

Cómo convertirse en hacker

Eric Steven Raymond

[Thyrsus Enterprises](#)

<esr@thyrsus.com>

Copyright © 2001 Eric S. Raymond

Traducción: Miquel Vidal <miquel@sindominio.net>, a partir de una versión previa realizada por [C. Ballardi](#).

Tabla de Contenidos

[¿Por qué existe este documento?](#)

[¿Qué es un hacker?](#)

[La actitud del hacker](#)

- [1. El mundo está lleno de problemas fascinantes que esperan ser resueltos](#)
- [2. Ningún problema tendría que resolverse dos veces](#)
- [3. El aburrimiento y el trabajo rutinario son perniciosos](#)
- [4. La libertad es buena](#)
- [5. La actitud no es sustituto para la competencia](#)

[Habilidades básicas para el hacking](#)

- [1. Aprende a programar](#)
- [2. Consigue uno de los Unices libres; aprende a usarlo y a ponerlo en funcionamiento](#)
- [3. Aprende a usar la Web y a programar en HTML](#)
- [4. Si no tienes un inglés funcional, apréndelo](#)

[Estatus en la cultura hacker](#)

- [1. Programar software libre](#)
- [2. Ayudar a probar y depurar software libre](#)
- [3. Publicar información útil](#)
- [4. Ayudar a mantener en funcionamiento la infraestructura](#)
- [5. Hacer algo por la cultura hacker en sí misma](#)

[La conexión hacker/nerd](#)

[Cuestiones de estilo](#)

[Otras fuentes de información](#)

[Preguntas Frecuentes](#)

¿Por qué existe este documento?

Como editor del [Jargon File](#) (el "Archivo de la jerga hacker") y autor de algunos documentos bien conocidos del mismo tipo, a menudo recibo emails de entusiastas novatos de la Red preguntando: "¿Cómo puedo aprender a ser un hacker mágico?" Triste es decir que no parece haber ninguna FAQ ni documento web que aborde este tema vital, así que aquí va el mío.

Si estás leyendo este documento sin conexión, la versión original en línea la puedes encontrar en <http://www.tuxedo.org/~esr/faqs/hacker-howto.html>.

Nota: hay una lista de [Preguntas Frecuentes](#) al final de este documento. Por favor léelo —2 veces— antes de mandarme un email con preguntas acerca de este documento.

Hay disponibles numerosas traducciones de este documento: [búlgaro](#), [catalán](#), [chino \(simplificado\)](#), [chino \(tradicional\)](#), [danés](#), [neerlandés](#), [francés](#), [alemán](#), [hebreo](#), [húngaro](#), [indonesio](#), [italiano japonés](#), [coreano](#), [portugués \(brasileño\)](#), [portugués \(europeo\)](#), [ruso](#), y [sueco](#). Ten en cuenta que, como el contenido de este documento cambia ocasionalmente, dichas traducciones pueden estar desactualizadas en grado diverso.

¿Qué es un hacker?

El [Jargon File](#) contiene un montón de definiciones del término "hacker", la mayoría basadas en la afición a lo técnico y en el placer de resolver problemas sobrepasando los límites. Si deseas saber cómo *convertirte en un hacker*, bien, solo 2 puntos son realmente relevantes.

Existe una comunidad, una cultura compartida, de programadores expertos y magos de las redes, cuya historia se remonta décadas atrás a los tiempos de los primeros miniordenadores de tiempo compartido y los tempranos experimentos con ARPAnet. Los miembros de esta cultura crearon el término "hacker". Los hackers construyeron Internet. Los hackers hicieron de Unix el sistema operativo que es hoy día. Los hackers hacen andar Usenet. Los hackers hacen funcionar la WWW. Si eres parte de esta cultura, si has contribuido a ella y otras personas saben quién eres y te llaman hacker, entonces eres un hacker.

Los mentalidad hacker no está confinada a esta cultura del software. Hay gente que aplica la actitud de hacker a otras cosas, como la electrónica o la música —de hecho, puedes encontrarla en los más altos niveles de cualquier ciencia o arte. Los hackers de software reconocen estos espíritus emparentados en otras partes y pueden llamarlos "hackers" también— y algunos sostienen que la naturaleza hacker es en realidad independiente del medio particular en el cual el hacker trabaja. Sin embargo, en el resto de este documento nos centraremos en las habilidades y actitudes de los hackers de software, y en las tradiciones de la cultura compartida que originó el término "hacker".

Existe otro grupo de personas que se llaman a sí mismos hackers, pero que no lo son. Son personas (generalmente varones adolescentes) que se divierten irrumpiendo ilegalmente en ordenadores y haciendo "phreaking" en el sistema telefónico. Los auténticos hackers tienen un nombre para esas personas: "crackers", y no quieren saber nada de ellos. Los auténticos hackers opinan que la mayoría de los crackers son perezosos, irresponsables y no muy brillantes, y fundamentan su crítica en que ser capaz de romper la seguridad no le hace a uno un hacker, de la misma manera que ser capaz de arrancar un coche con un puente en la llave no le convierte en ingeniero de automotores. Desafortunadamente, muchos periodistas y escritores utilizan erróneamente la palabra "hacker" para describir a los crackers; esto causa enorme irritación a los auténticos hackers.

La diferencia básica es esta: los hackers construyen cosas; los crackers las destruyen.

Si quieres ser un hacker, continúa leyendo. Si lo que quieres es ser un cracker, ve directo a leer el [alt.2600](#) y prepárate para soportar la dura realidad cuando descubras que no eres tan listo como crees. Y eso es todo lo que voy a decir de los crackers.

La actitud del hacker

Los hackers resuelven problemas y construyen cosas, y creen en la libertad y la ayuda voluntaria mutua. Para ser aceptado como hacker, deberás comportarte como si tuvieras esta actitud en tu interior. Y para comportarte como si tuvieras esta actitud, deberás creerte de verdad dicha actitud.

