

Proyecto Harmony considerado dañino

Bradley M. Kuhn

Proyecto Harmony considerado dañino

Bradley M. Kuhn



En Defensa del Software Libre

En Defensa del Software Libre es una revista de teoría sobre Software y Cultura Libres. Se edita en papel y se distribuye gratuita y libremente en formato digital.



En Defensa del Software Libre

En Defensa del Software Libre es una revista de teoría sobre Software y Cultura Libres. Se edita en papel y se distribuye gratuita y libremente en formato digital.

©2017– En Defensa del Software Libre.

<https://endefensadelsl.org>

Salvo donde se exprese lo contrario, los artículos y la edición se liberan bajo la Licencia de Producción de Pares.

https://endefensadelsl.org/ppl_deed_es.html

©2017– En Defensa del Software Libre.

<https://endefensadelsl.org>

Salvo donde se exprese lo contrario, los artículos y la edición se liberan bajo la Licencia de Producción de Pares.

https://endefensadelsl.org/ppl_deed_es.html

Licencia de Producción de Pares

Ud. es libre de

- Compartir - copiar, distribuir, ejecutar y comunicar públicamente la obra
- Hacer obras derivadas

Licencia de Producción de Pares

Ud. es libre de

- Compartir - copiar, distribuir, ejecutar y comunicar públicamente la obra
- Hacer obras derivadas

Bajo las condiciones siguientes:



Atribución – Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o que apoyan el uso que hace de su obra).



Compartir bajo la Misma Licencia – Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.



No Capitalista – La explotación comercial de esta obra sólo está permitida a cooperativas, organizaciones y colectivos sin fines de lucro, a organizaciones de trabajadores autogestionados, y donde no existan relaciones de explotación. Todo excedente o plusvalía obtenidos por el ejercicio de los derechos concedidos por esta Licencia sobre la Obra deben ser distribuidos por y entre los trabajadores.

Bajo las condiciones siguientes:



Atribución – Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o que apoyan el uso que hace de su obra).



Compartir bajo la Misma Licencia – Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.



No Capitalista – La explotación comercial de esta obra sólo está permitida a cooperativas, organizaciones y colectivos sin fines de lucro, a organizaciones de trabajadores autogestionados, y donde no existan relaciones de explotación. Todo excedente o plusvalía obtenidos por el ejercicio de los derechos concedidos por esta Licencia sobre la Obra deben ser distribuidos por y entre los trabajadores.

Entendiendo que

- **Renuncia** - Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor.
- **Dominio Público** - Cuando la obra o alguno de sus elementos se halle en el dominio público según la ley vigente aplicable, esta situación no quedará afectada por la licencia.
- **Otros derechos** - Los derechos siguientes no quedan afectados por la licencia de ninguna manera:
 - Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior;
 - Los derechos morales del autor;
 - Derechos que pueden ostentar otras personas sobre la propia obra o su uso, como por ejemplo derechos de imagen o de privacidad.
- **Aviso** - Al reutilizar o distribuir la obra, tiene que dejar muy en claro los términos de la licencia de esta obra. La mejor forma de hacerlo es enlazar a esta página.

Entendiendo que

- **Renuncia** - Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor.
- **Dominio Público** - Cuando la obra o alguno de sus elementos se halle en el dominio público según la ley vigente aplicable, esta situación no quedará afectada por la licencia.
- **Otros derechos** - Los derechos siguientes no quedan afectados por la licencia de ninguna manera:
 - Los derechos derivados de usos legítimos u otras limitaciones reconocidas por ley no se ven afectados por lo anterior;
 - Los derechos morales del autor;
 - Derechos que pueden ostentar otras personas sobre la propia obra o su uso, como por ejemplo derechos de imagen o de privacidad.
- **Aviso** - Al reutilizar o distribuir la obra, tiene que dejar muy en claro los términos de la licencia de esta obra. La mejor forma de hacerlo es enlazar a esta página.

Índice general

1 Proyecto Harmony considerado dañino — Bradley Kuhn	9
Una cesión de copyright que carece de garantías reales .	10
¿Regalar tus derechos para que las compañías sientan mariposas en la panza?	13
La “Elección de Ley” y los Arreglos Contractuales embarran los reclamos de copyright	14
Problemas para hacer cumplir el copyright individualmente contra terceras partes	16
Entrante=Saliente es todo lo que necesitas	18
Los Hackers de Linux han fijado ingeniosamente entrante=saliente	20
¿Qué pasa con el relicenciamiento?	21
¿Una parcialidad anti copyleft fuerte?	22
Esto no es Creative Commons, pero si lo fuera, ¿vale la pena emularlo?	23
Conclusiones	24
Véase también (en inglés)	26

Índice general

1 Proyecto Harmony considerado dañino — Bradley Kuhn	9
Una cesión de copyright que carece de garantías reales .	10
¿Regalar tus derechos para que las compañías sientan mariposas en la panza?	13
La “Elección de Ley” y los Arreglos Contractuales embarran los reclamos de copyright	14
Problemas para hacer cumplir el copyright individualmente contra terceras partes	16
Entrante=Saliente es todo lo que necesitas	18
Los Hackers de Linux han fijado ingeniosamente entrante=saliente	20
¿Qué pasa con el relicenciamiento?	21
¿Una parcialidad anti copyleft fuerte?	22
Esto no es Creative Commons, pero si lo fuera, ¿vale la pena emularlo?	23
Conclusiones	24
Véase también (en inglés)	26

Proyecto Harmony considerado dañino

—Bradley Kuhn—

Esta traducción y el artículo original se liberan bajo Creative Commons Atribución CompartirDerivadasIgual 3.0 de Argentina¹ y Estados Unidos² respectivamente. Traducido por fauno y mpj.

Mucha publicidad se diseña para convencernos de comprar o usar algo que no necesitamos. Cuando escucho a alguien zumbando sobre alguna cosa nueva, maravillosa, me preocupo pensando que esos tipos están tratando de venderme algo.

Muy pronto, es probable que vean una campaña de marketing para esta cosa llamada Proyecto Harmony (que recientemente ha lanzado la versión 1.0 de sus plantillas de documentos). Incluso el nombre es marketing: no es realmente descriptivo, sino que trata de venderte una “buena onda” sobre el proyecto aun sin saber de qué se trata. (Además tuvo una seria³ colisión⁴ de nombres⁵,

¹<http://ur1.ca/2djpm>

²<http://creativecommons.org/licenses/by-sa/3.0/us/legalcode>

³<http://www.projectharmony.com/>

⁴<http://www.projectharmonynyc.org/>

⁵<http://www.harmony-project.org/>

Proyecto Harmony considerado dañino

—Bradley Kuhn—

Esta traducción y el artículo original se liberan bajo Creative Commons Atribución CompartirDerivadasIgual 3.0 de Argentina¹ y Estados Unidos² respectivamente. Traducido por fauno y mpj.

Mucha publicidad se diseña para convencernos de comprar o usar algo que no necesitamos. Cuando escucho a alguien zumbando sobre alguna cosa nueva, maravillosa, me preocupo pensando que esos tipos están tratando de venderme algo.

Muy pronto, es probable que vean una campaña de marketing para esta cosa llamada Proyecto Harmony (que recientemente ha lanzado la versión 1.0 de sus plantillas de documentos). Incluso el nombre es marketing: no es realmente descriptivo, sino que trata de venderte una “buena onda” sobre el proyecto aun sin saber de qué se trata. (Además tuvo una seria³ colisión⁴ de nombres⁵,

¹<http://ur1.ca/2djpm>

²<http://creativecommons.org/licenses/by-sa/3.0/us/legalcode>

³<http://www.projectharmony.com/>

⁴<http://www.projectharmonynyc.org/>

⁵<http://www.harmony-project.org/>

incluso⁶ con otro proyecto de software libre.)

El Proyecto Harmony se vende a sí mismo como reparación de algo que nuestra comunidad realmente no considera roto. El Proyecto Harmony es un juego de plantillas de documentos, principalmente promulgadas y diseñadas por abogados corporativos, que intentan seducir a los desarrolladores a entregar el control de su trabajo a las compañías.

Mi análisis a continuación trata principalmente sobre cómo estos acuerdos son problemáticos para los desarrolladores individuales. Un análisis de los acuerdos a la luz de las compañías u organizaciones que los usen entre sí puede o no tener las mismas conclusiones, pero no puedo asegurarlo todavía.

[A propósito, soy consciente que he fallado en proveer una versión TL;DR de este artículo. Intenté hacerlo dos veces y finalmente decidí que no puedo. Estos problemas son complejos y tenía que tomar una década de licenciamiento de software libre, políticas y conocimiento organizacional para articular completamente qué es lo que está mal con los acuerdos del Proyecto Harmony. Me doy cuenta de que suena a “fue difícil de escribir, debería ser difícil de leer”, pero no sé cómo resumir estos problemas gordianos. No obstante espero que los desarrolladores se tomen el tiempo de leer esto antes de firmar un acuerdo del Proyecto Harmony, o, de hecho, cualquier CLA o ©AA.]

Una cesión de copyright que carece de garantías reales

Antes que nada, casi la mitad del Proyecto Harmony se trata de acuerdos de asignación de copyright (©AA por sus siglas en inglés). La asignación de copyright cede completamente el trabajo a alguien más. Una vez que el ©AA está firmado, el trabajo deja de pertenecer al asignante. Es como si el trabajo en realidad

⁶<http://harmony.apache.org/>

incluso⁶ con otro proyecto de software libre.)

El Proyecto Harmony se vende a sí mismo como reparación de algo que nuestra comunidad realmente no considera roto. El Proyecto Harmony es un juego de plantillas de documentos, principalmente promulgadas y diseñadas por abogados corporativos, que intentan seducir a los desarrolladores a entregar el control de su trabajo a las compañías.

Mi análisis a continuación trata principalmente sobre cómo estos acuerdos son problemáticos para los desarrolladores individuales. Un análisis de los acuerdos a la luz de las compañías u organizaciones que los usen entre sí puede o no tener las mismas conclusiones, pero no puedo asegurarlo todavía.

