

LICENCIAS 101: ¿CÓMO DEBO LICENCIAR MIS PROYECTOS?

Abdel G. Martínez L.

Software Freedom Day
Septiembre 19, 2015



Esta presentación tiene la licencia para uso y distribución bajo
Creative Commons Attribution-NonCommercial-ShareAlike 3.0
<http://creativecommons.org/licenses/by-nc-sa/3.0/es/legalcode.es>

AGENDA

Licencias 101: ¿Cómo debo licenciar mis proyectos?

- Entendiendo lo que son las licencias
- Secciones del software libre
- Lo que promete y lo que detiene
- Distribución de proyectos
- ¿Qué es un derivado?
- Niveles de libertad
- Escenarios:
 - *Contribuyendo a un proyecto existente*
 - *Desarrollando un proyecto de software*
 - *En términos de documentación*
 - *Otra data para los programas*

ENTENDIENDO LO QUE SON LAS LICENCIAS

- Una licencia se refiere a un permiso, así como el documento que lo guarda.
- Puede concederse por un partido a otro partido como un elemento de un acuerdo entre las partes.
- Es un contrato con cláusulas, donde participa el licenciante y el licenciatiario.



SECCIONES DEL SOFTWARE LIBRE



Libre distribución



Código fuente libre



Trabajos derivados



Integridad del código



Todas la personas tienen derecho



Derechos para todos los individuos



Distribución de la licencia



No es explícito a un producto



No debe restringir otro software



Licencia debe ser neutral

LO QUE PROMETE

Cuando son componentes de software modulares


Cuando quiere apalancar revisiones de colegas expertos

Cuando quieres que tu código sea fácilmente accesible y disponible

Si quiere probar y experimentar usando con un costo muy bajo

Reduce el riesgo de la alta dependencia a un único proveedor

LO QUE DETIENE

- 
- 1. Puede comprometer la propiedad intelectual
 - 2. Falta de integración con infraestructura actual
 - 3. Falta de certificaciones, no existen garantías
 - 4. No tiene la cultura para gobernarlo
 - 5. Ama tanto el software propietario que necesita sus beneficios costosos

DISTRIBUCIÓN DE PROYECTOS



Medio físico



Envío por red



Suscripciones

¿QUÉ ES UN DERIVADO?

- Consiste en modificar, combinar y enlazar el código existente en un nuevo cuerpo de trabajo.
- La obra derivada debe garantizar los derechos de autor de la obra original.
- Ejemplos: traducción, adaptación musical, adaptación a otro medio de comunicación o un resumen.



*Shepard Fairey creó un trabajo derivado (der)
basado en la foto de Manny García (izq)*

NIVELES DE LIBERTAD

Licencias Permisivas

"Dame crédito"

Derivados pueden re-licenciar

Dar crédito a autores originales

Ejemplo: BSD, Apache

Débilmente Permisivas

"Dame correcciones"

Diferencia entre fuente y binario

Permite desarrollo comercial

Tiende hacia librerías y plataformas

Ejemplo: Mozilla, LGPLv3, Eclipse

Fuertemente Protectora

"Dame todo"

Todo desarrollo de contribuido de vuelta

El dueño del copyright mucho el control

Limita la adaptación comercial

Ejemplo: GPLv3, AGPLv3

ESCENARIOS:

Contribuyendo a un proyecto existente

- Usualmente se libera la versión modificada bajo la misma licencia que el trabajo original.
- Esto facilita la cooperación con los mantenedores del proyecto.
- Sin embargo, si deseas liberar tus contribuciones bajo una licencia diferente, debes asegurarte que la licencia original permita el uso del material bajo la licencia elegida.
- En estos casos, se recomienda el uso de **copyleft**, para asegurarte que el trabajo derivado persista de esta manera.



ESCENARIOS:

Desarrollando un proyecto de software

Pequeños programas

- 300 líneas o menos
- Apache License 2.0
- Previene demandas por violación de patentes

Librerías

- En caso de librerías que implementan estándares libres (Ogg Vorbis, WebM) utilizar Apache License 2.0
- Para librerías generales utilizar GNU LGPL por su flexibilidad en términos de uso y libertad de software
- Para librerías especializadas y que no tienen competencia no libre utilizar GNU GPL

Software de servidor

- En caso de probabilidad de que hagan versiones mejoradas de tu programa y no lo distribuyan utilizar la versión GNU AGPL, asegurando una dependencia a la red.

