

EN EXCLUSIVO PARA HACKERS &amp; DEVELOPERS

# Despidiendo el año junto a Richard Stallman

por Eugenia Bahit

Inteligente, comprometido con sus ideales; divertido y amable, con la humildad que solo las personas de una grandeza espiritual distinguida pueden tener. Luchador, perseverante; honesto, directo y sin tapujos a la hora de expresarse y dar sus opiniones. Fiel amante de la libertad, con una distinguida trayectoria y una actitud coherente entre “el dicho y el hecho”, de la cuál pocos pueden darse el lujo. Richard Matthew Stallman: un hombre que inspira.

## Preparativos...

Corría el 4 de diciembre. Se acercaba el momento de hacer planes para el especial de fin de año. Recién terminábamos de publicar la segunda edición del Magazine y solo sabíamos que teníamos que hacer algo “verdaderamente especial”. Algo que como equipo, nos ayudara a marcar “un antes y un después”. Algo que nos fortaleciera como grupo y que nos hiciera sentir orgullosas de trabajar las unas con las otras.

Fuimos arrojando ideas y así surgieron los “tips” (de finales de artículos), los cupones de descuento y algo **tan ambicioso como utópico que jamás creímos podía convertirse en realidad**: conseguir una entrevista con Richard Stallman.

Utópica y ambiciosa (o no), semanas después, me animé y le escribí un e-mail en un inglés que seguramente daba pena. Largo. Larguísimo, como si no hubiese sido consciente del tiempo que le llevaría leerlo. Como si ni siquiera hubiera tenido consciencia, del poco conocimiento que tengo de su idioma natal.

Le conté sobre el Magazine; le hablé de lo que hacíamos y le ofrecí leer la revista. Para mi sorpresa, la respuesta estaba en mi bandeja de entradas, en menos de 24 horas: **“It sounds like a good goal (...)”**, fue la primera línea de su respuesta.



Foto: Martin Kozák (bajo dominio público, via Wikimedia Commons)

Intercambiamos algunos correos más, hasta que en un perfecto español, me dijo: **“¿Quieres llamarme el sábado?”** y me adjuntaba su número de teléfono.

Decir emoción, me suena a “poco” para describir el cúmulo de sensaciones que se han cruzado en mí, desde ese día, hasta hoy, sábado 29 de diciembre a las 17:30 HS de Argentina, momento en el cual, tuvimos la siguiente charla:

**Richard, ¿cómo es un día tuyo en la fundación dentro de la Free Software Foundation?** Mi trabajo es mayormente leer y contestar mensajes y son de muchos tipos. Por ejemplo hay noticias, me piden decidir algo, hay mensajes desde los usuarios... entonces son muchos tipos de trabajo pero casi todos llegan por correo y envío mi decisión o respuesta por correo.

*Por principios, en la FSF decidí no aceptar dinero*

**Y tu trabajo en la Free Software Foundation ¿es 100% voluntario? ¿No cobras un sueldo?** Si, no recibo salario y la FSF no paga mis viajes. Es por principios. Decidí no aceptar dinero de la FSF porque tenía que pedir a otros que trabajen como voluntarios (...).

**Y más allá de la Free Software Foundation ¿qué otro trabajo tienes o qué otro trabajo “remunerado”?** Me pagan por quizás la mitad de mis conferencias.

**¿Te gusta realmente viajar o te estresa?** Me gusta. No siempre, pero mayormente me gusta.

**¿Cómo surgen las campañas dentro de la FSF?** Alguien tiene la idea, a veces yo, a veces alguien del equipo. Porque tenemos a dos empleados para las campañas (...) **y si por ejemplo algún usuario envía un e-mail a la FSF proponiendo una campaña...** si nos gusta, lo haremos. **¡Es muy bueno saberlo!** Siempre valoremos las buenas sugerencias.