[A propósito, soy consciente que he fallado en proveer una versión TL;DR de este artículo. Intenté hacerlo dos veces y finalmente decidí que no puedo. Estos problemas son complejos y tenía que tomar una década de licenciamiento de software libre, políticas y conocimiento organizacional para articular completamente qué es lo que está mal con los acuerdos del Proyecto Harmony. Me doy cuenta de que suena a “fue difícil de escribir, debería ser difícil de leer”, pero no sé cómo resumir estos problemas gordianos. No obstante espero que los desarrolladores se tomen el tiempo de leer esto antes de firmar un acuerdo del Proyecto Harmony, o, de hecho, cualquier CLA o ©AA.]

Una cesión de copyright que carece de garantías reales

Antes que nada, casi la mitad del Proyecto Harmony se trata de acuerdos de asignación de copyright (©AA por sus siglas en inglés). La asignación de copyright cede completamente el trabajo a alguien más. Una vez que el ©AA está firmado, el trabajo deja de pertenecer al asignante. Es como si el trabajo en realidad

⁶<http://harmony.apache.org/>

hubiera sido hecho por el asignado. Admitido, hay algún valor en la asignación de copyright, particularmente cuando los desarrolladores quieren asegurarse que la GPL u otra forma de copyleft es debidamente protegida en su trabajo pero no tienen el tiempo de hacerlo ellos mismos. (Aunque los desarrolladores también pueden designar un agente de ejecución que lo haga en su nombre sin tener que asignar el copyright, así que esa necesidad es limitada.)

Uno debe tener una confianza inmensa en la organización asignada. Personalmente, yo sólo he asignado algunos de mis copyright a una sola organización en toda mi vida: la Free Software Foundation⁷, porque la FSF es la única organización que he encontrado que está institucionalmente comprometida a hacer lo correcto con los copyrights de forma similar a mis creencias morales personales.

En primer lugar, como he escrito antes⁸, los ©AA de la FSF hacen toda suerte de promesas al asignante. En segundo, la FSF está institucionalmente comprometida con la GPL⁹ y en hacer cumplir la GPL de forma de avanzar la militancia sin fines de lucro de la FSF por la libertad del software. Toda esta actividad está contenida en mis principios morales, por lo que no he tenido problema en firmar los ©AA de la FSF.

Aún así, he conocido *muchos* desarrolladores que se niegan a firmar los ©AA de la FSF. Mientras muchos de ellos aprueban la GPL, no necesariamente acuerdan con las posiciones morales de la FSF. En efecto, en muchos casos, estos desarrolladores se oponen completamente a asignar el copyright a nadie, sea la FSF o cualquier otro. Por ejemplo, Linus Torvalds¹⁰, fundador de Linux, ha declarado repetidas veces que “nunca quis[o] hacer asignaciones de copyright, por varias razones: personalmente piens[a] que son sucias y erróneas, odiaría hacer todo el papeleo y piens[a] que actuaría como detractor del modelo de desarrollo”.

Obviamente, mi posición no es tan radical como la de Linus; pienso que los ©AA pueden ser apropiados en ciertas ocasiones.

⁷<http://fsf.org>

⁸<http://ebb.org/bkuhn/blog/2010/02/01/copyright-not-all-equal.html>

⁹<http://www.gnu.org/licenses/gpl.html>

¹⁰<http://groups.google.com/group/fa.linux.kernel/msg/b0587ac4dcb7a79b>

hubiera sido hecho por el asignado. Admitido, hay algún valor en la asignación de copyright, particularmente cuando los desarrolladores quieren asegurarse que la GPL u otra forma de copyleft es debidamente protegida en su trabajo pero no tienen el tiempo de hacerlo ellos mismos. (Aunque los desarrolladores también pueden designar un agente de ejecución que lo haga en su nombre sin tener que asignar el copyright, así que esa necesidad es limitada.)

Uno debe tener una confianza inmensa en la organización asignada. Personalmente, yo sólo he asignado algunos de mis copyright a una sola organización en toda mi vida: la Free Software Foundation⁷, porque la FSF es la única organización que he encontrado que está institucionalmente comprometida a hacer lo correcto con los copyrights de forma similar a mis creencias morales personales.

En primer lugar, como he escrito antes⁸, los ©AA de la FSF hacen toda suerte de promesas al asignante. En segundo, la FSF está institucionalmente comprometida con la GPL⁹ y en hacer cumplir la GPL de forma de avanzar la militancia sin fines de lucro de la FSF por la libertad del software. Toda esta actividad está contenida en mis principios morales, por lo que no he tenido problema en firmar los ©AA de la FSF.

Aún así, he conocido *muchos* desarrolladores que se niegan a firmar los ©AA de la FSF. Mientras muchos de ellos aprueban la GPL, no necesariamente acuerdan con las posiciones morales de la FSF. En efecto, en muchos casos, estos desarrolladores se oponen completamente a asignar el copyright a nadie, sea la FSF o cualquier otro. Por ejemplo, Linus Torvalds¹⁰, fundador de Linux, ha declarado repetidas veces que “nunca quis[o] hacer asignaciones de copyright, por varias razones: personalmente piens[a] que son sucias y erróneas, odiaría hacer todo el papeleo y piens[a] que actuaría como detractor del modelo de desarrollo”.

Obviamente, mi posición no es tan radical como la de Linus; pienso que los ©AA pueden ser apropiados en ciertas ocasiones.

⁷<http://fsf.org>

⁸<http://ebb.org/bkuhn/blog/2010/02/01/copyright-not-all-equal.html>

⁹<http://www.gnu.org/licenses/gpl.html>

¹⁰<http://groups.google.com/group/fa.linux.kernel/msg/b0587ac4dcb7a79b>

Pero también creo que los desarrolladores nunca deberían asignar su copyright a una compañía u organización cuya filosofía moral no cabe en la suya propia.

La FSF, por su parte, se expide sobre su posición moral en sus ©AA mismos. Como mencioné en otra parte¹¹, y como Groklaw¹² recientemente cubrió en detalle, los ©AA de la FSF realizan varias promesas con validez legal a los desarrolladores que las firman. Mientras, los ©AA del Proyecto Harmony ponen unas pocas opciones que parecen vagamente aceptables (aunque tienen problemas propios que discuto más abajo), no las hacen obligatorias. Yo mismo he señalado en varias ocasiones a los que escribieron los borradores de Harmony que los términos¹³ que la FSF ha propuesto deberían ser obligatorios en cualquier ©AA de una compañía con fines de lucro, pero se han negado a incorporar estas garantías como un requisito de sus acuerdos. (Noten que tales garantías también deben incluirse en las opciones de los CLA; ver debajo para más detalles.)

Por último me gustaría señalar que la FSF **no** requieren ©AAs para todos los paquetes GNU¹⁴. Esta confusión es tan común que me gustaría llamar la atención sobre ella, aunque sea un punto tangencial a este contexto. Los ©AA de la FSF son obligatorios, según mi conocimiento, si los paquetes GNU fueron (a) desarrollados por empleados de la FSF en sus primeras versiones o (b) los desarrolladores originales le *pidieron* a la FSF que realice una asignación de copyright cuando su proyecto se unió a GNU. En todos los demás casos, la asignación a la FSF es opcional. Algunos proyectos GNU, como GNOME¹⁵, tienen sus propias posiciones sobre los ©AAs que difieren radicalmente de las de la FSF. Dudo seriamente que las compañías que adopten los acuerdos del Proyecto Harmony lleguen a ser tan flexibles en cuanto a la asignación de copyright como lo es la FSF, o que alguna de las opciones posibles que el Proyecto Harmony provea sean aceptables para la política

¹¹<http://ebb.org/bkuhn/blog/2010/02/01/copyright-not-all-equal.html>

¹²<http://www.groklaw.net/articlebasic.php?story=20110524120303815>

¹³<http://www.fsf.org/blogs/rms/assigning-copyright>

¹⁴<http://www.gnu.org/help/evaluation.html>

¹⁵<http://live.gnome.org/CopyrightAssignment>

Pero también creo que los desarrolladores nunca deberían asignar su copyright a una compañía u organización cuya filosofía moral no cabe en la suya propia.

La FSF, por su parte, se expide sobre su posición moral en sus ©AA mismos. Como mencioné en otra parte¹¹, y como Groklaw¹² recientemente cubrió en detalle, los ©AA de la FSF realizan varias promesas con validez legal a los desarrolladores que las firman. Mientras, los ©AA del Proyecto Harmony ponen unas pocas opciones que parecen vagamente aceptables (aunque tienen problemas propios que discuto más abajo), no las hacen obligatorias. Yo mismo he señalado en varias ocasiones a los que escribieron los borradores de Harmony que los términos¹³ que la FSF ha propuesto deberían ser obligatorios en cualquier ©AA de una compañía con fines de lucro, pero se han negado a incorporar estas garantías como un requisito de sus acuerdos. (Noten que tales garantías también deben incluirse en las opciones de los CLA; ver debajo para más detalles.)

Por último me gustaría señalar que la FSF **no** requieren ©AAs para todos los paquetes GNU¹⁴. Esta confusión es tan común que me gustaría llamar la atención sobre ella, aunque sea un punto tangencial a este contexto. Los ©AA de la FSF son obligatorios, según mi conocimiento, si los paquetes GNU fueron (a) desarrollados por empleados de la FSF en sus primeras versiones o (b) los desarrolladores originales le *pidieron* a la FSF que realice una asignación de copyright cuando su proyecto se unió a GNU. En todos los demás casos, la asignación a la FSF es opcional. Algunos proyectos GNU, como GNOME¹⁵, tienen sus propias posiciones sobre los ©AAs que difieren radicalmente de las de la FSF. Dudo seriamente que las compañías que adopten los acuerdos del Proyecto Harmony lleguen a ser tan flexibles en cuanto a la asignación de copyright como lo es la FSF, o que alguna de las opciones posibles que el Proyecto Harmony provea sean aceptables para la política

¹¹<http://ebb.org/bkuhn/blog/2010/02/01/copyright-not-all-equal.html>

¹²<http://www.groklaw.net/articlebasic.php?story=20110524120303815>

¹³<http://www.fsf.org/blogs/rms/assigning-copyright>

¹⁴<http://www.gnu.org/help/evaluation.html>

¹⁵<http://live.gnome.org/CopyrightAssignment>

actual de GNOME.