Pero si piensas en cultivar las actitudes de hacker solo como una forma de ganar aceptación en esta cultura, te estás equivocando. Transformarse en la clase de persona que cree estas cosas es importante para *ti* —para ayudarte a aprender y mantenerte motivado. Como en todas las artes creativas, el modo más efectivo de transformarse en un maestro es imitar la mentalidad de los maestros —no sólo intelectualmente, sino también emocionalmente.

O como dice el siguiente poema zen moderno:

Para seguir la trayectoria:
mira al maestro,
sigue al maestro,
camina junto con el maestro,
mira a través del maestro,
conviértete en el maestro.

Así que, si quieres ser un hacker, repite lo que sigue hasta que te creas lo que estás diciendo:

1. El mundo está lleno de problemas fascinantes que esperan ser resueltos

Es muy divertido ser un hacker, pero es la clase de diversión que requiere mucho esfuerzo. El esfuerzo requiere motivación. Los atletas triunfadores obtienen su motivación a partir de un tipo de placer físico que surge de trabajar su cuerpo, al forzarse a sí mismos más allá de sus propios límites físicos. De manera similar, para ser un hacker deberás sentir un estremecimiento de tipo primitivo cuando resuelves problemas, afinas tus habilidades y ejercitas tu inteligencia.

Si no eres la clase de persona que te sientes inclinado hacia estas cosas de forma natural, necesitarás poder experimentarlas para llegar a ser un hacker. De no ser así, encontrarás que tu energía para "hackear" se verá agotada por otras distracciones como el sexo, el dinero o la aprobación social.

(Además deberás desarrollar cierta clase de fe en tu propia capacidad de aprendizaje —la creencia de que, aun cuando puedes no saber todo lo que necesitas para resolver un problema, si tomas una parte de él y aprendes desde allí, aprenderás lo suficiente para resolver la siguiente parte, y así, hasta que lo tengas resuelto por completo.)

2. Ningún problema tendría que resolverse dos veces

Los cerebros creativos son un recurso valioso y limitado. No deben desperdiciarse reinventando la rueda cuando hay tantos y tan fascinantes problemas nuevos esperando por allí.

Para comportarte como un hacker, debes creer que el tiempo para pensar que emplean otros hackers es precioso —tanto, que es casi una obligación moral para ti el compartir la información, resolver los problemas y luego exponer la solución de manera que los otros hackers puedan resolver *nuevos* problemas, en lugar de enfrentarse perpetuamente con los viejos.

(No tienes que pensar que estás obligado a regalar *todo* tu producto creativo, aunque aquellos hackers que lo hacen son los que obtienen más respeto por parte de los demás hackers. Es consistente con los valores de un hacker venderse lo suficiente para pagar la comida, el alquiler y los ordenadores. También está bien la utilización de estas habilidades de hacker para sostener a la familia, o incluso hacerse rico, siempre que no olvides la lealtad a tu arte y a tus colegas hackers mientras lo haces.)

3. El aburrimiento y el trabajo rutinario son perniciosos

Los hackers (y las personas creativas en general) nunca debieran ser sometidas a trabajos estupidamente repetitivos, porque cuando esto sucede significa que no están haciendo lo único que son capaces de hacer: resolver nuevos problemas. Este desperdicio de talento daña a todo el mundo. Por ello, las tareas rutinarias, repetitivas y aburridas no sólo son desagradables, sino intrínsecamente perversas.

Para comportarte como un hacker, debes creer en esto lo suficiente como para automatizar las tareas rutinarias todo lo que se pueda, no solamente por ti mismo, sino para beneficio de todos los demás (especialmente otros hackers).

(Existe una aparente excepción a esta regla. Los hackers hacen cosas a veces que pueden parecer repetitivas o aburridas a un observador, pero son un ejercicio para lograr claridad mental o para adquirir cierta habilidad u obtener cierta clase de experiencia que no podría conseguirse de otro modo. Pero esto es una elección —ningún ser pensante debiera nunca ser forzado a una situación que les aburra.)

4. La libertad es buena

Los hackers son antiautoritarios por naturaleza. Cualquiera que pueda darte órdenes, puede obligarte a dejar de resolver ese problema que te está fascinando —y, dada la manera como trabajan las mentes autoritarias, encontrarán alguna razón espantosamente estúpida para hacerlo. Por eso, la actitud autoritaria debe ser combatida donde sea que se la encuentre, pues si se la deja te asfixiará, tanto a ti como a otros hackers.

(Esto no es lo mismo que combatir toda autoridad. Los niños necesitan guía, y los criminales, restricciones. Un hacker puede estar de acuerdo en aceptar algún tipo de autoridad para poder obtener algo que desea más que el tiempo que gasta en seguir órdenes. Pero este es un pacto limitado, consciente; la clase de sumisión que desean los autoritarios no está en oferta.)

Los autoritarios prosperan en la censura y el secreto. Y desconfían de la cooperación voluntaria y del intercambio de información —sólo les agrada la cooperación que tienen bajo su control. Así que, para comportarte como un hacker, deberás desarrollar una hostilidad instintiva hacia la censura, el secreto y la utilización de la fuerza o el fraude para someter a adultos responsables. Y deberás estar preparado para actuar en consecuencia.

5. La actitud no es sustituto para la competencia

Para ser un hacker, deberás desarrollar algunas de esas actitudes. Pero tener solo la actitud no te transformará en hacker, como tampoco te puede transformar en un atleta campeón o en estrella del rock. Para transformarte en hacker necesitarás inteligencia, práctica, dedicación y trabajo duro.

