UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
INSTITUTO DE INVESTIGACIONES CULTURALES-MUSEO

SOMBREROS BLANCOS: TRES HACKERS PROGRAMAN UN MUNDO MEJOR

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRA EN ESTUDIOS SOCIOCULTURALES

PRESENTA:

MIGUEL ÁNGEL LOZANO CHAIREZ

BAJO LA DIRECCIÓN DE

DR. ARTURO ONGAY FLORES

MEXICALI, B.C., NOVIEMBRE DE 2012.

Sombreros blancos: Tres hackers programan un mundo mejor

Miguel Ángel Lozano

Instituto de Investigaciones Culturales-Museo

Universidad Autónoma de Baja California

Resumen

Los *hackers* han capturado nuestra imaginación a través de memorables imágenes en televisión, cine y el periodismo. Los medios los presentan como criminales informáticos expertos; a veces como activistas que buscan la libertad en el Internet. Pero no todos son delincuentes: los *hackers* de "sombrero blanco" usan de manera innovadora la tecnología. El propósito de esta tesis es describir sus motivaciones y valores éticos. Un camino es conocer y ahondar sus prácticas. En este estudio colaboraron tres *hackers*, de quienes conocí las prácticas a través de la observación participante y entrevistas a profundidad. Mediante el análisis del discurso comprendí la dimensión ética que tienen con respecto a la tecnología. Los *hackers* comparten valores; formas de trabajar y una visión del mundo que lucha por una mayor libertad para todos. Su confianza en los ideales y el trabajo comunitario los asemeja más a un culto religioso que a los espías de la era digital retratados en el cine.

E-Mail: el.badbit@gmail.com

Director de tesis: Dr. Luis Arturo Ongay Flores

Índice

Agradecimientos

Esta tesis es el producto de un arduo y extenso trabajo. Hubiera sido imposible emprender este trabajo en solitario. Por tanto, quisiera mencionar y agradecer a las personas e instituciones que se involucraron directa e indirectamente en la construcción de esta tesis. Al Consejo Nacional de Ciencia y Tecnología por otorgar la beca que permitió que esta investigación fuera posible. A mis compañeros de maestría, quienes me acompañaron en el proceso de sufrir y descubrir este campo nuevo para nosotros: los estudios socioculturales. En especial a Morella Alvarado por sus consejos y visión fresca de las cosas que me resultaban ya demasiado familiares.

A mis maestros, en especial al Dr. Servando Ortoll, por sus acertados consejos sobre la redacción y la lógica de este documento. También por ayudarme a redactar el subtítulo de esta tesis. Al Dr. Luis Ongay por ser un director de tesis abierto y comprensivo que me permitió la libertad de cometer mis propios errores. Al Dr. Christian Fernández, por la retroalimentación que sin duda mejoró la investigación.

A todos los *hackers* que conocí durante el trabajo de campo. Primero a aquellos que no fueron informantes pero que mostraron un gran interés por esta tesis y contribuyeron con comentarios y sugerencias, en especial a Carlos Duarte y Jacobo Nájera. Pero también a los tres que si fueron informantes y tuvieron la amabilidad de prestarme su tiempo, atención y brindar su confianza en el proyecto.

Al grupo mxlOpenSource, por abrirme las puertas y ayudar desinteresadamente al desarrollo de esta tesis, en especial a Antonio Antillón y Jesús Manuel Olivas. Al *hackerspace* Noisebridge y todas las personas que conocí dentro de él, por la oportunidad de enriquecer este trabajo, en especial a Robert Young y Liz Henry.

A varios estudiantes de generaciones pasadas de la Maestría de Estudios Socioculturales que me ayudaron con consejo y auxilio en momentos desesperados. En especial a Víctor Gruel y Mario Bogarín.

A mi familia que toleró mi ausencia y poca atención al estar enfrascado en lecturas,

trabajo de campo y redacción del trabajo.

Por último, un muy especial agradecimiento a Carla Maldonado compañera de maestría y amiga. No sólo por sugerir el título de la tesis, también por acompanarme en los momentos más difíciles y darme los consejos más honestos por dolorosos que fueran. Algo de lo más grato de haber cursado la maestría fue haberla conocido.

A todos los demás, gracias. Incluso aquellos que no se interesaron demasiado en el proyecto: si no estorbaron, se los agradezco mucho.

La historia de un recetario

Cuando voy a un restaurante con mi mamá, ella generalmente tuerce el gesto cuando le sirven el plato. Lo hace cuando encuentra algo mal con el servicio. A veces el mesero toma el plato con el pulgar en la orilla y contamina la comida. A veces los bordes del plato están manchados de comida. En ocasiones la decoración no es la adecuada o el sabor de la comida no es bueno. Desde que estudia gastronomía, siempre hay algo mal, pero creo que ya me acostumbré. Supongo que es el riesgo de saber mucho sobre un tema: uno se vuelve quisquilloso, sabe lo que está mal y prefiere las cosas bien hechas.

Durante sus estudios mi mamá ha acumulado una gran cantidad de recetas de cocina de todo tipo. Tanto en papel como en la computadora, fotocopias y libros. Pronto se enfrentó con el problema de organizarlas y clasificarlas: perdía mucho tiempo para encontrar la receta adecuada. A veces requería algunas con ingredientes específicos; otras para una temporada determinada y otras más porque son sus favoritas. Pensó que algún software podría ayudarle en ese trabajo.