¿Regalar tus derechos para que las compañías sientan mariposas en la panza?

El Proyecto Harmony, sin embargo, proclama que la parte importante no son los ©AAs, sino los Acuerdos de Licencia del Contribuidor (CLA por sus siglas en inglés). Para considerar brevemente la historia de los CLAs de Software Libre, hay que notar que el CLA de Apache¹⁶ fue probablemente el primer CLA usado en la comunidad del Software Libre. La Apache Software Foundation siempre ha estado fuertemente influenciada por IBM y otras compañías, y tales compañías generalmente han sentido “mariposas en la panza” al lograr que cada contribuidor asienta a firmar un documento legal complejo que afirma varias garantías respecto al código y da ciertos poderes a la compañía.

El punto principal de un CLA (y hasta cierto punto válido) es asegurar que los desarrolladores han verificado su derecho a contribuir código bajo la licencia de copyright del proyecto. Tanto el CLA de Apache como los CLA del Proyecto Harmony llegan a extremos de longitud y verborragia para requerir que los desarrolladores acuerden que saben que la contribución es suya. De hecho, si un desarrollador firma uno de estos CLAs, realiza un contrato formal con la entidad (usualmente una compañía con fines de lucro) en el que reconoce que la contribución se licencia bajo la licencia del proyecto. ¡Y entonces el desarrollador asume toda responsabilidad si ese hecho es incorrecto o se pone en disputa!

Por supuesto, mover toda la responsabilidad sobre los orígenes del código produce muchas “mariposas en la panza” a los abogados de la compañía. Esos abogados saben que ahora pueden fácilmente **demandar a un desarrollador individual** por incumplimiento de contrato si el desarrollador se equivocó en el código. Si la

¹⁶<http://www.apache.org/licenses/icla.txt>

actual de GNOME.

¿Regalar tus derechos para que las compañías sientan mariposas en la panza?

El Proyecto Harmony, sin embargo, proclama que la parte importante no son los ©AAs, sino los Acuerdos de Licencia del Contribuidor (CLA por sus siglas en inglés). Para considerar brevemente la historia de los CLAs de Software Libre, hay que notar que el CLA de Apache¹⁶ fue probablemente el primer CLA usado en la comunidad del Software Libre. La Apache Software Foundation siempre ha estado fuertemente influenciada por IBM y otras compañías, y tales compañías generalmente han sentido “mariposas en la panza” al lograr que cada contribuidor asienta a firmar un documento legal complejo que afirma varias garantías respecto al código y da ciertos poderes a la compañía.

El punto principal de un CLA (y hasta cierto punto válido) es asegurar que los desarrolladores han verificado su derecho a contribuir código bajo la licencia de copyright del proyecto. Tanto el CLA de Apache como los CLA del Proyecto Harmony llegan a extremos de longitud y verborragia para requerir que los desarrolladores acuerden que saben que la contribución es suya. De hecho, si un desarrollador firma uno de estos CLAs, realiza un contrato formal con la entidad (usualmente una compañía con fines de lucro) en el que reconoce que la contribución se licencia bajo la licencia del proyecto. ¡Y entonces el desarrollador asume toda responsabilidad si ese hecho es incorrecto o se pone en disputa!

Por supuesto, mover toda la responsabilidad sobre los orígenes del código produce muchas “mariposas en la panza” a los abogados de la compañía. Esos abogados saben que ahora pueden fácilmente **demandar a un desarrollador individual** por incumplimiento de contrato si el desarrollador se equivocó en el código. Si la

¹⁶<http://www.apache.org/licenses/icla.txt>

compañía distribuye el código de un desarrollador y termina recibiendo una demanda por infracción por la que paga millones de dólares, pueden fácilmente darse vuelta y demandar al desarrollador.¹⁷ La compañía argumentará en la corte que el desarrollador rompió el contrato. Si este desenlace posible no te preocupa *inmediatamente* en tanto desarrollador individual cuando firmás un CLA del Proyecto Harmony por tu contribución libre, debería.

La “Elección de Ley” y los Arreglos Contractuales embarran los reclamos de copyright

El CLA de Apache no posee una cláusula de elección de ley, lo que es preferible en mi opinión. La mayoría de los abogados *aman* una cláusula de “elección de ley” (“choice of law” en inglés) por varias razones. La más grande es que significa que las reglas que se aplican sobre el acuerdo son las más familiares a los abogados y la jurisdicción para las disputas será la jurisdicción local de la compañía, no la del desarrollador. Adicionalmente, los abogados a menudo eligen jurisdicciones particulares que son muy favorables a su cliente pero no lo son para los otros firmantes.

Desafortunadamente, *todos* los borradores del Proyecto Harmony incluyen una cláusula de “elección de ley”.¹⁸ Espero que

¹⁷Los defensores del Proyecto Harmony reclamarán que su sección 5, “Renuncia consecucional de daño”, protege adecuadamente a los desarrolladores. Noto que deja afuera explícitamente a, por ejemplo, los daños estatutarios de la infracción de copyright. Además, no puede renunciarse a algunos tipos de daños (y es por eso que esa sección le grita al lector “EN LA MEDIDA EN QUE LA LEY LO PERMITA”). Noten mi discusión sobre las jurisdicciones en el texto principal de este artículo, y consideren el hecho que el receptor de un CLA obviamente seleccionará la jurisdicción donde se pueda renunciar a la menor cantidad de daños. Finalmente, noten que la parte “O NOSOTROS” de esa sección 5 está disponible opcionalmente, y seguramente los abogados corporativos la usarán, lo que significa que si ellos violan el acuerdo, básicamente no tenés forma de recibir alguna renuncia de daño de parte de ellos, aun si prometieron mantener el código bajo copyleft.

¹⁸*Nota:* Versiones tempranas de este artículo mezclaron “elección de sede”

compañía distribuye el código de un desarrollador y termina recibiendo una demanda por infracción por la que paga millones de dólares, pueden fácilmente darse vuelta y demandar al desarrollador.¹⁷ La compañía argumentará en la corte que el desarrollador rompió el contrato. Si este desenlace posible no te preocupa *inmediatamente* en tanto desarrollador individual cuando firmás un CLA del Proyecto Harmony por tu contribución libre, debería.

La “Elección de Ley” y los Arreglos Contractuales embarran los reclamos de copyright

El CLA de Apache no posee una cláusula de elección de ley, lo que es preferible en mi opinión. La mayoría de los abogados *aman* una cláusula de “elección de ley” (“choice of law” en inglés) por varias razones. La más grande es que significa que las reglas que se aplican sobre el acuerdo son las más familiares a los abogados y la jurisdicción para las disputas será la jurisdicción local de la compañía, no la del desarrollador. Adicionalmente, los abogados a menudo eligen jurisdicciones particulares que son muy favorables a su cliente pero no lo son para los otros firmantes.

Desafortunadamente, *todos* los borradores del Proyecto Harmony incluyen una cláusula de “elección de ley”.¹⁸ Espero que

¹⁷Los defensores del Proyecto Harmony reclamarán que su sección 5, “Renuncia consecucional de daño”, protege adecuadamente a los desarrolladores. Noto que deja afuera explícitamente a, por ejemplo, los daños estatutarios de la infracción de copyright. Además, no puede renunciarse a algunos tipos de daños (y es por eso que esa sección le grita al lector “EN LA MEDIDA EN QUE LA LEY LO PERMITA”). Noten mi discusión sobre las jurisdicciones en el texto principal de este artículo, y consideren el hecho que el receptor de un CLA obviamente seleccionará la jurisdicción donde se pueda renunciar a la menor cantidad de daños. Finalmente, noten que la parte “O NOSOTROS” de esa sección 5 está disponible opcionalmente, y seguramente los abogados corporativos la usarán, lo que significa que si ellos violan el acuerdo, básicamente no tenés forma de recibir alguna renuncia de daño de parte de ellos, aun si prometieron mantener el código bajo copyleft.

¹⁸*Nota:* Versiones tempranas de este artículo mezclaron “elección de sede”

los que escriben los borradores argumentarán en respuesta que la jurisdicción es una variable de configuración. No obstante, el problema es que la *compañía* es la que decide el valor de esa variable, que casi siempre será la que no prefiera el desarrollador individual. Probablemente el término no sea negociable en ese punto, aunque haya podido configurarse en la plantilla.

Y no sólo eso, imaginá un escenario mucho más probable para ese CLA: la compañía falla en usar la licencia saliente que prometieron. Por ejemplo, supongan que prometieron a los desarrolladores que sería AGPL para siempre (¡aunque ninguna opción como esta existe en el Proyecto Harmony, vean más abajo!), pero entonces la compañía lanza versiones propietarias. Los desarrolladores que firmaron el CLA todavía detentan copyright, por lo que pueden ejercer sus derechos bajo la ley de copyright que, por sí misma, puede lograr que los desarrolladores hagan cumplir la licencia bajo la ley en cualquier jurisdicción que les parezca (asumiendo que la infracción sucede en esa, por supuesto).

Sin embargo, al firmar un CLA con la cláusula de “elección de ley”, los desarrolladores acordaron mantenerse bajo la jurisdicción establecida en el CLA. El CLA convierte lo que de otra forma hubiera sido una demanda de infracción de copyright mundana operando solamente bajo la reglamentación de copyright local al desarrollador en una disputa contractual entre los desarrolladores y la compañía bajo las leyes de la jurisdicción elegida. Obviamente ese acuerdo incluiría la AGPL y/o GPL por referencia, pero el reclamo por infracción de copyright hecho por violación de la GPL está ahora embarrado por el contrato CLA que los desarrolladores firmaron, y donde cedieron algunos derechos y permisos a la compañía que van más allá de la GPL.

