

Esta es una traducción de la página original en inglés.

## El manifiesto de GNU

El manifiesto de GNU, que aparece a continuación, lo escribió Richard Stallman en 1985 para solicitar apoyo en el desarrollo del sistema operativo GNU. Parte del texto está tomado del anuncio original de 1983. A lo largo de 1987 se hicieron actualizaciones menores para dar cuenta de la evolución del proyecto; a partir de entonces, parece mejor dejarlo tal cual.

Con el transcurso del tiempo hemos aprendido que ciertos malentendidos comunes podían evitarse con una redacción diferente. Las aclaraciones al pie de página que hemos añadido a partir de 1993 ayudan a clarificar estos puntos.

Si desea instalar el sistema GNU/Linux, le recomendamos utilizar una de las distribuciones GNU/Linux que son 100% software libre. Para colaborar, consulte <http://www.gnu.org/help>.

El proyecto GNU forma parte del Movimiento del Software Libre, una campaña en favor de la libertad de los usuarios de software. Es un error asociar GNU con el «código abierto». Esta expresión fue acuñada en 1998 por gente que discrepaba de los valores éticos del Movimiento del Software Libre. Lo emplean para promover un punto de vista amoral de abordar el mismo asunto.

### ¿Qué es GNU? ¡GNU No es Unix!

GNU, que significa «Gnu No es Unix», es el nombre del sistema de software completamente compatible con Unix que estoy escribiendo para entregarlo libremente a todas las personas que puedan utilizarlo (1). Algunos voluntarios me están ayudando. Las aportaciones de tiempo, dinero, programas y equipos son muy necesarias.

Hasta el momento tenemos un editor de texto, Emacs con Lisp, para escribir comandos de edición, un depurador de código fuente, un generador parser compatible con yacc, un enlazador y alrededor de treinta y cinco utilidades. Una shell (intérprete de comandos) que está casi terminada. Un nuevo compilador portable y optimizador de C se autocompiló y posiblemente lo publicaremos este año. Existe un núcleo inicial, pero se necesitan muchas más características para emular a Unix. Cuando el núcleo y el compilador estén completos, será posible distribuir un sistema GNU apropiado para el desarrollo de programas. Usaremos el formateador de documentos TeX, pero también estamos trabajando en una versión de nroff. Usaremos también el sistema libre y portable de ventanas X. Después de esto agregaremos un Common Lisp portable, un juego Imperio, una hoja de cálculo y cientos de otras cosas, además de la documentación en línea. Esperamos proporcionar, con el tiempo, todas las utilidades que vienen normalmente con un sistema Unix y más.

GNU podrá ejecutar programas de Unix, pero no será idéntico a Unix. Haremos todas las mejoras que sean convenientes, en base a nuestra experiencia con otros sistemas operativos. Concretamente, planeamos tener nombres de archivos más largos, números para las versiones de los archivos, un sistema de archivo a prueba de caídas y tal vez incorporemos un sistema para completar los nombres de archivos, un soporte de visualización independiente del terminal y, quizá en el futuro, un sistema de ventanas basado en Lisp a través del cual varios programas Lisp y programas comunes de Unix puedan compartir una pantalla. Tanto C como Lisp estarán disponibles como lenguajes de programación del sistema. Intentaremos también dar soporte a UUCP, MIT Chaosnet y protocolos de Internet para las comunicaciones.

GNU está orientado inicialmente a las máquinas de la clase 68000/16000 con memoria virtual, porque son las máquinas donde es más sencilla su ejecución. El esfuerzo adicional para hacerlo funcionar en máquinas más pequeñas se lo dejaremos a quienes quieran utilizarlo en ellas.

Para evitar una horrible confusión, por favor pronuncie la *g* en la palabra «GNU» cuando se refiera al nombre de

este proyecto <sup>[1]</sup>.

## **Por qué debo escribir GNU**

Considero que la Regla de Oro me exige que si me gusta un programa lo debo compartir con otras personas a quienes también les guste. Los vendedores de software quieren dividir a los usuarios y dominarlos para llevarlos a aceptar no compartir su software con los demás. Me rehúso a romper la solidaridad con otros usuarios de esta manera. Mi conciencia me impide firmar un acuerdo de confidencialidad o un acuerdo de licencia de software. Durante años trabajé en el Laboratorio de Inteligencia Artificial oponiéndome a estas tendencias y otras descortesías, pero al final fueron demasiado lejos: no podía permanecer en una institución donde tales cosas se hicieran en mi nombre en contra de mi voluntad.