*Debian en sus foros, aconseja solucionar problemas instalando Software privativo*

**Con respecto a la FSF, el Software Libre y las distribuciones GNU/Linux aceptadas ¿en qué punto se encuentran actualmente con Debian?** Debian tiene problemas. Un problema es que distribuye y recomienda paquetes no libres. Pero también en sus foros hay preguntas “¿Cómo puedo hacer esto?” y la respuesta es instalar ese programa privativo. Por eso, no endosamos Debian. Saca algunos programas privativos como soluciones y tratarlos como soluciones es negar... es rehusar a considerarlos como problemas. Entonces esa contradicción

entre su práctica y nuestra filosofía es nuestro motivo de no avalar. **Leí hace poco en un foro, que Debian había presentado otra versión a la Free Software Foundation para ver si era aceptada ¿es cierto?** No recuerdo (...) me sorprendería. No comprendo como una nueva versión de algo podría resolver este problema. Porque otro criterio para aprobar una distribución GNU con Linux es que no tenga relación visible con ningún programa privativo. Es decir que si el mismo proyecto publica dos versiones y una contiene software privativo, no podemos aprobar ninguna. Tiene demasiada relación con software privativo. Si Debian echa los paquetes privativos; si corta su relación con esos paquetes, podríamos aprobarlo.

*Con Linus Torvalds, no somos amigos*

**Con Linus Torvalds ¿tienes alguna relación?** De disputa. No somos amigos. **No son amigos... ¿puedo preguntar por qué?** Nunca estuvo de acuerdo con la filosofía del Software Libre. Políticamente nunca estuvo de acuerdo con nosotros. Pero también tiene la tendencia de enojarse y decir cosas muy duras a quien quiera. Incluso a nosotros -que no es amigable-. Pero también nos critica por nuestro intento de proteger a los usuarios de la práctica de “tivoización” (NdR: se refiere a la castellanización del término inglés “Tivoization<sup>30</sup>”), que significa fabricar computadoras con Software bajo la GPL versión 2 (usualmente), de manera que el código fuente es libre pero que el ejecutable en la computadora es privativo. Y hoy en día muchos dispositivos Android lo hacen con Linux, porque Linux se publica bajo la versión 2 de la GPL que escribí en el año '91. En aquel año no existía el problema de la tivoización y no anticipaba este problema entonces no hice nada para evitar este problema. Y Torvalds quiere mantener Linux bajo la versión 2 de la GPL para que permita la tivoización. Es lo que dice. El está en favor de permitir que los usuarios pierdan su libertad, así. Decidió quedarse con la versión 2

30 <http://es.wikipedia.org/wiki/Tivoizaci%C3%B3n>

que es un problema. Pero el aspecto central de este problema es la tivoización que quita la libertad a muchos usuarios. Pero, otra cosa es, que escribió un programa importante. El núcleo que se usa con el sistema operativo GNU en la combinación GNU con Linux.

*Muchas personas llaman Linux al sistema entero, dándole el crédito de nuestro trabajo a otros*

Pero erróneamente muchos se acostumbraban a llamar el sistema entero como Linux, dándole todo el crédito para nuestro trabajo anterior. Entonces, comencé a pedir siempre que la gente reconozca nuestro trabajo. Pero esto no es una crítica de Torvalds. Durante unos años no se oponía a esta campaña nuestra pero más tarde, desde hace unos años, intenta argumentar que es incorrecto llamar al sistema como GNU con Linux. Intenta persuadir a la gente que no reconozca nuestro trabajo. **Pero ¿qué fue lo que sucedió con Linus Torvalds para que de repente, primero estaba a favor y no criticaba y ahora, hace unos años empezó?** No se. No se porqué pensaba “así” y luego “pensaba así”. Uno tendría que conocerlo mucho más que yo (...). **Y ¿alguna vez le preguntaste?** No. Porque cuando alguien llega a ese punto, no tengo porque hablar con él. Cuando alguien intenta convencer a la gente de que no reconozca nuestro trabajo, es tratarnos muy mal.