Aún peor, si el desarrollador realiza la demanda en su propia jurisdicción, esa jurisdicción se ve forzada a interpretar las leyes de otro lugar. Esto lleva a resultados altamente variables y confusos.

con “elección de ley”. El fraseo ha sido aclarado para solucionar este problema. Por favor envíenme comentarios o correo si creen que no ha sido corregido adecuadamente.

los que escriben los borradores argumentarán en respuesta que la jurisdicción es una variable de configuración. No obstante, el problema es que la *compañía* es la que decide el valor de esa variable, que casi siempre será la que no prefiera el desarrollador individual. Probablemente el término no sea negociable en ese punto, aunque haya podido configurarse en la plantilla.

Y no sólo eso, imaginá un escenario mucho más probable para ese CLA: la compañía falla en usar la licencia saliente que prometieron. Por ejemplo, supongan que prometieron a los desarrolladores que sería AGPL para siempre (¡aunque ninguna opción como esta existe en el Proyecto Harmony, vean más abajo!), pero entonces la compañía lanza versiones propietarias. Los desarrolladores que firmaron el CLA todavía detentan copyright, por lo que pueden ejercer sus derechos bajo la ley de copyright que, por sí misma, puede lograr que los desarrolladores hagan cumplir la licencia bajo la ley en cualquier jurisdicción que les parezca (asumiendo que la infracción sucede en esa, por supuesto).

Sin embargo, al firmar un CLA con la cláusula de “elección de ley”, los desarrolladores acordaron mantenerse bajo la jurisdicción establecida en el CLA. El CLA convierte lo que de otra forma hubiera sido una demanda de infracción de copyright mundana operando solamente bajo la reglamentación de copyright local al desarrollador en una disputa contractual entre los desarrolladores y la compañía bajo las leyes de la jurisdicción elegida. Obviamente ese acuerdo incluiría la AGPL y/o GPL por referencia, pero el reclamo por infracción de copyright hecho por violación de la GPL está ahora embarrado por el contrato CLA que los desarrolladores firmaron, y donde cedieron algunos derechos y permisos a la compañía que van más allá de la GPL.

Aún peor, si el desarrollador realiza la demanda en su propia jurisdicción, esa jurisdicción se ve forzada a interpretar las leyes de otro lugar. Esto lleva a resultados altamente variables y confusos.

con “elección de ley”. El fraseo ha sido aclarado para solucionar este problema. Por favor envíenme comentarios o correo si creen que no ha sido corregido adecuadamente.

Problemas para hacer cumplir el copyright individualmente contra terceras partes

Además, aún cuando los desarrolladores individuales retienen sus copyrights, los CLAs del Proyecto Harmony garantizan muchos derechos y permisos transferibles al receptor del CLA (otra vez, usualmente a una compañía). Aún *si* las razones para requerirlos fuesen nobles, introducen un manojito de permisos extra que pueden ser pasados a otras entidades.

De repente, lo que una vez fuera una simple demanda para hacer cumplir el copyright hecha por un desarrollador que descubre una violación del copyleft se convierte en una pregunta: “¿Recibió de alguna manera esa entidad violatoria permisos especiales de esta otra entidad colectora del CLA?” Los violadores van a moverse rápidamente a este tipo de defensa. Aunque la defensa pueda no tener mérito (por ejemplo, el receptor del CLA ni siquiera conoce al violador), introduce confusión. La mayoría de los procedimientos legales que involucran software ya son suficientemente confusos para las cortes debido a la compleja tecnología involucrada. Agregar algo como esto sólo causará problemas y demoras, agregando más peso a nuestros mínimamente financiados esfuerzos por hacer cumplir el copyleft de la comunidad.

Problemas para hacer cumplir el copyright individualmente contra terceras partes

Además, aún cuando los desarrolladores individuales retienen sus copyrights, los CLAs del Proyecto Harmony garantizan muchos derechos y permisos transferibles al receptor del CLA (otra vez, usualmente a una compañía). Aún *si* las razones para requerirlos fuesen nobles, introducen un manojito de permisos extra que pueden ser pasados a otras entidades.

De repente, lo que una vez fuera una simple demanda para hacer cumplir el copyright hecha por un desarrollador que descubre una violación del copyleft se convierte en una pregunta: “¿Recibió de alguna manera esa entidad violatoria permisos especiales de esta otra entidad colectora del CLA?” Los violadores van a moverse rápidamente a este tipo de defensa. Aunque la defensa pueda no tener mérito (por ejemplo, el receptor del CLA ni siquiera conoce al violador), introduce confusión. La mayoría de los procedimientos legales que involucran software ya son suficientemente confusos para las cortes debido a la compleja tecnología involucrada. Agregar algo como esto sólo causará problemas y demoras, agregando más peso a nuestros mínimamente financiados esfuerzos por hacer cumplir el copyleft de la comunidad.



Entrante=Saliente es todo lo que necesitas

Mientras tanto, la pregunta sobre el CLA es esta única consideración fundamental: ¿Necesitamos esto? La respuesta del Proyecto Harmony es clara: sus defensores claman que existe una confusión masiva sobre los CLAs y ninguna estandarización, por lo que el Proyecto Harmony debe proveer un set estándar de acuerdos que encarnen todas las opciones usadas típicamente.

Aun más, el Proyecto Harmony ha rechazado a propósito ofrecer la opción más simple y popular de todas, que mi colega Richard Fontana (un abogado de Red Hat que también se opone¹⁹ al Proyecto Harmony) ha llamado²⁰ el año pasado “entrante=saliente”²¹. Específicamente, el acuerdo por defecto en la abrumadora mayoría de los proyectos FLOSS es simplemente esta: cada contribuidor acuerda licenciar cada contribución bajo la licencia de copyright específica del proyecto (o una licencia compatible).

No importa de qué lado mires al Proyecto Harmony, los otros problemas contractuales descritos arriba vuelven imposible un verdadero entrante=saliente porque el receptor del CLA nunca está legalmente obligado por la licencia del proyecto. Mientras tanto, y aún bajo su mejor configuración, el Proyecto Harmony no puede aproximarse adecuadamente a entrante=saliente. Específicamente, el Proyecto Harmony intenta limitar el licenciamiento de saliente en su sección 2.3 (llamada “Licencia saliente”). Sin embargo, todas las versiones copyleft de esta plantilla incluyen una cláusula que dice: “Nosotros [los receptores del CLA] acordamos licenciar la Contribución [...] bajo los términos de las [...] licencias que Nosotros estemos utilizando a la Fecha de Envío del Material”. Pero *no hay forma* de que el contribuyente verifique con seguridad cuáles licencias usa en privado la entidad que recibe el CLA. Si esa entidad ya tiene un modelo de negocio de relicenciamiento propie-

¹⁹<http://opensource.com/law/11/7/trouble-harmony-part-1>

²⁰<http://identi.ca/conversation/45589896>

²¹<http://ref.fedorapeople.org/fontana-linuxcon.html>

Entrante=Saliente es todo lo que necesitas

Mientras tanto, la pregunta sobre el CLA es esta única consideración fundamental: ¿Necesitamos esto? La respuesta del Proyecto Harmony es clara: sus defensores claman que existe una confusión masiva sobre los CLAs y ninguna estandarización, por lo que el Proyecto Harmony debe proveer un set estándar de acuerdos que encarnen todas las opciones usadas típicamente.

Aun más, el Proyecto Harmony ha rechazado a propósito ofrecer la opción más simple y popular de todas, que mi colega Richard Fontana (un abogado de Red Hat que también se opone¹⁹ al Proyecto Harmony) ha llamado²⁰ el año pasado “entrante=saliente”²¹. Específicamente, el acuerdo por defecto en la abrumadora mayoría de los proyectos FLOSS es simplemente esta: cada contribuidor acuerda licenciar cada contribución bajo la licencia de copyright específica del proyecto (o una licencia compatible).

No importa de qué lado mires al Proyecto Harmony, los otros problemas contractuales descritos arriba vuelven imposible un verdadero entrante=saliente porque el receptor del CLA nunca está legalmente obligado por la licencia del proyecto. Mientras tanto, y aún bajo su mejor configuración, el Proyecto Harmony no puede aproximarse adecuadamente a entrante=saliente. Específicamente, el Proyecto Harmony intenta limitar el licenciamiento de saliente en su sección 2.3 (llamada “Licencia saliente”). Sin embargo, todas las versiones copyleft de esta plantilla incluyen una cláusula que dice: “Nosotros [los receptores del CLA] acordamos licenciar la Contribución [...] bajo los términos de las [...] licencias que Nosotros estemos utilizando a la Fecha de Envío del Material”. Pero *no hay forma* de que el contribuyente verifique con seguridad cuáles licencias usa en privado la entidad que recibe el CLA. Si esa entidad ya tiene un modelo de negocio de relicenciamiento propie-

¹⁹<http://opensource.com/law/11/7/trouble-harmony-part-1>

²⁰<http://identi.ca/conversation/45589896>

²¹<http://ref.fedorapeople.org/fontana-linuxcon.html>

tario²² a la Fecha de Envío, entonces el contribuyente concede su permiso para tal relicenciamiento en esa contribución nueva, aún si el resto de la sección 2.3 le promete copyleft. Esto no es hipotético: ha habido muchos casos donde no está claro si la compañía estaba o no estaba involucrada en relicenciamiento propietario, para más tarde descubrir que lo habían estado haciendo en privado durante años. Tal como está escrita, por lo tanto, **cualquier configuración de la sección 2.3 del Proyecto Harmony es inútil para prevenir la apropiación propietaria.**

Aunque ese bug sea arreglado, lo más cercano que el Proyecto Harmony va a estar de entrante=saliente es restringiendo sus CLA a “la lista de la FSF de licencias copyleft recomendadas”. Sin embargo, esta categoría no hace ninguna distinción entre la AGPL²³ y la GPL, y en última instancia le permite a la FSF el poder de relicenciar (ya que la FSF puede cambiar²⁴ su lista de copyleft recomendadas a voluntad). Si los contribuyentes son serios sobre la AGPL, entonces el Proyecto Harmony **no** puede asegurar que sus cambios permanezcan bajo esta licencia. Aún más, los contribuyentes *deben* confiar en la FSF a perpetuidad, aún *más de lo que actualmente necesitan* en las opciones “... o subsiguientes” de las licencias de la FSF en existencia. Estoy de acuerdo en confiar en la FSF en la mayoría de los casos. Sin embargo, ya que prefiero sólo AGPLv3 o subsiguientes para mi código, el Proyecto Harmony es completamente incapaz de acomodarse a mis preferencias y aproximarse a un entrante=saliente que sea AGPL (aún si ignoro los numerosos problemas ya discutidos).

