qué licencia usa bitcoin?

Bitcoin usa la licencia MIT.

22. ¿Qué licencia usa OpenOffice? ¿Y LibreOffice?

OpenOffice usa la licencia apache 2.0 y LibreOffice usa una licencia dual de LGPL y MPL.

Parte 4

Reconocer y clasificar las aplicaciones según su propósito

23. Clasifica las siguientes aplicaciones según los tipos estudiados:

24. Indica 2 ejemplos de aplicaciones de los siguientes tipos (no usar los anteriores)