Por lo tanto, debes aprender a desconfiar de la actitud y respetar la competencia en todas sus formas. A ningún hacker le gusta perder el tiempo con quienes adoptan la pose de hacker, sino que veneran la competencia —especialmente la competencia al hackear, pero la competencia en cualquier ámbito está bien. Especialmente buena es la competencia en habilidades exigentes que poca gente domina, y la mejor es la competencia en habilidades exigentes que requieren agudeza mental, destreza y concentración.

Si respetas la competencia, disfrutarás desarrollándola en ti mismo —el trabajo duro y la dedicación se transformarán en una especie de juego intenso, y no una rutina. Esa actitud es vital para transformarse en hacker.

Habilidades básicas para el hacking

La actitud hacker es vital, pero más aún lo son las habilidades. La actitud no es sustituto para la competencia, y existe un cierto conjunto de herramientas básicas que debes dominar antes que cualquier hacker sueñe con llamarse así.

Este conjunto de herramientas cambia lentamente a lo largo del tiempo a medida que la tecnología crea nuevas habilidades y descarta otras por obsoletas. Por ejemplo, se incluía la programación en lenguaje máquina, y no se hablaba de HTML hasta hace poco. Pero hacia finales de 1996, está claro que se debe incluir lo siguiente:

1. Aprende a programar

Esta es, por supuesto, la habilidad fundamental del hacker. Si no conoces ningún lenguaje de programación, recomiendo empezar con Python. Es limpio, bien documentado, y relativamente fácil para novatos. A pesar de ser un buen primer lenguaje, no es un juguete; es muy poderoso, flexible y se adapta bien a grandes proyectos. He escrito un detallado [análisis de Python](#). Hay disponibles buenos [tutoriales](#) en el [sitio web de Python](#).

Java es también un buen lenguaje para iniciarse en la programación. Es más difícil que Python, pero produce código más rápido que Python. Creo que es un buen segundo lenguaje para aprender.

Pero no creas que podrás ser un hacker, siquiera un programador, si solo conoces un único lenguaje —debes aprender cómo pensar en los problemas de programación de una manera general, independiente de cualquier lenguaje. Para ser un hacker de verdad, precisas llegar al punto en el cual aprendas un lenguaje nuevo en días, relacionando lo que está en el manual con lo que ya sabe de antes. Esto significa que debes aprender varios lenguajes muy diferentes entre sí.

Si deseas meterte seriamente en programación, tendrás que aprender C, la base de Unix. C++ es muy parecido a C; si conoces uno, aprender el otro no te resultará difícil. Sin embargo, ninguno de ellos es bueno para aprender a programar. Y hoy día, cuanto más puedas evitar programar en C, mas productivo serás.

C es muy eficiente, y trabaja bien con los recursos de las maquinas. Desafortunadamente, C obtiene su eficiencia a base de mucho manejo a bajo nivel de los recursos (como la memoria) a mano. Todo esa programación a bajo nivel es muy compleja y propensa a errores, y tendrás que dedicar gran cantidad de tiempo a la depuración de los programas. Con las máquinas de hoy, tan poderosas, esto ya no es tan necesario —es más inteligente usar un lenguaje que use el tiempo de máquina menos eficientemente, pero en cambio use tu tiempo *más* eficientemente. O sea, Python.

Otros lenguajes de particular importancia para los hackers son [Perl](#) y [LISP](#). Perl vale la pena aprenderlo por razones prácticas: es amplísimamente usado por páginas web dinámicas y por sistemas de administración, así que incluso si nunca has escrito en Perl, deberías aprender a leerlo. Mucha gente usa Perl en la forma que yo sugiero usar Python, para evitar la programación en C en trabajos que no requieran la eficiencia de C. Necesitarás ser capaz de entenderlo.

Vale la pena aprender LISP por una razón diferente — La profunda e iluminadora experiencia que obtendrás cuando finalmente lo logres te hará ser mejor programador el resto de tus días, incluso si no usas LISP mucho. (Puedes obtener fácilmente algo de primera experiencia con LISP escribiendo y modificando los modos de edición del editor Emacs.)

Lo mejor hoy día es aprender estos cinco (Python, Java, C/C++, Perl y LISP). Además de ser estos lenguajes los mas importantes en el hacking, representan muy diversas formas de aproximarse a la programación, y cada uno te educará de diferentes maneras.

No puedo dar instrucciones completas en este documento de cómo puedes aprender a programar —es una habilidad compleja. Pero puedo adelantarte que los libros y los cursos no servirán (muchos, tal vez *la mayoría* de los mejores hackers, son autodidactas). Puedes aprender las características de los lenguajes —partes de conocimiento— de libros, pero el verdadero conocimiento lo adquieres en la vida real aplicando lo que ya sabes. Lo que sí servirá es a) *leer código* y b) *escribir código*.

El aprendizaje de la programación es como aprender a escribir bien en lenguaje natural. La mejor manera de aprender es leer algo de las cosas escritas por los maestros del estilo, luego escribir algunas cosas tú mismo, leer mucho más, escribir un poco más... y repetir esto hasta que lo que escribes empieza a mostrar la clase de fuerza y economía que aprecias en tus modelos.

Antes era difícil encontrar código de buena calidad para leer, porque había pocos

programas de cierta magnitud disponibles en código fuente de tal manera que los hackers noveles pudieran meterles mano. Esta situación ha cambiado drásticamente: software de fuente abierta, herramientas de programación libres y sistemas operativos libres (todo ello hecho por hackers) están ampliamente disponibles en la actualidad. Lo que me lleva elegantemente a nuestro siguiente tema...

2. Consigue uno de los Unices libres; aprende a usarlo y a ponerlo en funcionamiento

Asumo que tienes un ordenador personal o puedes acceder a uno (a estos muchachos de hoy día les resulta tan sencillo obtenerlo... :-)). El paso con mucho más importante que cualquier novato puede dar en el camino de adquirir las habilidades de un hacker es obtener una copia de Linux o alguno de los otros Unices tipo BSD libres, instalarlo en una máquina personal y hacerlo funcionar.