Mientras tanto, la normal, mundana y extendida práctica entrante=saliente es simple, efectiva y no mezcla sus complicadas disputas contractuales y estructuras de control con la gobernanza del proyecto. En esencia, para la mayoría de los proyectos FLOSS, la licencia de copyright del proyecto sirve como su Constitución y no le mezcla ninguna otra complicación. El Proyecto Harmony busca darle maripositas a los abogados a expensas de tirarles el fardo legal y molesto a los desarrolladores.

²²<http://ebb.org/bkuhn/blog/2009/10/16/open-core-shareware.html>

²³<http://www.gnu.org/licenses/agpl.html>

²⁴<http://www.gnu.org/licenses/recommended-copylefts.html>

tario²² a la Fecha de Envío, entonces el contribuyente concede su permiso para tal relicenciamiento en esa contribución nueva, aún si el resto de la sección 2.3 le promete copyleft. Esto no es hipotético: ha habido muchos casos donde no está claro si la compañía estaba o no estaba involucrada en relicenciamiento propietario, para más tarde descubrir que lo habían estado haciendo en privado durante años. Tal como está escrita, por lo tanto, **cualquier configuración de la sección 2.3 del Proyecto Harmony es inútil para prevenir la apropiación propietaria.**

Aunque ese bug sea arreglado, lo más cercano que el Proyecto Harmony va a estar de entrante=saliente es restringiendo sus CLA a “la lista de la FSF de licencias copyleft recomendadas”. Sin embargo, esta categoría no hace ninguna distinción entre la AGPL²³ y la GPL, y en última instancia le permite a la FSF el poder de relicenciar (ya que la FSF puede cambiar²⁴ su lista de copyleft recomendadas a voluntad). Si los contribuyentes son serios sobre la AGPL, entonces el Proyecto Harmony **no** puede asegurar que sus cambios permanezcan bajo esta licencia. Aún más, los contribuyentes *deben* confiar en la FSF a perpetuidad, aún *más de lo que actualmente necesitan* en las opciones “... o subsiguientes” de las licencias de la FSF en existencia. Estoy de acuerdo en confiar en la FSF en la mayoría de los casos. Sin embargo, ya que prefiero sólo AGPLv3 o subsiguientes para mi código, el Proyecto Harmony es completamente incapaz de acomodarse a mis preferencias y aproximarse a un entrante=saliente que sea AGPL (aún si ignoro los numerosos problemas ya discutidos).

Mientras tanto, la normal, mundana y extendida práctica entrante=saliente es simple, efectiva y no mezcla sus complicadas disputas contractuales y estructuras de control con la gobernanza del proyecto. En esencia, para la mayoría de los proyectos FLOSS, la licencia de copyright del proyecto sirve como su Constitución y no le mezcla ninguna otra complicación. El Proyecto Harmony busca darle maripositas a los abogados a expensas de tirarles el fardo legal y molesto a los desarrolladores.

²²<http://ebb.org/bkuhn/blog/2009/10/16/open-core-shareware.html>

²³<http://www.gnu.org/licenses/agpl.html>

²⁴<http://www.gnu.org/licenses/recommended-copylefts.html>

Los Hackers de Linux han fijado ingeniosamente entrante=saliente

Hace casi 10 años atrás, recuerdo haber asistido a un sesión del USENIX 2001 Linux BoF²⁵. En esa sesión, Ted T'so²⁶ y yo tuvimos un acalorado debate; yo afirmé que el ©AA de la FSF aseguraba certeza legal sobre el código GNU, pero que Linux no poseía tal seguro. (Por cierto, incluso *yo* estaba confundido en esos días y pensaba que todos los paquetes de GNU requerían un ©AA de la FSF.) Ted explicó, en la manera clara y brillante habitual, que tales métodos duros no eran necesarios para darle certeza legal a la GPL y que la comunidad de Linux quería encontrar una alternativa.

Me fui sacudiendo la cabeza escépticamente. Recuerdo haber pensado: “Ted no entiende”. Pero estaba equivocado; él *sí* entendía. De hecho, muchos de los desarrolladores clave de Linux lo hacían. Tres años después de esa conversación pública con Ted, el Certificado de Origen del Desarrollador²⁷ (DCO por sus siglas en inglés) se convirtió en la forma oficial de manejar el “problema de los CLA” en Linux y sigue siendo la política oficial²⁸ al día de hoy. (Ver el ítem 12 en el archivo Documentation/SubmittingPatches de Linux.)

El DCO, en efecto, ¡es el único CLA que necesita cualquier proyecto FLOSS! Implementa entrante=saliente en una forma simple y directa, sin darle poderes especiales a ninguna compañía o entidad en particular. Los desarrolladores mantienen su propio copyright y unilateralmente atestiguan su derecho a contribuir y la licencia de su contribución. (Además pueden firmar un ©AA con alguna otra entidad, como la FSF, si quieren.) El DCO también provee una metodología simple (es decir, la etiqueta Signed-off-by:) para realizar ese atestiguamiento.

²⁵<http://www.usenix.org/event/usenix01/bofschedule.html>

²⁶<http://thunk.org/tytso>

²⁷<http://permalink.gmane.org/gmane.linux.kernel.commits.head/33254>

²⁸<http://www.kernel.org/doc/Documentation/SubmittingPatches>

Los Hackers de Linux han fijado ingeniosamente entrante=saliente

Hace casi 10 años atrás, recuerdo haber asistido a un sesión del USENIX 2001 Linux BoF²⁵. En esa sesión, Ted T'so²⁶ y yo tuvimos un acalorado debate; yo afirmé que el ©AA de la FSF aseguraba certeza legal sobre el código GNU, pero que Linux no poseía tal seguro. (Por cierto, incluso *yo* estaba confundido en esos días y pensaba que todos los paquetes de GNU requerían un ©AA de la FSF.) Ted explicó, en la manera clara y brillante habitual, que tales métodos duros no eran necesarios para darle certeza legal a la GPL y que la comunidad de Linux quería encontrar una alternativa.

Me fui sacudiendo la cabeza escépticamente. Recuerdo haber pensado: “Ted no entiende”. Pero estaba equivocado; él *sí* entendía. De hecho, muchos de los desarrolladores clave de Linux lo hacían. Tres años después de esa conversación pública con Ted, el Certificado de Origen del Desarrollador²⁷ (DCO por sus siglas en inglés) se convirtió en la forma oficial de manejar el “problema de los CLA” en Linux y sigue siendo la política oficial²⁸ al día de hoy. (Ver el ítem 12 en el archivo Documentation/SubmittingPatches de Linux.)

El DCO, en efecto, ¡es el único CLA que necesita cualquier proyecto FLOSS! Implementa entrante=saliente en una forma simple y directa, sin darle poderes especiales a ninguna compañía o entidad en particular. Los desarrolladores mantienen su propio copyright y unilateralmente atestiguan su derecho a contribuir y la licencia de su contribución. (Además pueden firmar un ©AA con alguna otra entidad, como la FSF, si quieren.) El DCO también provee una metodología simple (es decir, la etiqueta Signed-off-by:) para realizar ese atestiguamiento.

²⁵<http://www.usenix.org/event/usenix01/bofschedule.html>

²⁶<http://thunk.org/tytso>

²⁷<http://permalink.gmane.org/gmane.linux.kernel.commits.head/33254>

²⁸<http://www.kernel.org/doc/Documentation/SubmittingPatches>

Debo admitir que me he burlado de la simplicidad (que consideraba naïve) del DCO en comparación al ©AA de la FSF. Desde entonces me he convencido que el DCO de Linux cumple claramente con su trabajo principal a la vez que se ajusta a la forma de trabajo preferida por muchos desarrolladores. Los ©AA tienen su lugar, particularmente cuando los desarrolladores encuentran una organización confiable que se alinea con su código moral personal y se encargarán de defender el copyleft por ellos. No obstante, para los CLAs, el DCO de Linux cumple con este importante trabajo y deja de lado el costado pro-corporativo inútil.

Francamente, si tengo que elegir entre hacer las cosas fáciles para los desarrolladores o hacerlas fáciles para los abogados corporativos, voy a elegir lo primero en todos los casos: los desarrolladores escriben el código; mientras que la mayor parte del tiempo, los departamentos legales de una compañía sólo se meten en el medio. La comunidad de FLOSS necesita cubrirse el culo lo suficiente como para zafar; el DCO muestra lo que es realmente *necesario*, en oposición a lo que los abogados corporativos *desean* que hagan sus desarrolladores.

¿Qué pasa con el relicenciamiento?

El DCO de Linux no permite que una sola entidad relicencie el código; esta es la razón por la que el cambio a GPLv3 en Linux será una ardua tarea de procesos públicos por conseguir el permiso para el cambio. Sin embargo, hay que notar que la cultura linuxera *crea* que la GPLv2 es el fundamento moral y el principio de su comunidad. No es un principio con el que concuerdo; la mayoría de mis lectores saben que mi licencia preferida²⁹ es la AGPLv3 o superior. Pero este es el punto: entrante=saliente es *la* forma en que una comunidad FLOSS implementa su moralidad; el Proyecto Harmony busca remover las decisiones comunitarias sobre el licenciamiento de casi todos los proyectos.