Para poder continuar a utilizar las computadoras sin deshonra, he decidido agrupar un conjunto suficiente de software libre para poder vivir sin usar ningún software que no sea libre. He renunciado al Laboratorio de Inteligencia Artificial para evitar que el MIT pueda usar alguna excusa legal que me impida distribuir software de GNU (2).

## **Por qué GNU será compatible con Unix**

Unix no es mi sistema ideal, pero no es tan malo. Las características esenciales de Unix parecen ser buenas, y pienso que puedo añadir lo que le falta sin echarlas a perder. Y un sistema compatible con Unix facilitará su adopción por parte de muchas otras personas.

## **Cómo estará disponible GNU**

GNU no está en el dominio público. Todos tendrán permiso para modificar y redistribuir GNU, pero a ningún distribuidor se le permitirá restringir su redistribución posterior. Es decir, no se autorizarán modificaciones privativas. Quiero asegurarme que todas las versiones de GNU permanezcan libres.

## **Por qué muchos programadores quieren colaborar**

He encontrado muchos programadores que están entusiasmados con GNU y quieren colaborar.

Muchos programadores están descontentos con la comercialización del software de sistema. Puede permitirles ganar más dinero, pero los hace sentirse en conflicto con otros programadores en lugar de sentirse como compañeros. El fundamento de la amistad entre programadores es el compartir programas, pero los acuerdos de mercadotecnia que los programadores suelen utilizar básicamente prohíben tratar a los demás como amigos. El comprador de software debe escoger entre la amistad y la obediencia a la ley. Naturalmente, muchos deciden que la amistad es más importante. Pero aquellos que creen en la ley a menudo no se sienten a gusto con ninguna de las opciones. Se vuelven cínicos y piensan que la programación es sólo una manera de ganar dinero.

Al desarrollar y utilizar GNU en lugar de programas privativos, podemos ser hospitalarios con todos y obedecer la ley. Además, GNU sirve como ejemplo de inspiración y como bandera para animar a otros a unirse a nosotros en el compartir. Esto puede darnos una sensación de armonía que es imposible obtener cuando utilizamos software que no es libre. Porque para cerca de la mitad de los programadores con quienes hablo, esto es un motivo de felicidad importante que el dinero no puede reemplazar.

## **Cómo colaborar**

Hoy en día, para conocer las tareas en las que puede colaborar en el ámbito del software, consulte la lista de proyectos prioritarios y la lista se busca ayuda, la lista general de tareas para paquetes software de GNU. Para colaborar de otras formas, consulte la guía para colaborar con el Proyecto

## GNU.

Pido a los fabricantes de ordenadores que donen máquinas y dinero. A los individuos les pido donaciones en forma de programas y trabajo.

Una de las consecuencias que puede esperar si dona máquinas es que GNU se ejecutará en ellas con anticipación. Las máquinas deben estar completas, listas para utilizar sistemas, aprobadas para su uso en una zona residencial y no requerir ventilación o fuentes de energía sofisticadas.

He encontrado muchos programadores ansiosos de contribuir trabajando a tiempo parcial para GNU. Para la mayoría de los proyectos, tal trabajo distribuido a tiempo parcial sería muy difícil de coordinar, las partes escritas de forma independiente no funcionarían correctamente unidas. Pero para la tarea particular de reemplazar Unix, este problema no existe. Un sistema completo Unix contiene cientos de programas de utilidades, cada uno de los cuales se documenta por separado. La mayoría de las especificaciones de interfaz se fijan por compatibilidad con Unix. Si cada colaborador puede escribir un reemplazo compatible para una sola utilidad Unix, y hacer que funcione correctamente en lugar del original en un sistema Unix, entonces estas utilidades funcionarán correctamente cuando se ensamblen. Aun teniendo en cuenta las leyes de Murphy acerca de algunos problemas inesperados, el montaje de estos componentes será una tarea factible (el núcleo requerirá una comunicación más estrecha y deberá trabajarse en un grupo pequeño y compacto).

Si obtengo más donaciones, podría contratar a algunas personas a tiempo completo o parcial. El sueldo no será alto para los estándares de los programadores, pero estoy buscando a gente para quien la construcción de un espíritu comunitarios sea tan importante como ganar dinero. Lo veo como una forma de permitir que estas personas se dediquen con todas sus energías a trabajar en GNU ahorrándoles la necesidad de ganarse la vida de otra manera.