Sí, es cierto que existen otros sistemas operativos en el mundo además de Unix. Pero sólo pueden encontrarse en formato binario —no puedes ni leer el código ni modificarlo—. Si intentas aprender a hackear en una máquina DOS, Windows o en un MacOS te sentirás como si tratas de aprender a bailar con el cuerpo escayolado.

Bajo OS/X es posible, pero solo una parte del del sistema es código de fuente abierta —Es como si estuvieras golpeando un muro de varias capas de cemento, y aparte tienes que ser cuidadoso para no desarrollar el mal hábito de depender del código propietario de Apple. Si te concentras en Unix bajo el capó, puedes aprender cosas mas útiles.

Unix es el sistema operativo de Internet. Si bien puedes aprender a usar Internet sin saber Unix, nunca podrás ser un hacker en Internet sin conocerlo. Por este motivo, la cultura hacker actual está muy centrada en Unix. (Esto no fue siempre así, e incluso la situación le desagrada a algunos de los hackers más veteranos, pero la simbiosis entre Unix e Internet es tan fuerte que hasta la energía de Microsoft parece incapaz de hacerle mella.)

Así que consigue un Unix —personalmente me gusta Linux, pero hay otros (y sí, *puedes* hacer funcionar Linux y DOS/Windows en la misma máquina). Apréndelo. Hazlo funcionar. Métele mano, ámate. Comunícate en Internet mediante él. Lee el código. Modifícalo. En ese sistema se incluyen las mejores herramientas de programación (que incluyen C, Lisp y Perl) de lo que cualquier sistema operativo de Microsoft ni siquiera podría soñar, además te divertirás y te sumergirás en un caudal de conocimientos de tal magnitud que nunca hubieras imaginado, hasta que en cierto momento, mirando hacia atrás, te darás cuenta que ya eres un maestro hacker consumado.

Si quieres saber más acerca del aprendizaje de Unix, ve a [The Loginataka](#).

Si quieres meter tus manos en Linux, echa un ojo en [Dónde puedo obtener Linux](#).

Puedes encontrar ayuda y recursos de BSD Unix en www.bsd.org.

He escrito sobre [los fundamentos básicos de Unix y de Internet](#).

(Nota: No recomiendo instalar ni Linux ni BSD a solas si eres novato. Para Linux, encuentra un grupo de usuarios local de Linux y pide ayuda; o contáctalo mediante

[Open Projects Network](#). LISC mantiene [canales IRC](#) donde puedes obtener ayuda.)

3. Aprende a usar la Web y a programar en HTML

La mayoría de las cosas que ha construido la cultura hacker funcionan fuera de la vista del gran público, ayudando en el funcionamiento de fábricas, oficinas y universidades, y carecen de un impacto obvio en la vida de quienes no son hackers. La Web es la única gran excepción, y es tan enorme y brillante este juguete de hackers que incluso los *políticos* admiten que está cambiando el mundo. Sólo por esta razón (y hay un montón de otras igualmente buenas), debes aprender a trabajar en la Web.

No me estoy refiriendo a aprender a manejar un navegador (cualquiera puede hacer eso), sino que debes aprender a escribir HTML, el lenguaje de marcas de la Web. Si aún no sabes cómo programar, el aprendizaje que implica la escritura de HTML te enseñará algunos de los hábitos mentales que te ayudarán luego con la programación. Así que hazte una página personal. Prueba con XHTML, que es más claro que el clásico HTML. (Existen buenos tutoriales para principiantes en la Web; [aquí hay uno](#).)

Pero no vayas a creer que por tener una página personal estás más cerca de ser un hacker. La Web está llena de páginas personales. La mayoría de ellas son banales, basura sin contenido —basura muy vistosa, pero basura al fin (si deseas saber más sobre este tema, mira en [The HTML Hell Page](#)).

Para que valga la pena, tu página debe tener *contenido* —debe ser interesante y/o útil para los otros hackers. Y esto nos lleva al próximo tema...

4. Si no tienes un inglés funcional, apréndelo

Como estadounidense y hablante nativo de inglés, era reacio a sugerir esto, por si se interpretaba como una suerte de imperialismo cultural. Pero hablantes nativos de otras lenguas me han urgido a que señale que el inglés es el lenguaje con el que trabaja la cultura hacker e Internet, y necesitarás saberlo para funcionar en la comunidad hacker.

Esto es muy cierto. Tiempo atrás, alrededor de 1991, aprendí que muchos hackers que tenían el inglés como segundo idioma lo usaban en discusiones técnicas, incluso cuando compartían su lengua natal; me comentaron que el inglés tiene un vocabulario técnico más rico que cualquier otra lengua y sencillamente por esto era mejor herramienta para el trabajo. Por razones similares, las traducciones de libros técnicos escritos originalmente en inglés son con frecuencia insatisfactorias (si es que se hacen).

Linus Torvalds, un finlandés, comenta su código en inglés (al parecer nunca se le ocurrió hacerlo de otra manera). Su fluidez con el inglés ha sido un importante factor en su habilidad para reclutar una comunidad mundial de desarrolladores de Linux. Y esto nos lleva al próximo tema...

Estatus en la cultura hacker

De manera similar a muchas otras culturas sin economía monetaria, el hackerismo se basa en la reputación. Estás intentando resolver problemas interesantes, pero cuando interesantes y buenas son las soluciones que encuentres es algo que sólo tus iguales o

superiores técnicamente hablando serán capaces de juzgar.