²⁹<http://ebb.org/bkuhn/blog/2011/05/26/choose.html>

Debo admitir que me he burlado de la simplicidad (que consideraba naïve) del DCO en comparación al ©AA de la FSF. Desde entonces me he convencido que el DCO de Linux cumple claramente con su trabajo principal a la vez que se ajusta a la forma de trabajo preferida por muchos desarrolladores. Los ©AA tienen su lugar, particularmente cuando los desarrolladores encuentran una organización confiable que se alinea con su código moral personal y se encargarán de defender el copyleft por ellos. No obstante, para los CLAs, el DCO de Linux cumple con este importante trabajo y deja de lado el costado pro-corporativo inútil.

Francamente, si tengo que elegir entre hacer las cosas fáciles para los desarrolladores o hacerlas fáciles para los abogados corporativos, voy a elegir lo primero en todos los casos: los desarrolladores escriben el código; mientras que la mayor parte del tiempo, los departamentos legales de una compañía sólo se meten en el medio. La comunidad de FLOSS necesita cubrirse el culo lo suficiente como para zafar; el DCO muestra lo que es realmente *necesario*, en oposición a lo que los abogados corporativos *desean* que hagan sus desarrolladores.

¿Qué pasa con el relicenciamiento?

El DCO de Linux no permite que una sola entidad relicencie el código; esta es la razón por la que el cambio a GPLv3 en Linux será una ardua tarea de procesos públicos por conseguir el permiso para el cambio. Sin embargo, hay que notar que la cultura linuxera *crea* que la GPLv2 es el fundamento moral y el principio de su comunidad. No es un principio con el que concuerdo; la mayoría de mis lectores saben que mi licencia preferida²⁹ es la AGPLv3 o superior. Pero este es el punto: entrante=saliente es *la* forma en que una comunidad FLOSS implementa su moralidad; el Proyecto Harmony busca remover las decisiones comunitarias sobre el licenciamiento de casi todos los proyectos.

²⁹<http://ebb.org/bkuhn/blog/2011/05/26/choose.html>

Estoy a favor de permitir el relicenciamiento “o superior”; la GPL, la LGPL y la AGPL han dejado la opción a la comunidad ya que la GPLv1³⁰ fue publicada a finales de los '80. Los proyectos se declaran “GPLv2 o superior” o “LGPL o superior”, o incluso “GPLv1 o superior o Artística”³¹ (como Perl 5) para identificar su cultura y permisos de relicenciamiento. Aunque a veces sería bueno tener una autoridad en relicenciamiento, el precio es muy caro: el abandono de una claridad comunitaria con respecto a los términos en que se define su cultura de desarrollo.

¿Una parcialidad anti copyleft fuerte?

Lo que es aun peor, es que el Proyecto Harmony se mantiene parcial contra algunas de las más afinadas versiones de la cultura copyleft. Por ejemplo, Allison Randal³², quien está muy involucrada en el Proyecto Harmony, argumentó en el episodio 204³³ de Linux Outlaws que “la mayoría de los desarrolladores que contribuyen bajo una licencia copyleft, estarían contentos de hacerlo bajo cualquier otra, sea AGPL, GPL o LGPL”. Sin embargo existen varias razones bien definidas³⁴ de por qué los desarrolladores elegirían la GPL antes que la LGPL. Por eso, entregarle a una compañía con fines de lucro (o una sin fines de lucro que no necesariamente comparta los valores de los desarrolladores) el poder unilateral de tomar decisiones de relicenciamiento y convertir un proyecto bajo GPL en LGPL o cualquier otra licencia de copyleft débil es ridículo.

En su lanzamiento 1.0, el Proyecto Harmony intentó agregar una opción de “sólo copyleft fuerte”. Por supuesto que no funciona, por las razones discutidas más arriba. Pero aun así, esta solución es sólo una de las muchas y no es requerida por defecto cuando un proyecto ya es copyleft.

³⁰<http://www.gnu.org/licenses/gpl-1.0.txt>

³¹<http://dev.perl.org/licenses/>

³²<http://ebb.org/bkuhn/blog/2011/06/26/identica-weekly.html>

³³<http://linuxoutlaws.com/podcast/ogg/204>

³⁴<http://www.gnu.org/philosophy/why-not-lgpl.html>

Estoy a favor de permitir el relicenciamiento “o superior”; la GPL, la LGPL y la AGPL han dejado la opción a la comunidad ya que la GPLv1³⁰ fue publicada a finales de los '80. Los proyectos se declaran “GPLv2 o superior” o “LGPL o superior”, o incluso “GPLv1 o superior o Artística”³¹ (como Perl 5) para identificar su cultura y permisos de relicenciamiento. Aunque a veces sería bueno tener una autoridad en relicenciamiento, el precio es muy caro: el abandono de una claridad comunitaria con respecto a los términos en que se define su cultura de desarrollo.

¿Una parcialidad anti copyleft fuerte?

Lo que es aun peor, es que el Proyecto Harmony se mantiene parcial contra algunas de las más afinadas versiones de la cultura copyleft. Por ejemplo, Allison Randal³², quien está muy involucrada en el Proyecto Harmony, argumentó en el episodio 204³³ de Linux Outlaws que “la mayoría de los desarrolladores que contribuyen bajo una licencia copyleft, estarían contentos de hacerlo bajo cualquier otra, sea AGPL, GPL o LGPL”. Sin embargo existen varias razones bien definidas³⁴ de por qué los desarrolladores elegirían la GPL antes que la LGPL. Por eso, entregarle a una compañía con fines de lucro (o una sin fines de lucro que no necesariamente comparta los valores de los desarrolladores) el poder unilateral de tomar decisiones de relicenciamiento y convertir un proyecto bajo GPL en LGPL o cualquier otra licencia de copyleft débil es ridículo.

En su lanzamiento 1.0, el Proyecto Harmony intentó agregar una opción de “sólo copyleft fuerte”. Por supuesto que no funciona, por las razones discutidas más arriba. Pero aun así, esta solución es sólo una de las muchas y no es requerida por defecto cuando un proyecto ya es copyleft.

³⁰<http://www.gnu.org/licenses/gpl-1.0.txt>

³¹<http://dev.perl.org/licenses/>

³²<http://ebb.org/bkuhn/blog/2011/06/26/identica-weekly.html>

³³<http://linuxoutlaws.com/podcast/ogg/204>

³⁴<http://www.gnu.org/philosophy/why-not-lgpl.html>

Por último, es importante notar que las GPLv3, AGPLv3 y LGPLv3³⁵ ya ofrecen una “opción de apoderado (proxy)”;

los proyectos pueden nombrar a alguien para decidir cambiar a “o superior” más tarde. Por eso, para aquellos proyectos que estén usando cualquiera del grupo “sólo LGPLv3”, “sólo AGPLv3”, “sólo GPLv3”, “GPLv2 o superior”, “GPLv1 o superior” o “LGPLv2.1 o superior”, sus desarrolladores *ya poseen* mecanismos para moverse a versiones superiores de la licencia con facilidad al especificar un apoderado. No hay necesidad de un CLA para cumplir tal tarea en la familia de licencias GPL, a menos que el objetivo sea erosionar los copylefts fuertes para convertirlos en copylefts débiles.

Esto no es Creative Commons, pero si lo fuera, ¿vale la pena emularlo?

Los defensores del Proyecto Harmony aman compararlo con Creative Commons³⁶, pero la comparación no es particularmente apta. Aún más, no estoy convencido de la que la comunidad FLOSS deba emular al grupo de licencias de CC del todo, ya que algunos aspectos de la estructura de CC son problemáticos al migrarlos al licenciamiento FLOSS.

En primer lugar, Larry Lessig³⁷ (quien es ampliamente considerado como un visionario) inició el licenciamiento CC para capturar un movimiento por la Cultura Libre que se estaba moldeando a semejanza del movimiento del Software Libre (la cual pasó una década estudiando). Sin embargo, Lessig realizó varios compromisos morales en su intento de construir un puente a la mentalidad “algunos derechos reservados”. Así, muchas de las licencias CC - notablemente las que incluyen las cláusulas NoComercial (NC) o SinDerivadas (ND)- son consideradas demasiado restrictivas y son

³⁵<http://www.gnu.org/licenses/gpl.html#section14>

³⁶<http://creativecommons.org/>

³⁷http://en.wikipedia.org/wiki/Lawrence_Lessig

Por último, es importante notar que las GPLv3, AGPLv3 y LGPLv3³⁵ ya ofrecen una “opción de apoderado (proxy)”;

los proyectos pueden nombrar a alguien para decidir cambiar a “o superior” más tarde. Por eso, para aquellos proyectos que estén usando cualquiera del grupo “sólo LGPLv3”, “sólo AGPLv3”, “sólo GPLv3”, “GPLv2 o superior”, “GPLv1 o superior” o “LGPLv2.1 o superior”, sus desarrolladores *ya poseen* mecanismos para moverse a versiones superiores de la licencia con facilidad al especificar un apoderado. No hay necesidad de un CLA para cumplir tal tarea en la familia de licencias GPL, a menos que el objetivo sea erosionar los copylefts fuertes para convertirlos en copylefts débiles.

Esto no es Creative Commons, pero si lo fuera, ¿vale la pena emularlo?

Los defensores del Proyecto Harmony aman compararlo con Creative Commons³⁶, pero la comparación no es particularmente apta. Aún más, no estoy convencido de la que la comunidad FLOSS deba emular al grupo de licencias de CC del todo, ya que algunos aspectos de la estructura de CC son problemáticos al migrarlos al licenciamiento FLOSS.