### **Por qué se beneficiarán todos los usuarios de computadoras.**

Una vez que GNU esté terminado, todo el mundo podrá obtener un buen sistema de software tan libre como el aire. (3)

Esto significa mucho más que ahorrarse el dinero para pagar una licencia Unix. Significa evitar el derroche inútil de la duplicación de esfuerzos en la programación de sistemas. Este esfuerzo se puede invertir en cambio en el avance de la tecnología.

El código fuente del sistema completo estará disponible para todos. Como resultado, un usuario que necesite cambios en el sistema siempre será libre de hacerlo él mismo, o contratar a cualquier programador o empresa disponible para que los haga. Los usuarios ya no estarán a merced de un programador o empresa propietaria de las fuentes y que sea la única que puede realizar modificaciones.

Las escuelas podrán ofrecer un entorno mucho más educativo, y alentar a todos los alumnos a estudiar y mejorar el código. El laboratorio de computación de Harvard solía tener la política de que ningún programa podía ser instalado en el sistema si no se publicaba previamente su código fuente, llegando al punto de negarse a instalar ciertos programas. Yo me inspiré mucho en esa política.

Por último, el lastre de considerar quién es dueño de qué sistema de software y de lo que está o no está permitido hacer con él, habrá desaparecido.

Los acuerdos que obligan a la gente a pagar por usar un programa, incluyendo el licenciamiento de las copias, siempre incurren en un costo enorme para la sociedad a través de los mecanismos engorrosos necesarios para calcular la cantidad (es decir, qué programas) una persona debe pagar. Y sólo un estado policial puede forzar a todos a obedecer. Considere la posibilidad de una estación espacial en donde el aire debe fabricarse con un gran costo: cobrar a cada persona por litro de aire puede ser justo, pero usar una máscara para medir el aire durante todo el día y toda la noche es insoportable, incluso si todo el mundo puede permitirse el lujo de pagar la factura

del aire. Y las cámaras de video en todas partes para ver si alguna vez alguien se quita la máscara son indignantes. Es mejor apoyar a la planta de aire con un impuesto y desechar las máscaras.

Copiar todo o parte de un programa es tan natural para un programador como respirar, además es productivo. Debería ser libre.

## **Algunas objeciones, fácilmente rebatibles, a los objetivos de GNU**

**«Nadie lo va a usar si es libre, porque eso significa que no cuenta con ningún tipo de asistencia».**

**«Hay que cobrar por el programa para pagar por el servicio de asistencia».**

Si la gente prefiere pagar por GNU más el servicio en lugar de recibir GNU sin servicio, una empresa que preste solamente el servicio a las personas que hayan obtenido GNU debe ser rentable. (4)

Debemos distinguir entre el soporte en forma de trabajo de programación real y lo que es simplemente guiar al usuario. El primero es algo que uno no puede confiar a un proveedor de software. Si su problema no es compartido por bastante gente, el vendedor no se preocupará en solucionarlo.

Si su empresa necesita poder contar con soporte, la única manera es tener todo el código fuente y las herramientas necesarias. Entonces puede contratar a cualquier persona disponible para corregir el problema, y no estará a merced de ningún individuo. Con Unix, el precio del código fuente deja fuera de consideración a la mayoría de las empresas. Con GNU esto será sencillo. Puede ser que no haya ninguna persona competente disponible, pero por este problema no se puede culpar a los acuerdos de distribución. GNU no elimina todos los problemas del mundo, sólo algunos de ellos.

Mientras tanto, los usuarios que no saben nada acerca de las computadoras necesitan que los guíen: hacer cosas que podrían hacer por sí mismos fácilmente, pero no saben cómo.

Estos servicios podrán ser prestados por empresas que vendan solamente el servicio de asesoría y reparación. Si bien es cierto que los usuarios prefieren gastar dinero y obtener un producto con el servicio, también estarán dispuestos a adquirir el servicio habiendo obtenido el producto en forma gratuita. Las empresas de servicios competirán en calidad y precio, los usuarios no estarán atados a ninguna en particular. Mientras tanto, aquellos de nosotros que no necesitamos el servicio deberíamos tener la posibilidad de utilizar el programa sin tener que pagar por el servicio.