De acuerdo a esto, cuando juegas el juego del hacker, aprendes a puntuarte principalmente en función de lo que los otros hackers piensan acerca de tus habilidades (este es el motivo por el cual no puedes ser un hacker de verdad hasta que otros hackers te denominen así de manera consistente). Este hecho está empañado por la imagen del hacking como trabajo solitario; también por un tabú cultural hacker (que ahora está decayendo, pero aún es fuerte) que impide que se admita el ego o la validación externa como elementos involucrados en la propia motivación.

Específicamente, el hackerismo es lo que los antropólogos denominan una *cultura del don*. Adquieres estatus y reputación no mediante la dominación de las otras personas, ni por ser hermoso/a, ni por tener cosas que las otras personas desean, sino por donar cosas. Específicamente, al donar tu tiempo, tu creatividad, y el resultado de tu destreza.

Existen básicamente cinco clases de cosas que puedes hacer para obtener el respeto de los hackers:

1. Escribir software de fuente abierta

Lo primero (lo más central y más tradicional) es escribir programas que los otros hackers opinen que son divertidos o útiles, y donar los fuentes del programa a la cultura hacker para que sean utilizados.

(Solíamos llamarlo *free software* pero esto confundía a demasiada gente, que no estaban seguros de qué se suponía que significaba *free* [en inglés, el término *free* es polisémico y puede significar "libre" o "gratis". En castellano no se da tal ambigüedad por lo que la expresión "software libre" resulta perfectamente adecuada — N. del T.]. La mayoría de nosotros, al menos un ratio 2:1 de acuerdo con los análisis del contenido de la web, ahora preferimos el término software "[open source](#)" [En castellano, sin embargo, se sigue usando habitualmente "software libre", aunque en esta versión hemos respetado, como es natural, el cambio en la terminología de Eric, que ha sustituido " free software" por *open source* en todos sus documentos, y lo hemos traducido por "código de fuente abierta" — N. del T.]).

Los más reverenciados semidioses del hackerismo son las personas que han escrito programas de gran magnitud, con grandes capacidades que satisfacen necesidades de largo alcance, y los donan, de tal manera que cualquiera pueda utilizarlos.

2. Ayudar a probar y depurar software de fuente abierta

También son reconocidas aquellas personas que depuran los errores del software de fuente abierta. En este mundo imperfecto, inevitablemente pasaremos la mayoría de nuestro tiempo de desarrollo en la fase de depuración. Este es el motivo por el cual los desarrolladores de software de fuente abierta piensan que un buen "beta-tester" (probador de versiones beta, alguien que sabe cómo describir claramente los síntomas, que puede localizar correctamente los problemas, que tolera los errores en una entrega apurada, y que está dispuesto a aplicar unas cuantas rutinas sencillas de diagnóstico) vale su peso en oro. Aun contando con un solo probador de estos, puede hacerse que el proceso de depuración pase de ser una prolongada pesadilla que lo deja a uno exhausto a ser sólo una saludable molestia.

Si eres novato, trata de encontrar un programa en desarrollo en el cual estés interesado, y conviértete en un buen probador beta. Hay una progresión natural desde ayudar a probar programas, luego ayudar a depurarlos y más adelante ayudar a modificarlos. Aprenderás un montón de esa manera, y la gente te ayudará en el futuro.

3. Publicar información útil

Otra buena cosa que puedes hacer es recopilar y filtrar información útil e interesante y construir páginas web o documentos tipo FAQ ("Preguntas Frecuentes"), y ponerlos a disposición de los demás.

La gente que mantiene las FAQ técnicas más importantes gozan de casi tanto respeto como los autores de software libre.

4. Ayudar a mantener en funcionamiento la infraestructura

La cultura hacker (y el desarrollo ingenieril de Internet, para el caso) funciona gracias al trabajo voluntario. Existe una gran cantidad de trabajo necesario pero sin glamour que debe hacerse para que esto siga marchando —administrar listas de correo, moderar foros de discusión, mantener sitios donde se archivan grandes cantidades de software, desarrollar [RFCs](#) y otros estándares técnicos.

La gente que desarrolla estas actividades goza de mucho respeto, porque todos saben que esos trabajos son grandes consumidores de tiempo y no tan divertidos como meterse con el código. Al hacerlos demuestran su dedicación.

5. Hacer algo por la cultura hacker en sí misma

Finalmente, puedes propagar la cultura hacker en sí misma (por ejemplo, escribiendo un texto acerca de cómo transformarse en hacker :-)). Esto es algo que no estarás en disposición de hacer hasta que seas bien conocido en el ambiente por alguna de las cuatro cosas recién descritas.

La cultura hacker no tiene líderes exactamente, pero tiene héroes culturales, ancianos de la tribu, historiadores y portavoces. Cuando hayas estado en las trincheras tiempo suficiente, podrás crecer y transformarte en uno de ellos. Pero ten cuidado: los hackers desconfían del ego ruidoso en sus jefes tribales, así que la búsqueda visible de esa clase de fama es peligrosa. En lugar de esforzarte en ello, es mejor colocarse en una posición tal que ella caiga sobre ti, y luego debes comportarte de manera modesta y con gracia con tu estatus.

La conexión hacker/nerd

Contrariamente al mito popular, no tienes que ser un *nerd* [literalmente "empollón", pero en la [jerga hacker](#) se ha asumido de forma irónica, perdiendo el matiz originalmente despectivo, y ha acabado usándose como sinónimo de alguien que se preocupa por las cosas importantes y no se entretiene en trivialidades. — N. del T.] para ser hacker. Ayuda, sin embargo, y muchos hackers son nerds. Al ser un marginado social, el nerd puede mantenerse concentrado en las cosas realmente importantes, como pensar y hackear.