En primer lugar, Larry Lessig³⁷ (quien es ampliamente considerado como un visionario) inició el licenciamiento CC para capturar un movimiento por la Cultura Libre que se estaba moldeando a semejanza del movimiento del Software Libre (la cual pasó una década estudiando). Sin embargo, Lessig realizó varios compromisos morales en su intento de construir un puente a la mentalidad “algunos derechos reservados”. Así, muchas de las licencias CC - notablemente las que incluyen las cláusulas NoComercial (NC) o SinDerivadas (ND)- son consideradas demasiado restrictivas y son

³⁵<http://www.gnu.org/licenses/gpl.html#section14>

³⁶<http://creativecommons.org/>

³⁷http://en.wikipedia.org/wiki/Lawrence_Lessig

por lo tanto rechazadas tanto por activistas de la Cultura Libre³⁸ como por militantes del Software Libre³⁹.

Después de una década, esos militantes empezaron lentamente a convencer a los titulares de copyright de evitar las opciones NC y ND de CC, pero la promulgación continua de estas opciones por parte de CC deslegitiman esos intentos. Así, CC y el Proyecto Harmony cometen el mismo error: actúan amoralmente en su intento de construir una estructura de licencias/acuerdos que trata de aunar entendimiento entre una comunidad FaiF (“libre como en libertad”, en inglés “Free as in Freedom”) y aquellos que recién están comenzando a conocerla. Elegí la palabra amoral, como usualmente hago⁴⁰, para denotar una situación donde existen importantes principios morales, pero sus actores principales buscan remover la moralidad de consideración con la excusa de dejar el poder de decisión “a la magia del mercado”. El Proyecto Harmony repite el error de las licencias CC que la comunidad de la Cultura Libre ha pasado una década (y contando) de limpiar.

Conclusiones

Por favor noten que no soy un abogado y que esto no es un aviso legal. Sólo soy un desarrollador tanto individual como comunitario enfocado en políticas de software libre que tiene serias preocupaciones por la forma en que operan los acuerdos del Proyecto Harmony. No puedo proveer un análisis legal refinado, porque francamente soy un amateur cuando se trata de leyes, pero *soy* un experto en políticas de libertad del software. En esa vena -sin tener en cuenta el apoyo de los abogados corporativos- mi opinión es que el Proyecto Harmony debe abandonarse por completo.

De hecho, la distinción entre experticia política y legal mues-

³⁸<http://blog.ninapaley.com/2011/07/04/rantifesto/>

³⁹http://en.wikipedia.org/wiki/Creative_Commons#Other_criticism_of_the_non-commercial_license

⁴⁰<http://ebb.org/bkuhn/blog/2010/06/23/open-source.html#footnote-amoral-word-choice>

por lo tanto rechazadas tanto por activistas de la Cultura Libre³⁸ como por militantes del Software Libre³⁹.

Después de una década, esos militantes empezaron lentamente a convencer a los titulares de copyright de evitar las opciones NC y ND de CC, pero la promulgación continua de estas opciones por parte de CC deslegitiman esos intentos. Así, CC y el Proyecto Harmony cometen el mismo error: actúan amoralmente en su intento de construir una estructura de licencias/acuerdos que trata de aunar entendimiento entre una comunidad FaiF (“libre como en libertad”, en inglés “Free as in Freedom”) y aquellos que recién están comenzando a conocerla. Elegí la palabra amoral, como usualmente hago⁴⁰, para denotar una situación donde existen importantes principios morales, pero sus actores principales buscan remover la moralidad de consideración con la excusa de dejar el poder de decisión “a la magia del mercado”. El Proyecto Harmony repite el error de las licencias CC que la comunidad de la Cultura Libre ha pasado una década (y contando) de limpiar.

Conclusiones

Por favor noten que no soy un abogado y que esto no es un aviso legal. Sólo soy un desarrollador tanto individual como comunitario enfocado en políticas de software libre que tiene serias preocupaciones por la forma en que operan los acuerdos del Proyecto Harmony. No puedo proveer un análisis legal refinado, porque francamente soy un amateur cuando se trata de leyes, pero *soy* un experto en políticas de libertad del software. En esa vena -sin tener en cuenta el apoyo de los abogados corporativos- mi opinión es que el Proyecto Harmony debe abandonarse por completo.

De hecho, la distinción entre experticia política y legal mues-

³⁸<http://blog.ninapaley.com/2011/07/04/rantifesto/>

³⁹http://en.wikipedia.org/wiki/Creative_Commons#Other_criticism_of_the_non-commercial_license

⁴⁰<http://ebb.org/bkuhn/blog/2010/06/23/open-source.html#footnote-amoral-word-choice>

tra la raíz del problema del Proyecto Harmony. Es un sistema de documentos diseñados por un comité compuesto principalmente por abogados corporativos, aun cuando se muestra como un consenso de desarrolladores FLOSS. En efecto, el Proyecto Harmony fue iniciado por Amanda Brock⁴¹, una abogada corporativa con fines de lucro de Canonical, Ltd, que aun trabaja en los borradores. Más tarde, Canonical, Ltd.⁴² contrató a Mark Radcliffe (un abogado que ha defendido⁴³ violadores de la GPL⁴⁴) para escribir el borrador de las versiones alpha del documento, y aun continúa haciéndolo. Aún más, el proceso primario de escritura de esos borradores se hizo en secreto en reuniones cerradas dominadas por abogados corporativos hasta que los documentos estuvieron casi completos; el proceso no se hizo público a la comunidad hasta abril del 2011. La versión 1.0 de los documentos poco difiere de los borradores que se lanzaron en abril, y por lo tanto se mantienen como documentos que fueron principalmente redactados en secreto por abogados corporativos que sólo tienen una lejana familiaridad con la cultura del software libre.

He preguntado⁴⁵ muchas veces a los defensores del Proyecto Harmony quién está a cargo del mismo hoy, y nadie puede darme una respuesta directa. Uno se pregunta quién toma las decisiones finales y cuál es el proceso que existe para incluir o excluir texto en los borradores. El proceso que una vez fue secreto ahora parece ser caótico porque fue abierto mucho más tarde de lo necesario para resolver sus problemas fundamentales.

Sólo unos pocos desarrolladores están involucrados en el proyecto. Pero el Proyecto Harmony no es algo que los desarrolladores hayan pedido; fue iniciado por compañías que *quisieran* convencerlos de adoptar pasivamente estos extralimitados CLAs y ©AAs. Para mí, el proceso completo del Proyecto Harmony se siente co-

⁴¹<http://www.linkedin.com/pub/dir/Amanda/Brock>

⁴²<http://identi.ca/notice/74444380>

⁴³<http://www.archive.org/download/gov.uscourts.nysd.327540/gov.uscourts.nysd.327540.3.0.pdf>

⁴⁴<http://sec.gov/Archives/edgar/data/1375365/000119312509084731/file name1.htm>

⁴⁵<http://identi.ca/conversation/74175630#notice-76902928>

tra la raíz del problema del Proyecto Harmony. Es un sistema de documentos diseñados por un comité compuesto principalmente por abogados corporativos, aun cuando se muestra como un consenso de desarrolladores FLOSS. En efecto, el Proyecto Harmony fue iniciado por Amanda Brock⁴¹, una abogada corporativa con fines de lucro de Canonical, Ltd, que aun trabaja en los borradores. Más tarde, Canonical, Ltd.⁴² contrató a Mark Radcliffe (un abogado que ha defendido⁴³ violadores de la GPL⁴⁴) para escribir el borrador de las versiones alpha del documento, y aun continúa haciéndolo. Aún más, el proceso primario de escritura de esos borradores se hizo en secreto en reuniones cerradas dominadas por abogados corporativos hasta que los documentos estuvieron casi completos; el proceso no se hizo público a la comunidad hasta abril del 2011. La versión 1.0 de los documentos poco difiere de los borradores que se lanzaron en abril, y por lo tanto se mantienen como documentos que fueron principalmente redactados en secreto por abogados corporativos que sólo tienen una lejana familiaridad con la cultura del software libre.

He preguntado⁴⁵ muchas veces a los defensores del Proyecto Harmony quién está a cargo del mismo hoy, y nadie puede darme una respuesta directa. Uno se pregunta quién toma las decisiones finales y cuál es el proceso que existe para incluir o excluir texto en los borradores. El proceso que una vez fue secreto ahora parece ser caótico porque fue abierto mucho más tarde de lo necesario para resolver sus problemas fundamentales.

Sólo unos pocos desarrolladores están involucrados en el proyecto. Pero el Proyecto Harmony no es algo que los desarrolladores hayan pedido; fue iniciado por compañías que *quisieran* convencerlos de adoptar pasivamente estos extralimitados CLAs y ©AAs. Para mí, el proceso completo del Proyecto Harmony se siente co-

⁴¹<http://www.linkedin.com/pub/dir/Amanda/Brock>

⁴²<http://identi.ca/notice/74444380>

⁴³<http://www.archive.org/download/gov.uscourts.nysd.327540/gov.uscourts.nysd.327540.3.0.pdf>

⁴⁴<http://sec.gov/Archives/edgar/data/1375365/000119312509084731/file name1.htm>

⁴⁵<http://identi.ca/conversation/74175630#notice-76902928>

mo una guerra de desgaste para convencer a los desarrolladores de aceptar algo que no necesariamente quieren con un mínimo de disenso. En definitiva, la necesidad del Proyecto Harmony no se ha articulado para los desarrolladores.

Por último, pregunto, ¿qué es lo que está realmente roto? La industria ha estado adoptando GNU y Linux durante años de manera continua. GNU, por su parte, tiene las asignaciones de la FSF para muchos de sus proyectos tempranos, pero los últimos proyectos (GNOME⁴⁶ en particular) han sido absolutamente contrarios tanto al uso de ©AAs como de CLAs, o se han mostrado indiferentes y han usado entrante=saliente. Linux, por su lado, usa el DCO, que realiza el trabajo de manejar las partes más urgentes e importantes de un CLA sin ponerse en el camino de los desarrolladores y sin forzarles riesgos legales extra y entregarle las decisiones de licenciamiento (incluyendo las debilitantes del copyleft) a una única entidad (usualmente con fines de lucro).