**«No se puede llegar a muchas personas sin publicidad, y para financiarla es necesario cobrar por el programa».**

**«No tiene sentido publicitar un programa que la gente puede obtener gratuitamente».**

Hay diversas formas de publicidad gratuita o muy barata que se puede utilizar para informar a los usuarios de computadoras acerca de algo como GNU. Pero quizás sea cierto que uno puede llegar a más usuarios de microcomputadoras con publicidad. Si esto es realmente así, un negocio que publicite el servicio pago de copiado y envío por correo del software de GNU debería ser lo suficientemente exitoso como para pagar por su publicidad y mucho más. De esta manera, solo los usuarios que se benefician de esta publicidad la pagarán.

Por otro lado, si mucha gente consigue GNU de sus amigos, y esas empresas no tienen éxito, esto demostrará que la publicidad no era realmente necesaria para difundir GNU. ¿Por qué es que los defensores del libre mercado no quieren dejar que el libre mercado lo decida? (5)

**«Mi compañía necesita un sistema operativo privativo para tener una ventaja competitiva».**

GNU quitará el software de sistema operativo del entorno de la competencia. No podrá obtener una ventaja en esta área, pero tampoco la competencia podrá tenerla frente a usted. Ambos competirán en otras áreas, mientras

se benefician mutuamente en esta. Si su negocio consiste en vender un sistema operativo, no le gustará GNU, pero ese es su problema. Si su negocio es de otro ámbito, GNU puede salvarlo de ser empujado dentro del costoso negocio de la venta de sistemas operativos.

Me gustaría ver que el desarrollo de GNU se mantuviera gracias a donaciones de algunos fabricantes y usuarios, reduciendo el coste para todos. (6)

### **«¿No merecen los programadores una recompensa por su creatividad?»**

Si hay algo que merece una recompensa, es la contribución social. La creatividad puede ser una contribución social, pero solo en la medida en que la sociedad sea libre de aprovechar los resultados. Si los programadores merecen ser recompensados por la creación de programas innovadores, entonces, por la misma razón merecen ser castigados si restringen el uso de estos programas.

### **«¿No debería un programador poder pedir una recompensa por su creatividad?»**

No hay nada malo en querer un pago por el trabajo o en buscar maximizar los ingresos personales, siempre y cuando no se utilicen medios que sean destructivos. Pero los medios habituales en el campo del software hoy en día se basan en la destrucción.

Extraer dinero de los usuarios de un programa limitando su uso es destructivo porque las restricciones reducen la cantidad y las formas en que el programa puede ser utilizado. Esto reduce la cantidad de beneficios que la humanidad obtiene del programa. Cuando hay una elección deliberada de restringir, las consecuencias dañinas son una destrucción deliberada.

La razón por la que un buen ciudadano no utiliza estos medios destructivos para volverse más rico es que si todos lo hicieran, podríamos empobrecernos todos por una mutua destrucción. Esto es ética kantiana, o la Regla de Oro. Como no me gustan las consecuencias que resultarían si todos acapararan información, debo considerar como erróneo que alguien lo haga. Específicamente, el deseo de ser recompensado por la creatividad de uno no justifica privar al mundo en general de toda o parte de esa creatividad.

### **«¿No se morirán de hambre los programadores?»**

Podría responder que nadie está obligado a ser programador. La mayoría de nosotros no puede conseguir dinero parándose en la calle y haciendo muecas. No estamos, por consiguiente, condenados a pasar nuestras vidas de pie en la calle haciendo muecas, y muriéndonos de hambre. Podemos dedicarnos a otra cosa.

Sin embargo, esta es una respuesta errónea porque acepta la suposición implícita del interrogador: que sin la propiedad del software a los programadores no se les puede pagar un centavo. En este supuesto es todo o nada.

La verdadera razón por la que los programadores no se morirán de hambre es porque aún es posible que se les pague por programar, solo que no se les pagará tanto como en la actualidad.

Restringir la copia no es la única forma para hacer negocios con el software. Es la forma más común (7) porque es con la que se obtiene más dinero. Si se prohibiera o fuese rechazada por el comprador, el negocio del software se desplazaría hacia otras formas de organización que actualmente no se usan tan a menudo. Siempre existen muchos modos para organizar cualquier tipo de negocio.