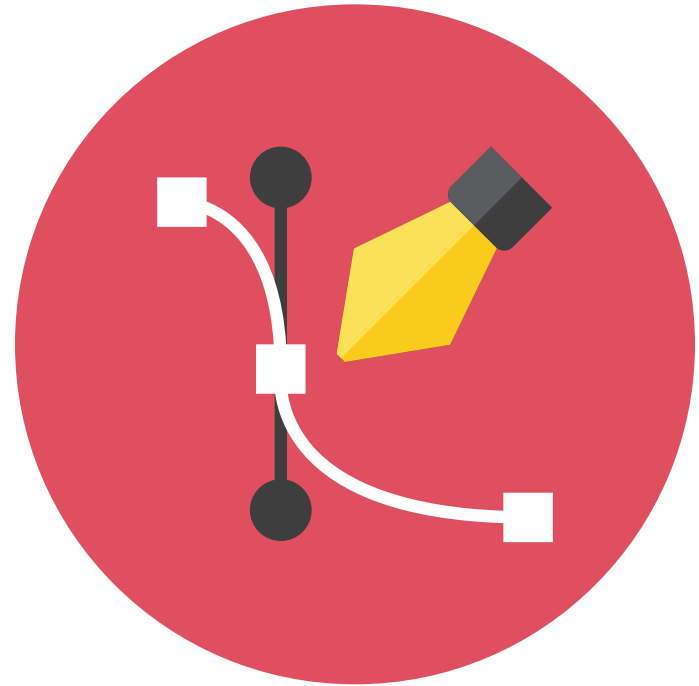
ESCENARIOS: En términos de documentación

- Tutoriales, manuales de referencia u otras documentaciones grandes, utilizar GNU FDL.
- Si es una documentación más puntual, como una tarjeta de presentación, utilizar CC-BY.
- Si la documentación incluye código fuente de software se puede hacer una mezcla de GNU FDL con CC0.



ESCENARIOS: Otra data para los programas

- Esto incluye íconos, artes, gráficos, fuentes y data geográfica.
- Si se utilizan para un proyecto de software, utilizar la misma licencia existente.
- Esto evita dudas en distribución.
- Si no es un proyecto de software, utilizar Creative Commons Attribution-ShareAlike.



RESUMIENDO



Lo quiero simple y permisivo

- MIT License
- Ejemplo: jQuery y Rails



Me preocupa las patentes

- Apache License
- Ejemplo: Apache, SVN



Quiero compartir mejoras

- GPLv2 o GPLv3
- Ejemplo: Linux, Git, Wordpress



Mi contenido no es código

- Creative Commons



Quiero más opciones

- Apache, GPL, MIT, Artistic License, Eclipse Public License, BSD, LGPL, Mozilla Public License, No License, CC0

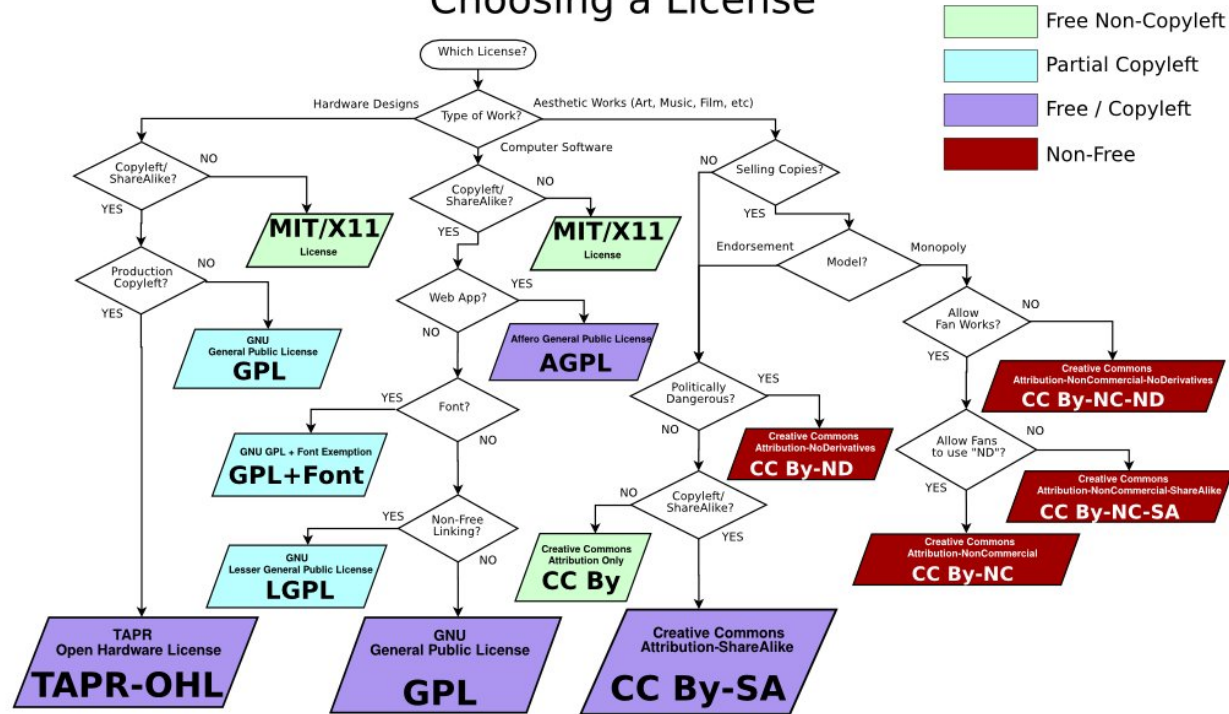


No quiero elegir una licencia

- Esto no evita las leyes de copyright
- Dominio público: Unlicense o CC0

RESUMIENDO (EXTRA)

A "Best Practices" Flowchart for Choosing a License



¿PREGUNTAS?

Abdel G. Martínez L.
@abdelgmartinezl
<http://abdelmartinez.com>