En definitiva, el Proyecto Harmony es una solución mal diseñada buscando un problema.

Véase también (en inglés)

- El problema con Harmony, Parte I⁴⁷ de Richard Fontana
- Los acuerdos Harmony 1.0⁴⁸ de Dave Neary
- Algunos pensamientos sobre la asignación de copyright⁴⁹ de Michael Meek
- La asignación de copyright y otras barreras para ingresar⁵⁰ de Dave Neary

⁴⁶<http://live.gnome.org/CopyrightAssignment/Guidelines>

⁴⁷<http://opensource.com/law/11/7/trouble-harmony-part-1>

⁴⁸<http://blogs.gnome.org/bolsh/2011/07/06/harmony-agreements-reach-1-0/>

⁴⁹<http://people.gnome.org/~michael/blog/copyright-assignment.html>

⁵⁰<http://blogs.gnome.org/bolsh/2009/04/08/copyright-assignment-and-other-barriers-to-entry/>

mo una guerra de desgaste para convencer a los desarrolladores de aceptar algo que no necesariamente quieren con un mínimo de disenso. En definitiva, la necesidad del Proyecto Harmony no se ha articulado para los desarrolladores.

Por último, pregunto, ¿qué es lo que está realmente roto? La industria ha estado adoptando GNU y Linux durante años de manera continua. GNU, por su parte, tiene las asignaciones de la FSF para muchos de sus proyectos tempranos, pero los últimos proyectos (GNOME⁴⁶ en particular) han sido absolutamente contrarios tanto al uso de ©AAs como de CLAs, o se han mostrado indiferentes y han usado entrante=saliente. Linux, por su lado, usa el DCO, que realiza el trabajo de manejar las partes más urgentes e importantes de un CLA sin ponerse en el camino de los desarrolladores y sin forzarles riesgos legales extra y entregarle las decisiones de licenciamiento (incluyendo las debilitantes del copyleft) a una única entidad (usualmente con fines de lucro).

En definitiva, el Proyecto Harmony es una solución mal diseñada buscando un problema.

Véase también (en inglés)

- El problema con Harmony, Parte I⁴⁷ de Richard Fontana
- Los acuerdos Harmony 1.0⁴⁸ de Dave Neary
- Algunos pensamientos sobre la asignación de copyright⁴⁹ de Michael Meek
- La asignación de copyright y otras barreras para ingresar⁵⁰ de Dave Neary

⁴⁶<http://live.gnome.org/CopyrightAssignment/Guidelines>

⁴⁷<http://opensource.com/law/11/7/trouble-harmony-part-1>

⁴⁸<http://blogs.gnome.org/bolsh/2011/07/06/harmony-agreements-reach-1-0/>

⁴⁹<http://people.gnome.org/~michael/blog/copyright-assignment.html>

⁵⁰<http://blogs.gnome.org/bolsh/2009/04/08/copyright-assignment-and-other-barriers-to-entry/>

-
- El relicenciamiento propietario es el nuevo shareware⁵¹ por mí mismo
 - Cuando una compañía te pide tu copyright⁵² de RMS
 - La FSF y el Proyecto Harmony⁵³ de Brett Smith
 - Hay⁵⁴ varios⁵⁵ hilos⁵⁶ diferentes⁵⁷ que⁵⁸ pueden⁵⁹ encontrarse⁶⁰ en⁶¹ identi.ca⁶² donde⁶³ se⁶⁴ discuten⁶⁵ los⁶⁶ acuerdos⁶⁷ del⁶⁸ Proyecto⁶⁹ Harmony⁷⁰. El hashtag “#Harmony”⁷¹ se usa a menudo. El hashtag “#CLA”⁷² también puede ser de interés.
 - El problema de traer armonía a la asignación de copyright⁷³ de Jos Poortvliet
 - Balanceando transparencia y privacidad⁷⁴ de Simon Phipp
 - Política de asignación de copyright de GNOME⁷⁵

⁵¹<http://ebb.org/bkuhn/blog/2009/10/16/open-core-shareware.html>

⁵²<http://www.fsf.org/blogs/rms/assigning-copyright/>

⁵³<http://www.fsf.org/blogs/licensing/project-harmony>

⁵⁴<http://identi.ca/conversation/60847947>

⁵⁵<http://identi.ca/conversation/60848873>

⁵⁶<http://identi.ca/conversation/61240820>

⁵⁷<http://identi.ca/conversation/68596018>

⁵⁸<http://identi.ca/conversation/68858735>

⁵⁹<http://identi.ca/conversation/68886640>

⁶⁰<http://identi.ca/conversation/68887235>

⁶¹<http://identi.ca/conversation/69308469>

⁶²<http://identi.ca/conversation/70389379>

⁶³<http://identi.ca/conversation/70648339>

⁶⁴<http://identi.ca/conversation/71854529>

⁶⁵<http://identi.ca/conversation/72024908>

⁶⁶<http://identi.ca/conversation/73129548>

⁶⁷<http://identi.ca/conversation/73225057>

⁶⁸<http://identi.ca/conversation/73225057>

⁶⁹<http://identi.ca/conversation/74175630>

⁷⁰<http://identi.ca/conversation/74979814>

⁷¹<http://identi.ca/tag/harmony>

⁷²<http://identi.ca/tag/cla>

⁷³<http://www.linuxuser.co.uk/news/the-issue-of-bringing-harmony-to-copyright-assignment/>

⁷⁴<http://opensource.com/life/11/4/balancing-transparency-and-privacy>

⁷⁵<http://live.gnome.org/CopyrightAssignment>

-
- El relicenciamiento propietario es el nuevo shareware⁵¹ por mí mismo
 - Cuando una compañía te pide tu copyright⁵² de RMS
 - La FSF y el Proyecto Harmony⁵³ de Brett Smith
 - Hay⁵⁴ varios⁵⁵ hilos⁵⁶ diferentes⁵⁷ que⁵⁸ pueden⁵⁹ encontrarse⁶⁰ en⁶¹ identi.ca⁶² donde⁶³ se⁶⁴ discuten⁶⁵ los⁶⁶ acuerdos⁶⁷ del⁶⁸ Proyecto⁶⁹ Harmony⁷⁰. El hashtag “#Harmony”⁷¹ se usa a menudo. El hashtag “#CLA”⁷² también puede ser de interés.
 - El problema de traer armonía a la asignación de copyright⁷³ de Jos Poortvliet
 - Balanceando transparencia y privacidad⁷⁴ de Simon Phipp
 - Política de asignación de copyright de GNOME⁷⁵

⁵¹<http://ebb.org/bkuhn/blog/2009/10/16/open-core-shareware.html>

⁵²<http://www.fsf.org/blogs/rms/assigning-copyright/>

⁵³<http://www.fsf.org/blogs/licensing/project-harmony>

⁵⁴<http://identi.ca/conversation/60847947>

⁵⁵<http://identi.ca/conversation/60848873>

⁵⁶<http://identi.ca/conversation/61240820>

⁵⁷<http://identi.ca/conversation/68596018>

⁵⁸<http://identi.ca/conversation/68858735>

⁵⁹<http://identi.ca/conversation/68886640>

⁶⁰<http://identi.ca/conversation/68887235>

⁶¹<http://identi.ca/conversation/69308469>

⁶²<http://identi.ca/conversation/70389379>

⁶³<http://identi.ca/conversation/70648339>

⁶⁴<http://identi.ca/conversation/71854529>

⁶⁵<http://identi.ca/conversation/72024908>

⁶⁶<http://identi.ca/conversation/73129548>

⁶⁷<http://identi.ca/conversation/73225057>

⁶⁸<http://identi.ca/conversation/73225057>

⁶⁹<http://identi.ca/conversation/74175630>

⁷⁰<http://identi.ca/conversation/74979814>

⁷¹<http://identi.ca/tag/harmony>

⁷²<http://identi.ca/tag/cla>

⁷³<http://www.linuxuser.co.uk/news/the-issue-of-bringing-harmony-to-copyright-assignment/>

⁷⁴<http://opensource.com/life/11/4/balancing-transparency-and-privacy>

⁷⁵<http://live.gnome.org/CopyrightAssignment>

- Los lineamientos de la Fundación GNOME sobre la asignación de copyright⁷⁶
- El Proyecto Harmony busca mejorar los acuerdos de contribución⁷⁷ de Amanda Brock
- Los archivos de la lista de correo del Proyecto Harmony⁷⁸
- Los borradores de los Acuerdos Harmony⁷⁹
- Políticas de contribución de proyectos de código abierto⁸⁰ diapositivas de la charla de Richard Fontana.

⁷⁶<http://live.gnome.org/CopyrightAssignment/Guidelines>

⁷⁷<http://opensource.com/law/10/6/project-harmony-looks-improve-contribution-agreements-0>

⁷⁸<http://lists.harmonyagreements.org/mailman/listinfo>

⁷⁹<http://harmonyagreements.org/agreements.html>

⁸⁰<http://ref.fedorapeople.org/fontana-linuxcon.html>

- Los lineamientos de la Fundación GNOME sobre la asignación de copyright⁷⁶
- El Proyecto Harmony busca mejorar los acuerdos de contribución⁷⁷ de Amanda Brock
- Los archivos de la lista de correo del Proyecto Harmony⁷⁸
- Los borradores de los Acuerdos Harmony⁷⁹
- Políticas de contribución de proyectos de código abierto⁸⁰ diapositivas de la charla de Richard Fontana.

⁷⁶<http://live.gnome.org/CopyrightAssignment/Guidelines>

⁷⁷<http://opensource.com/law/10/6/project-harmony-looks-improve-contribution-agreements-0>

⁷⁸<http://lists.harmonyagreements.org/mailman/listinfo>

⁷⁹<http://harmonyagreements.org/agreements.html>

⁸⁰<http://ref.fedorapeople.org/fontana-linuxcon.html>