Probablemente la programación no será tan lucrativa bajo esta nueva forma como lo es actualmente. Pero esto no es un argumento en contra del cambio. No se considera una injusticia que los empleados en los comercios obtengan los salarios que ganan actualmente. Si los programadores ganaran lo mismo, no será tampoco una injusticia (en la práctica ganarán considerablemente más).

### **«¿La gente no tiene derecho a controlar cómo se usa su creatividad?»**

El «control del uso de las ideas de alguien» realmente constituye el control de las vidas de otras personas, y por lo general se utiliza para hacerles la vida más difícil.

Las personas que han estudiado cuidadosamente el tema de los derechos de propiedad intelectual<sup>(8)</sup> (por ejemplo los abogados) dicen que no existe un derecho intrínseco a la propiedad intelectual. Los tipos de los supuestos derechos de propiedad intelectual que reconoce el gobierno fueron creados mediante actos legislativos específicos con fines específicos.

Por ejemplo, el sistema de patentes se estableció para animar a los inventores a revelar los detalles de sus inventos. El objetivo era ayudar a la sociedad más que a los inventores. El periodo de validez de diecisiete años para una patente era corto comparado con el ritmo de desarrollo de la técnica. Dado que las patentes solo son relevantes para los fabricantes, para quienes el costo y el esfuerzo de un acuerdo de licencia son pequeños comparados con la puesta en marcha de la producción, las patentes a menudo no hacen mucho daño. No representan un obstáculo para la mayoría de los individuos que usan productos patentados.

La idea del copyright no existía en tiempos antiguos, cuando los autores frecuentemente copiaban extensivamente a otros autores en obras de no ficción. Esta práctica era útil, y ha sido la única forma de que las obras de muchos autores, aunque solo sea en parte, hayan sobrevivido. El sistema de copyright se creó expresamente con el propósito de promover la autoría. En el ámbito para el que se inventó —libros, que sólo podían copiarse de forma económica en una imprenta— hacía muy poco daño y no obstruía a la mayor parte de los individuos que leían los libros.

Todos los derechos de propiedad intelectual son solamente licencias otorgadas por la sociedad porque se pensaba, con razón o sin ella, que la sociedad en su conjunto se beneficiaría de su concesión. Pero, en cada situación particular, tenemos que preguntarnos: ¿nos beneficia realmente otorgar esta licencia? ¿qué tipo de acto le estamos permitiendo hacer a una persona?

El caso de los actuales programas es muy diferente al de los libros de hace cien años. El hecho de que la forma más sencilla de copiar un programa sea de un vecino a otro, el hecho de que un programa esté formado tanto por el código fuente como el código objeto, siempre distintos, y el hecho de que el programa se use en lugar de leerlo y disfrutarlo, se combinan para crear una situación en la que una persona que hace valer un copyright está dañando a la sociedad en su conjunto tanto materialmente como espiritualmente; nadie debería hacerlo a pesar de que la ley se lo permita.

### **«La competición hace que las cosas se hagan mejor».**

El paradigma de la competencia es una carrera: al premiar al ganador, estamos alentando a todos a correr más rápido. Cuando el capitalismo realmente funciona de esta manera, hace un buen trabajo; pero sus partidarios están equivocados al suponer que siempre funciona así. Si los corredores olvidan por qué se otorga el premio y se centran en ganar sin importar cómo, pueden encontrar otras estrategias, como atacar a los otros corredores. Si los corredores se enredan en una pelea a puñetazos, todos llegarán tarde a la meta.

El software privativo y secreto es el equivalente moral de los corredores en una pelea a puñetazos. Es triste decirlo, pero el único árbitro que tenemos no parece objetar las peleas, solo las regula («por cada diez metros que corras, puedes realizar un disparo»). Lo que debería hacer es separarlos y penalizar a los corredores, incluso por tratar de enredarse en una pelea.

### **«¿No dejarán todos de programar si no hay un incentivo económico?»**

De hecho, mucha gente programará sin absolutamente ningún incentivo económico. La programación ejerce una atracción irresistible en algunas personas, generalmente en quienes son los mejores en ese ámbito. No hay escasez de músicos profesionales que sigan en lo suyo aunque no tengan esperanzas de ganarse la vida de esa forma.

En realidad esta pregunta, aunque se formula muchas veces, no es adecuada para la situación. El pago a los programadores no va a desaparecer, solo se va a reducir. La pregunta correcta es: ¿Alguien programará si se

reduce el incentivo económico? Mi experiencia muestra que sí lo harán.