Por esta razón, muchos hackers han adoptado la etiqueta "nerd" e incluso utilizan el desagradable término [geek](#) como insignia de orgullo —es su forma de declarar su propia independencia de las expectativas sociales normales. Si deseas ver una discusión más extensa, dirígete a [The Geek Page](#).

Si puedes concentrarte lo suficiente en el hacking para ser bueno en ello, y encima disfrutar de vida personal, está bien. Es muchísimo más fácil ahora que cuando yo era un novato en los años setenta; la cultura dominante ve con mejores ojos a los tecno-nerds en la actualidad. Hay además un número creciente de personas que se dan cuenta de que con frecuencia los hackers son un material de alta calidad para novio/novia/marido/esposa.

Si te sientes atraído por la actividad hacker porque no tienes vida propia, está bien también —al menos no tendrá problemas de concentración—. A lo mejor más adelante puedes conseguir una vida propia como el resto de la gente.

Cuestiones de estilo

De nuevo, para ser un hacker, debes desarrollar la mentalidad del hacker. Existen algunas cosas que puedes hacer cuando estás sin ordenador, que pueden ayudarte. Estas cosas no son sustituto para la actividad propia de hackear (nada lo es) pero muchos hackers las realizan, y sienten que de alguna manera primitiva se conectan con la esencia de la actividad de hackear.

- Aprende a escribir correctamente en tu lengua. A pesar de que existe el estereotipo de que los programadores no son capaces de escribir, una cantidad sorprendente de hackers (incluyendo a los mejores que conozco), son escritores competentes.
- Lee ciencia-ficción. Ve a las reuniones sobre ciencia-ficción (es un buen método para conocer hackers y proto-hackers).
- Estudia zen, y/o practica artes marciales. (La disciplina mental se parece en ambos casos.)
- Desarrolla un oído analítico para la música. Aprende a apreciar clases peculiares de música. Aprenda a tocar correctamente algún instrumento musical, o a cantar.
- Desarrolla inclinación por los dobles sentidos y los juegos de palabras.

Cuantas más cosas de estas hayas hecho, es más probable que poseas material natural para hacker. Por qué estas cosas en particular y no otras es algo que no está completamente claro, pero todas están conectadas con un mezcla de tu parte izquierda y derecha de las habilidades de tu cerebro, lo que parece ser una cosa importante; los hackers ser capaces tanto de razonamiento lógico como de dar pasos al margen de lógica aparente de un problema en un momento dado.

Trabaja tan intensamente como juegas y juega tan intensamente como trabajas. Para los verdaderos hackers, la diferencia entre "juego", "trabajo", "ciencia" y "arte" tienden a desaparecer, o mezclarse en un alto nivel de creatividad. Además, no te des por satisfecho con tener un estrecho rango de habilidades. Aunque la mayoría de los hackers

se autodescriben como programadores, suelen ser más que competentes en actividades diversas —administración de sistemas, diseño web y arreglar problemas de hardware son algo común—. Un hacker que, por un lado, es administrador de sistemas, por el otro es también habilidoso en programación de scripts y diseño de webs. Los hackers no hacen las cosas a medias; si se meten de lleno en un tema, tienden a ser muy buenos en él.

Finalmente, unas cuantas cosas que *no* debes hacer:

- No uses un nombre de usuario tonto o grandioso.
- No te metas en "flamewars" (discusiones airadas) en Usenet (ni en ningún otro lado).
- No te llames a ti mismo "ciberpunk", y no pierdas tu tiempo con nadie que lo haga.
- No publiques en los foros ni escribas correos con montones de errores de tecleo o construcciones gramaticales incorrectas.

La única reputación que te ganarás con esas costumbres es la de tonto. Los hackers tienen una gran memoria —puede costar años hasta que te acepten de nuevo después de estas cosas.

El problema con los apodos virtuales merece alguna explicación. Ocultar tu identidad detrás de un apodo es un comportamiento infantil y tonto que es característico de los crackers, warez d00dz, y otras formas inferiores de vida. Los hackers no hacen eso; ellos están orgullosos de lo que que hacen y lo quieren asociado con sus nombres *reales*. Así que si tienes un nombre de estos, abandónalo. En la cultura hacker solo servirán para marcarte como un perdedor.

Otras fuentes de información

Peter Seebach mantiene un excelente [Hacker FAQ](#) para directivos que no entiendan el trato con hackers. Si la pagina de Peter no responde, en la siguiente [búsqueda en Excite](#) encontrarás una copia.

También he escrito [A Brief History Of Hackerdom](#).

He escrito un artículo [The Cathedral and the Bazaar](#) [existe traducción castellana: [La catedral y el bazar](#) — N. del T.], donde se explica bastante sobre cómo funciona la cultura de Linux y del software de fuente abierta. He estudiado este tema más directamente en su segunda parte, [Homesteading the Noosphere](#) [existe traducción castellana: [Cultivando la noosfera](#) — N. del T.].

Rick Moen ha escrito un excelente documento en [how to run a Linux user group](#).

Rick Moen y yo hemos colaborado en otro documento en [How To Ask Smart Questions](#) [existe traducción castellana: [Cómo hacer preguntas de manera inteligente](#) — N. del T.]. Esto te ayudará a encontrar asistencia en el camino.

Si necesitas instrucciones para lo básico como ordenadores personales, Unix e Internet , ve a [The Unix and Internet Fundamentals HOWTO](#).

Cuando publicas software o escribes parches, procura seguir las directrices en el [Software Release Practice HOWTO](#).