*No admiro a Dennis Ritchie*

**Y hablando de Linus Torvalds y “famosos” en el ámbito de la informática, alguna vez ¿tuviste oportunidad de encontrarte con Dennis Ritchie?** No. **¿nunca tuviste oportunidad, nunca la buscaste...?** No, ni ganas. No me interesó. **No te interesó... ¿puedo preguntar por qué?** ¿Qué hizo que me interesaría? **A ver...** solo hizo trabajos técnicos.

No lo admiro por eso. Desarrolló un sistema operativo no ético que tuve que reemplazar. Eso no es lo que admiro.

**Hablando de privativo... Python actualmente, tiene una licencia compatible con la GPL. Pero tuvo una especie de ida y vuelta con si era compatible o no era compatible, en versiones creo que anterior a la 2. Fue hace mucho tiempo. No recuerdo los detalles. Hablamos con el desarrollador y por fin... ¿Con Guido Van Rossum?** Si. Pero no fue solo él, porque fue su empleador el que hizo el problema y no él. **Ah, mira que interesante...** trabajaba, quizás para una Universidad, no lo recuerdo, pero su empleador deseaba cambiar la licencia y creó un problema, que por fin, resolvimos.



Foto: Lionel Allorge - CC-BY-SA-3.0

**Yendo a algo más personal... ¿eres de esas personas que se aburren con facilidad...?** Si. **¿Si? Y ¿eres de los que necesita estar siempre haciendo algo productivo?** No necesariamente productivo. A veces hago algo por placer y también a veces necesito divertirme. **Y ¿Qué cosas haces para divertirte?** A veces cuando tengo quizás un poco de sueño y no sostengo concentrar más en algo difícil, me divierto con algún videojuego por algún tiempo. **¿Te gusta jugar videojuegos! ¿Qué videojuego juegas?** Prefiero no decirlo, pero es Software Libre.

**Y con respecto a la programación, ¿sigues programando a menudo?** No. **¿estás más avocado a lo que es las charlas y demás...?** Si. Mi trabajo es el avance del movimiento. **Y ¿no extrañas sentarte a programar durante horas?** No. **¿No? No. Bueno, bien. Y hablando de tus proyectos, GNUPedia ¿en que quedó?** Se fusionó con Nupedia para lanzar Wikipedia. **Ah, no sabía.** Así nació Wikipedia, con la fusión de dos proyectos: Nupedia y GNUPedia...

Y lo divertido es, que hablaba con ellos durante más o menos un año, pero solo raras veces y entonces, no sabía que había dos. Y por fin llegaron dos mensajes dentro de bastante poco tiempo para que supiera que eran dos y luego los puse en contacto entre ellos.

*Ser un buen programador es hacer programas útiles que funcionen bien*

**¿Qué significa para vos ser un buen programador?** Hacer programas útiles, que funcionen bien y en hacerlo rápidamente. Para ser programador ético, hace falta distribuirlos respetando la libertad de sus usuarios. **Y a los programadores que hasta hoy en día no se animaron a liberar sus códigos y a desarrollar Software Libre ¿qué les dirías para que se animen?** Ah, no se. Porque depende de cuáles son sus valores. Pero lo que digo generalmente, es que un programa ofrecido sin libertad es una trampa y aceptarlo es ser tonto. Y crear trampas para meter a la gente en la trampa, es un abuso. No se debe.

**Y hablando de trampas y demás: Open Source es una contrapartida a lo que es el movimiento del Software Libre. No tienen como principio la libertad sino que se refieren más al código fuente pero no hacen hincapié en lo que socialmente significa el Software Libre...** ese término fue inventado como una manera de hablar del Software Libre, de los mismos programas, pero sin plantearlo como un asunto ético (...) si sabiendo esto, miras lo q' dicen, verás que siempre evitan decir "El Software no libre es injusto". Este es el punto, que no quieren decirlo y nunca lo dicen. **Y también sucede que muchas compañías desarrolladoras de Software, utilizan el término Open Source para referirse a que simplemente colocan el código fuente de un programa, a disposición de cualquier usuario, pero sin embargo, no respetan las cuatro libertades que sí propone el Software Libre.**

*La definición de Open Source es más o menos equivalente a la de Software Libre, pero el término genera problemas*

Eso es un abuso del término Open Source. Porque no siguen la definición que es más o menos equivalente a la de Software Libre. Pero este mal entendido es muy común y empeora el problema que el término causa. **Y con la gente de la Iniciativa Open Source, ¿tienes algún tipo de trato?** Muy poco, muy poco.