Por más de diez años, muchos de los mejores programadores del mundo trabajaron en el Laboratorio de Inteligencia Artificial por mucho menos dinero de lo que podrían haber obtenido en otro sitio. Tenían muchos tipos de recompensas que no eran económicas: fama y aprecio, por ejemplo. Y la creatividad también es divertida, es una recompensa en sí misma.

Luego la mayoría se fue cuando se les ofreció la oportunidad de hacer ese mismo trabajo interesante por mucho dinero.

Lo que muestran los hechos es que la gente programa por razones distintas a la de la riqueza; pero si se les da la oportunidad de ganar también mucho dinero, eso los llenará de expectativas y lo van a exigir. Las organizaciones que pagan poco no podrán competir con las que pagan mucho, pero no tendría que irles tan mal si las que pagan mucho fueran prohibidas.

**«Necesitamos a los programadores desesperadamente. Si ellos nos pidieran que dejemos de ayudar a nuestro prójimo, tendríamos que obedecer».**

Uno nunca está tan desesperado como para tener que obedecer este tipo de exigencia. Recuerde: millones para nuestra defensa, ¡pero ni un centavo para tributos! <sup>[2]</sup>.

**«Los programadores necesitan tener alguna forma de ganarse la vida».**

A corto plazo, esto es verdad. Sin embargo, hay bastantes maneras de que los programadores puedan ganarse la vida sin vender el derecho a usar un programa. Esta manera es frecuente ahora porque es la que les da a los programadores y hombres de negocios más dinero, no porque sea la única forma de ganarse la vida. Es fácil encontrar otras formas, si quieres encontrarlas. He aquí unos cuantos ejemplos:

Un fabricante que introduce una nueva computadora pagará por adecuar los sistemas operativos al nuevo hardware.

La venta de enseñanza, los servicios de asistencia y mantenimiento también pueden dar trabajo a programadores.

La gente con ideas nuevas podría distribuir programas como *freeware* (9), pidiendo donaciones a los usuarios satisfechos, o vendiendo servicios de asistencia. Yo he conocido a personas que ya trabajan así y con mucho éxito.

Los usuarios con que tengan las mismas necesidades pueden formar un grupo de usuarios y pagar sumas de dinero. Un grupo contratará a empresas de programación para escribir programas que a los miembros del grupo les gustaría utilizar.

Todo tipo de desarrollo puede ser financiado con un impuesto al software:

Supongamos que todos los que compran una computadora tengan que pagar un tanto por ciento de su precio como impuesto de software. El Gobierno entrega este dinero a una agencia como la la Fundación Nacional de las Ciencias (NSF) para que lo emplee en el desarrollo de software.

Pero si el comprador de la computadora hace por sí mismo un donativo para el desarrollo de software puede verse exento de este impuesto. Puede donar al proyecto de su elección —a menudo, elegido porque espera utilizar los resultados tan pronto como se haya completado. Puede tomar crédito por cada cantidad de donación hasta la totalidad del impuesto que tenía que pagar.

La tasa total de impuesto podría decidirse mediante el voto de los contribuyentes, sopesada de acuerdo con la cantidad sobre la que se aplicará el impuesto.

Las consecuencias:

- La comunidad usuaria de computadoras apoya el desarrollo de software.
- Esta comunidad decide qué nivel de apoyo es necesario.
- Los usuarios a quienes les importa a qué proyectos se destine su parte pueden escogerlos por sí mismos.

A largo plazo, hacer programas libres es un paso hacia el mundo post-escasez, donde nadie tendrá que trabajar duro para ganarse la vida. La gente será libre para dedicarse a actividades entretenidas, como la programación, después de haber dedicado diez horas obligatorias a la semana a las tareas requeridas, como legislar, el asesoramiento familiar, la reparación de robots y la exploración de asteroides. No habrá necesidad de ganarse la vida mediante la programación.

Hemos alcanzado ya una gran reducción de la cantidad de trabajo que la sociedad en su conjunto debe realizar para mantener su productividad actual, pero solo un poco de esta reducción se ha traducido en descanso para los trabajadores, dado que hay mucha actividad no productiva que se requiere para acompañar a la actividad productiva. Las causas principales de esto son la burocracia y las luchas isométricas contra la competencia. El software libre reducirá en gran medida estos drenajes en el campo de producción de software. Debemos hacerlo, para que los avances técnicos en la productividad se traduzcan en menos trabajo para nosotros.