Preguntas Frecuentes

- P: [¿Me enseñarás a hackear?](#)
- P: [¿Cómo puedo iniciarme entonces?](#)
- P: [¿Cuándo conviene empezar? ¿Es demasiado tarde para aprender?](#)
- P: [¿Cuánto tiempo me costará aprender a hackear?](#)
- P: [¿Son Visual Basic o C# buenos lenguajes para empezar?](#)
- P: [¿Me ayudarías a crackear un sistema, o me enseñarías cómo hacerlo?](#)
- P: [¿Cómo puedo obtener la contraseña de la cuenta de otra persona?](#)
- P: [¿Cómo puedo acceder/leer/monitorizar el correo de otra persona?](#)
- P: [¿Cómo puedo robar privilegios de operador de canal en el IRC?](#)
- P: [He sido crackeado. ¿Me ayudarás a defenderme de futuros ataques?](#)
- P: [Estoy teniendo problemas con mi Windows, ¿me ayudarás?](#)
- P: [¿Dónde puedo encontrar a hackers de verdad para charlar con ellos?](#)
- P: [¿Me puedes recomendar libros útiles sobre hacking y temas relacionados?](#)
- P: [¿Necesito ser bueno en matemáticas para llegar a ser un hacker?](#)
- P: [¿Qué lenguaje debería aprender primero?](#)
- P: [¿Qué clase de hardware necesito?](#)
- P: [Deseo contribuir. ¿Puedes ayudarme a escoger un problema en el que trabajar?](#)
- P: [¿Necesito odiar y golpear a Microsoft?](#)
- P: [¿Pero no imposibilitará el software de fuente abierta ganarse la vida a los programadores?](#)
- P: [¿Cómo puedo iniciarme? ¿Dónde puedo obtener un Unix libre?](#)

P:
¿Me enseñarás a hackear?

R:
Desde la primera publicación de esta página, me llegan varias peticiones por semana (a menudo varios por día) de gente que quiere que le "enseñe todo sobre hackeo". Desafortunadamente, no tengo ni el tiempo ni la energía para hacer eso; mis propios proyectos de hackeo consumen el 110% de mi tiempo.

Aun si pudiera hacerlo, el hackeo es una actitud y habilidad que debes aprender por ti mismo. Te darás cuenta de que los hackers de verdad quieren ayudarte, pero no te respetarán si comienzas a rogarles que te alimenten en la boca con todo lo que saben.

Aprende unas cuantas cosas primero. Muéstrales que lo estás intentando, que eres capaz de aprender por tu cuenta. Entonces dirígete a los hackers que conozcas pero ya con preguntas específicas.

Si escribes un email a un hacker pidiéndole consejo, hay 2 cosas que debes saber antes. Primero, hemos encontrado que la gente que es perezosa o descuidada en sus

escritos suelen ser demasiado perezosos o descuidados para pensar en ser buenos hackers —Así que ten cuidado en deletrear correctamente, y usar buena gramática y puntuación. En caso contrario, probablemente serás ignorado. En segundo lugar, no *te atrevas* a pedir una respuesta con una cuenta de ISP diferente a la cuenta desde la que estás enviando; nos encontramos gente que lo hacen que suelen ser timadores usando cuentas robadas, y no tenemos interes en ayudar o apoyar el robo.

P:
¿Cómo puedo iniciarme entonces?

R:
La mejor manera para empezar probablemente es pasarte por un LUG ("Linux User Group", un grupo de usuarios de Linux local). Puedes encontrar grupos por ejemplo en la [LDP General Linux Information Page](#); probablemente haya uno cerca de tu localidad, asociado con un colegio o universidad. Los miembros de los LUG probablemente te darán un Linux si lo pides, y ciertamente te ayudarán a instalarlo e iniciarte.

P:
¿Cuándo conviene empezar? ¿Es demasiado tarde para aprender?

R:
Cualquier edad en la cual estés motivado para iniciarte es buena edad. La mayoría de la gente parece interesarse entre los 15 y 20 años, pero conozco excepciones en ambas direcciones.

P:
¿Cuánto tiempo me costará aprender a hackear?

R:
Esto depende de tu talento y lo duro que trabajes. La mayoría de la gente puede adquirir un nivel de habilidad respetable a los dieciocho meses o dos años, si se concentran. Sin embargo, no creas que has finalizado ahí; si eres un hacker de verdad, te pasarás el resto de tu vida aprendiendo y perfeccionando tu material.

P:
¿Son Visual Basic o C# buenos lenguajes para empezar?

R:
No, porque no son portables. No hay ninguna implementación libre de esos lenguajes, así que estarías cerrándote solo a las plataformas a las que el fabricante decide dar soporte. Aceptar esta situación de monopolio no es el camino del hacker.

Visual Basic es especialmente tremendo. El hecho de que sea un lenguaje propiedad de Microsoft es suficiente para descalificarlo, y como otras versiones de Basic tiene un diseño pobre y te enseñará malos hábitos de programación.

Uno de esos malos hábitos es hacerte dependiente de las bibliotecas, módulos y herramientas de desarrollo procedentes de un solo fabricante. En general, cualquier

lenguaje que no esté soportado bajo Linux o un de los BSDs, y/o cuando menos pueda ser ejecutado en 3 diferentes sistemas operativos, es un lenguaje deficiente para aprender a hackear.

P:

¿Me ayudarías a crackear un sistema, o me enseñarías cómo hacerlo?

R:

No. Alguien que pregunta esto después de leer este documento es demasiado estúpido para poder ser instruido, incluso aunque tuviese tiempo para ello. Cualquier correo con este tipo de preguntas lo ignoraré o responderé con extrema rudeza.

P:

¿Cómo puedo obtener la contraseña de la cuenta de otra persona?

R:

Eso es cracking. Desaparece, idiota.

P:

¿Cómo puedo acceder/leer/monitorizar el correo de otra persona?

A:

Eso es cracking. Piérdete, imbécil.

P:

¿Cómo puedo robar privilegios de operador de canal en el IRC?

A:

Eso es cracking. Fuera de aquí, cretino.

P:

He sido crackeado. ¿Me ayudarás a defenderme de futuros ataques?