Foto: Victor Hermida Prada – CC-BY-2.0

*A los programadores que quieren contribuir con el movimiento,*

*les sugiero  
participar en la  
Ingeniería Inversa*

**Richard ¿algo que nos quieras decir a todos los programadores, seamos programadores de Software Libre o los que están en la duda?**

Es muy amplio. Pero a los que quieren contribuir éticamente al movimiento, lo que sugiero es participar en la ingeniería inversa. Es un campo absolutamente importante para nuestro progreso, porque nos encontramos bloqueados por los periféricos cuyo modo de empleo es secreto. Sobre todo, por ejemplo, los aceleradores de vídeo. Hace falta descubrir su modo de empleo para poder escribir Software Libre y entonces, hace falta la ingeniería inversa.

**En la Free Software Foundation, uno ¿de qué forma puede aportar para colaborar con ustedes?** Hay varios grupos de trabajo. Por ejemplo hay un grupo que contesta preguntas sobre las licencias. Y hay otro grupo que sugiere proyectos de programación a los voluntarios. Hay otros trabajos que también hacemos en los cuales los voluntarios pueden participar. Dice en fsf.org como participar. Otras maneras de contribuir al movimiento es organizar grupos activistas locales y en Argentina hay que luchar contra la distribución de Windows (*NdR: en las Netbooks que otorga el Estado Nacional*) (...)

*Cuando hay errores  
reproducibles en el  
código, lanzar el  
depurador es la  
manera más fiable  
de detectarlos*

¿Quieres un poco de consejo para los programadores? ¡Me encantaría! Cuando hay un error lanza inmediatamente el depurador, para saber que pasa dentro del programa. No intentes adivinar. Si no es obvio no puedes adivinar. Pero puedes detectar el problema con el depurador. Y la otra cosa que no hay que hacer, es intentar varias maneras de invocar el

programa para buscar un patrón de los casos del error, porque no puedes deducir donde está el error, así. La manera fiable de detectar el error, es con el depurador, porque así puedes hacer que el programa pare en algunos puntos para mirar los datos intermediarios y así puedes localizar en el programa y en la ejecución de este caso, dónde sucede un valor erróneo (...) suponiendo que es un error reproducible. Para un error infrecuente, irreproducible, es más difícil siempre, pero cuando es reproducible, siempre es posible de manera, conceptualmente sencilla, localizar el error en una línea de código.

*Lisp, es el lenguaje  
más elegante y  
poderoso; tiene  
capacidades  
ausentes en los  
otros lenguajes*

**Muy buen consejo Richard y, aprovecho y te hago una pregunta más: ¿cuál es tu lenguaje de programación, el que más te apasione programar?** Lisp. Lisp es el lenguaje más elegante y poderoso. Porque Lisp tiene capacidades que faltan en todos los otros lenguajes. **Por ejemplo ¿cuáles?** Cuando lanzas el sistema de Lisp, hace un bucle de read-eval-print. Read, significa leer una expresión y convertirla en datos. Porque en Lisp, cualquier expresión tiene una representación natural y sencilla como datos. Luego hace eval, es la evaluación de una expresión como datos, para poder producir un resultado que también es datos. Luego print que convierte los datos en representación textual. En otros sistemas, no hay read, no hay eval y no hay print. Porque no hay conversiones generales entre texto y datos y porque un programa no tiene representación natural como datos. Entonces read no tiene sentido y tampoco eval tiene sentido. Entonces, casi todo en Lisp es ausente en otro lenguaje.