## Notas

1. Esta expresión resultó poco precisa. La intención era decir que nadie tendría que pagar por el **permiso** para usar el sistema GNU. Pero la expresión no es del todo clara, y a menudo se interpreta que las copias de GNU deberían distribuirse siempre a un costo bajo o sin costo. Esta nunca fue la intención. Más adelante, el manifiesto menciona la posibilidad de que las empresas provean servicios de distribución para obtener ganancias. A partir de entonces, aprendí a distinguir cuidadosamente entre «*free*» (libre) en el sentido de libertad y «*free*» (gratis) referido al precio <sup>[3]</sup>. El software libre es aquel que ofrece a los usuarios la libertad de distribuirlo y modificarlo. Algunos pueden obtener copias sin pagar, mientras que otros pagan para obtenerlas, y si los fondos ayudan a apoyar la mejora del software, tanto mejor. Lo importante es que todos los que posean una copia tengan la libertad de colaborar con los demás al usar el programa.
2. La expresión «regalar» es otro indicio de que yo todavía no había separado claramente la cuestión del precio de la cuestión de la libertad. Ahora recomendamos no usar esta expresión al hablar acerca del software libre. Para una explicación más detallada, consulte el artículo «Palabras y frases confusas que vale la pena evitar».
3. Aquí también omití distinguir cuidadosamente entre los dos diferentes significados de «*free*» (que en inglés puede significar «gratis» o «libre», N. d. T.). La afirmación tal como está escrita no es falsa, se pueden obtener copias gratuitas de software de GNU —de los amigos o a través de Internet—, pero sugiere una idea errónea.
4. Ya existen varias compañías de este tipo.
5. Aunque es una organización sin ánimo de lucro más que una empresa, la *Free Software Foundation* durante diez años ha obtenido la mayoría de los fondos a partir de su servicio de distribución. Puede comprar artículos de la FSF para apoyar su actividad.
6. Un grupo de empresas de informática alrededor de 1991 reunió fondos para apoyar el mantenimiento del compilador C de GNU.
7. Creo que me equivoqué al decir que el software privativo era la base más común para ganar dinero en el campo del software. Parece ser que en realidad el modelo de negocio más común era y es el desarrollo de software a medida, que no ofrece la posibilidad de percibir una renta, por lo que la empresa tiene que seguir haciendo el trabajo real para seguir recibiendo ingresos. El negocio del software a medida podrá seguir existiendo, más o menos igual, en un mundo de software libre. Por lo tanto, ya no supongo que los programadores ganarían menos en un mundo de software libre.
8. En la década de 1980 todavía no me había dado cuenta de lo confuso que era hablar de «la cuestión» de la «propiedad intelectual». Esa expresión es obviamente prejuiciosa, más sutil es el hecho de que agrupa leyes dispares que plantean cuestiones muy diferentes. Hoy en día insto a la gente a rechazar completamente el término «propiedad intelectual», para no inducir a otros a pensar que esas leyes forman un tema coherente. Para hablar con claridad, hay que referirse a las patentes, el copyright y las marcas

registradas por separado. Véase una explicación más amplia de cómo esta expresión genera confusión y prejuicios.

9. Posteriormente aprendimos a distinguir entre «software libre» y «*freeware*». El término «*freeware*» significa que el software se puede redistribuir libremente, pero por lo general no ofrece la libertad para estudiar y modificar el código fuente, así que la mayoría de esos programas no son software libre. Véase “palabras y frases confusas que vale la pena evitar” para más detalles.

### Notas de traducción

[1]: GNU se pronuncia en inglés de forma muy similar a «*new*», que significa «nuevo».

[2]: Véase [http://es.wikipedia.org/wiki/Caso\\_XYZ](http://es.wikipedia.org/wiki/Caso_XYZ) para más información sobre el contexto de esta sentencia.

[3]: En inglés, el término «*free*» puede referirse tanto a la libertad como al precio.

---

Copyright © 1985, 1993, 2003, 2005, 2007, 2008, 2009, 2010, 2014, 2015 Free Software Foundation, Inc.

Se autoriza la copia literal o la distribución de este documento en su totalidad, por cualquier medio, siempre y cuando se conserven las notas del copyright y de la autorización, y siempre y cuando el distribuidor otorgue a los destinatarios la autorización para la ulterior redistribución según los términos de esta nota.

No se permite la realización de copias modificadas.