R:

No. Todo el tiempo me han estado preguntando eso, se debe a la deficiente programación de Microsoft Windows. No es posible proteger a Windows de forma efectiva contra ataques de crackers; sencillamente, su código y su arquitectura tienen demasiados defectos. La única prevención realizable consiste en cambiar a Linux o a algún otro sistema operativo que por lo menos esté diseñado para poder ser protegido.

P:

Estoy teniendo problemas con mi Windows, ¿me ayudarás?

R:

Sí. Ve a una línea de comandos de DOS y teclea: `'format c:'`. Cualquier problema que estés experimentando, desaparecerá en pocos minutos.

P:

¿Dónde puedo encontrar hackers de verdad para charlar con ellos?

R:

La mejor manera es encontrar un grupo de usuarios local de Unix or Linux e ir a sus reuniones (puedes encontrar links a varias listas de grupos de usuarios en el sitio de [LDP](#) en Metalab).

(Antes solía decir que no encontrarías hackers reales en el IRC, pero estoy dándome cuenta que eso está cambiando. Al parecer algunas comunidades de hackers reales, vinculadas a cosas como GIMP y Perl, tienen canales de IRC ahora.)

P:

¿Me puedes recomendar libros útiles sobre hacking y temas relacionados?

R:

Mantengo un [Linux Reading List HOWTO](#) que puedes encontrarlo útil. El [Loginataka](#) podría ser también interesante.

Para una introducción a Python, echa un vistazo a los [materiales introductorios](#) en el sitio web de Python.

P:

¿Necesito ser bueno en matemáticas para llegar a ser un hacker?

R:

No. Aunque necesitas ser capaz de pensar en términos lógicos y seguir cadenas de razonamiento exacto, el hacking usa muy poca matemática formal o aritmética.

En particular, no necesitarás trigonometría, cálculo o análisis (eso se lo dejamos a los ingenieros electricos :-)). Algo de matemáticas finitas (incluyendo álgebra booleana, teoría de conjuntos finitos, combinatoria y teoría de grafos) puede ser útil.

P:

¿Qué lenguaje debería aprender primero?

R:

XHTML (el dialecto más reciente de HTML) si todavía no lo sabes. Existe muchos *muy malos* libros de HTML afuera, y desafortunadamente solo algunos buenos. El que a mí me parece el mejor es [HTML: The Definitive Guide](#).

Pero HTML no es un lenguaje de programación completo. Cuando estés listo para empezar a programar, recomendaría empezar con [Python](#). Escucharás a mucha gente que te recomiende Perl, y Perl es aún más popular que Python, pero es mas difícil de aprender y (en mi opinión) peor diseñado.

C es realmente importante, pero es tambien mucho mas difícil que Python y Perl. No intentes aprenderlo primero.

Usuarios de Windows: *no* os conforméis con Visual Basic. Os enseñará malos hábitos, y no es portable fuera de Windows. Evitadlo.

P:
¿Qué clase de hardware necesito?

R:
Solía pasar que los ordenadores personales tenían pocas prestaciones y poca memoria, lo cual impuso límites artificiales en el proceso de aprendizaje de un hacker. Esto dejó de ser cierto hace algún tiempo; cualquier máquina de Intel 486DX50 o superior es lo suficientemente poderosa para el el trabajo de desarrollo, X y conexión a Internet, y los disquetes más pequeños que puedes comprar hoy son suficientes.

Lo importante al escoger una máquina para aprender es si su hardware es compatible con Linux (o compatible con BSD). Esto vale también para las máquinas más modernas. La única cuestión complicada son los modems: algunas máquinas tiene hardware específico para Windows que no funciona con Linux. [Hay un sitio web llamado linmodems.org donde podrás determinar si, pese a ser un "winmodem", puedes conseguir que tu dispositivo funcione bajo GNU/Linux — N. del T.]

Hay una FAQ acerca de la compatibilidad de hardware; la última versión está [aquí](#).

P.:
Deseo contribuir. ¿Puedes ayudarme a escoger un tema en el que trabajar?

R.:
No, porque desconozco cuáles son tus talentos o intereses. Debes estar automotivado o no te dedicarás seriamente a ello, por eso casi nunca funciona que otras personas elijan por ti.

Prueba esto. Mira los anuncios de [Freshmeat](#) durante algunos días. Cuando veas uno que te haga pensar: "¡Guay, me gustaría trabajar en eso!", únete.

P:
¿Necesito odiar y golpear a Microsoft?

R:
No, no hace falta. Y no porque Microsoft no sea asqueroso, sino porque la cultura hacker existía mucho antes que Microsoft y seguirá existiendo aún largo tiempo después de que Microsoft sea historia. Cualquier energía que gastes odiando a Microsoft sería mejor empleada entusiasmándote con tus habilidades. Escribe buen código —esto será un golpe más duro para Microsoft y no requiere que ensucies tu karma.

P:
¿Pero no imposibilitará el software de fuente abierta ganarse la vida a los programadores?

R:

Esto parece improbable —hasta ahora la industria de software de fuente abierta parece estar creando empleos más que quitándolos. Si tener un programa escrito en una economía de red gana sobre no tenerlo escrito, un programador será remunerado independientemente de si el programa va a ser código abierto después de realizado. Y no importa cuánto software "gratis" tienes escrito, parece que siempre habrá más demanda para aplicaciones nuevas y especializadas. He escrito más sobre esto en las páginas de [Open Source](#).

P:

P: ¿Cómo puedo iniciarme? ¿Dónde puedo obtener un Unix libre?

R:

En otro lugar de esta página incluyo punteros a lugares donde puedes obtener el Unix libre más usado. Para ser un hacker necesitas motivación y la iniciativa y la habilidad para educarte por ti mismo. Empieza ahora...