*“C” no tiene el  
poder de Lisp pero  
me gusta bastante*

Hay una variante de Lisp que se llama Scheme, puedes aprender uno u otro. Porque tienen en común a la mayoría de las cosas interesantes (...) pero en cuanto a los lenguajes algebraicos, me gusta C... **Y sí, C es como el lenguaje por excelencia, ¿no?** Si, un amigo tenía una chapa que decía "El lenguaje C combina el poder del ensamblador con la comodidad del ensamblador" (NdR: Richard Stallman me tuvo que explicar el chiste, porque para variar, mi espíritu Sheldon Cooper, me impidió entender el chiste por mis propios medios). Pero de todos modos (...) me gusta bastante C. No tiene el poder de Lisp: un programa no tiene forma como datos, en C, pero para compilar algo, si aceptas que no vas a poder cambiar el programa durante su ejecución, entonces me gusta.

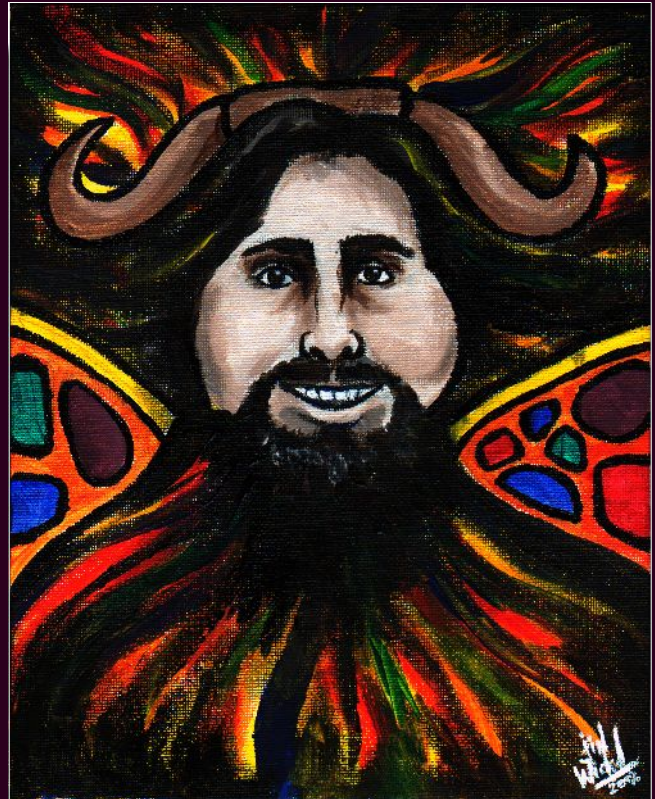
**Bueno Richard, nuevamente, un millón de gracias. Ha sido un enorme placer hablar contigo. Happy Hacking! Happy Hacking, Richard! Bye! Bye Bye!**

## Información adicional

Richard Stallman, nació en Estados Unidos en el año 1953. Preside la fundación sin ánimo de lucro, **Free Software Foundation**. En el sitio Web de la fundación, [www.fsf.org](http://www.fsf.org) puedes encontrar suficiente información para participar activamente en proyectos, grupos de voluntarios o haciendo un donativo.

GNU, es el sistema operativo que utiliza el núcleo creado por Linus Torvalds. La mayor parte de las distribuciones que utilizan este núcleo, están basadas en el sistema operativo GNU en combinación con el núcleo Linux. De allí, que no es correcto decir "distribución Linux", sino "**distribución GNU/Linux**", puesto que **muchísimos otros sistemas operativos utilizan el mismo kernel (núcleo Linux) y sin embargo, no son libres**. Información completa sobre esto, puedes encontrar en el sitio Web del proyecto GNU, ingresando en [www.gnu.org](http://www.gnu.org).

**Richard Stallman**, dedica parte de su tiempo, a mantener su **sitio Web personal** [www.stallman.org](http://www.stallman.org) actualizado con noticias, información personal y artículos de opinión propios, no solo referidos al Software Libre, sino también, a temas como el arte y la política.



*Ilustración creada por Jin Wicked  
con permiso para distribución exacta  
(sin modificaciones).*

<http://www.jinwicked.com/>