Soy ingeniero en sistemas computacionales, así que con frecuencia me pide ayuda para este tipo de problemas. No siempre tengo tiempo para resolverlos, pero lo intento. Encontramos algunos programas disponibles en el mercado, pero ninguno la satisfacía por completo. La mayor parte de los programas se limitaba a almacenar la lista de ingredientes y el procedimiento de preparación. Pero ella también quería calcular costos de un banquete y conocer el contenido calórico de un platillo con base en sus ingredientes.

Decidió entonces recurrir a un programador que desarrollara un software a su medida. Encontró una pequeña empresa ubicada precisamente a unas cuantas cuadras de mi casa. Agendó una cita y me pidió que asistiera en papel de "intérprete" ya que ella no es ninguna experta en computadoras. Con mi presencia quería evitar que la engañaran y también expresar mejor sus necesidades al programador.

Acudimos a la oficina y aparentemente se estaban mudando ya que todo se encontraba en desorden. Nos mencionó el programador algunos de sus proyectos anteriores: sistemas

de cobro para restaurantes, bases de datos sencillas y sistemas para pequeñas y medianas empresas. Pregunté el lenguaje de programación que manejaba y respondió que Visual Basic. Pregunté también que sucedería si mi mamá requería cambios al software en un futuro, y comentó que los podía implementar con un costo adicional.

Estas circunstancias no me agradaron en absoluto. Aprendí Visual Basic cuando cursaba la preparatoria. Es un lenguaje de programación creado por Microsoft; ya obsoleto cuando estudiaba ingeniería. Este tipo de lenguajes poseen reglas de sintaxis y de redacción que sirven para facilitar el dar instrucciones a la computadora. Una vez redactadas, se traducen al lenguaje nativo de una máquina por un software llamado compilador. De ahí en adelante, cualquier usuario puede ejecutar el software sin necesidad de saber programación o de poseer el compilador.

Visual Basic funciona bien para aplicaciones pequeñas como la de mi mamá. Pero con el tiempo, si las recetas llegan a más del millar, es posible el programa no funcione tan bien como al principio. Por otra parte, es posible que pronto el recetario fuese incompatible con futuros sistemas operativos, y deje de funcionar pronto.

Pero lo que más me preocupaba de la propuesta del programador es que dependeríamos de él toda la vida. Para cambiar el programa se requiere el llamado "código fuente", que son las instrucciones escritas en el lenguaje original. Estoy familiarizado con Visual Basic, podría implementer cambios pequeños si lo quisiera; pero es imposible si el programador mantiene el código del software en secreto.

Tampoco podría a pedirle a nadie más que lo modificara, ya que es necesario el código. ¿Qué tal si la pequeña empresa desapareciera repentinamente? Nos quedaríamos botados con un software que no podríamos actualizar ni modificar. Este modelo es conocido como "software propietario" y es utilizado por muchas corporaciones como Microsoft y Apple Inc.

Le comenté estas inquietudes a mi mamá, pero le urgía el programa. No tenía tiempo de hacerlo por mí mismo: estudiaba la maestría que culminó con esta tesis, así que no pude alegar demasiado. El programador presupuestó en 600 dólares el costo para desarrollar el

software. Mi madre aceptó. Quedó acordado y supuestamente se reportaría después de un mes para mostrar avances y solicitar el primer pago. No supimos más de él.

Con la necesidad encima, ella volvió a preguntarme si conocía otra opción. En esas fechas asistía a las reuniones de un grupo de programadores de software libre; conducía observación participante para esta investigación. Se trata de una pequeña comunidad que se reúne cada mes con el objetivo de compartir conocimiento y ofrecer talleres. El software libre se distingue del software comercial en que no tiene propietario. Cualquier usuario puede ver el código, compartir los programas y adaptarlos a sus necesidades.

Pensé que dentro de los programadores del grupo, habría alguno interesado en ganar 600 dólares haciendo ese pequeño trabajo. Al final de una de las reuniones, antes de irme, solté tímidamente la idea. Expliqué todo lo que acabo de escribir, y pregunté si alguien estaba interesado en "aventarse la chamba". Todos se vieron unos a otros y se guardó un silencio incómodo. Sentí que lo que propuse no encajaba con el espíritu de las reuniones.

Finalmente un ingeniero de software soltó una idea: "¿qué tal si tomamos ese proyecto como *pet-project*?". Es decir, adoptar grupalmente el proyecto. Esto implica desarrollarlo, aprender todos durante el proceso y, una vez terminado, publicar el programa como software libre. La idea entusiasmó a todos los presentes. Era una nueva oportunidad de aprender, pero también aportar a la sociedad: cualquier persona con la misma necesidad utilizaría el software para resolver su problema sin costo alguno. Esto beneficiaría directamente a los compañeros de clase de mi mamá.

Mientras manejaba de regreso a casa pensé en lo que acababa de presenciar y la gran diferencia entre los mundos del software comercial y el software libre. Cualquiera de los programadores de la reunión pudo tomar el trabajo y ganar 600 dólares. En vez de ello optaron por formar una comunidad: aprender todos y liberar un programa útil para personas con necesidades similares.

Este es el mundo de los *hackers* de sombrero blanco. Programadores entusiastas que creen en el trabajo comunitario y en la libre compartición del conocimiento. Un mundo donde

no hay códigos secretos y los desarrolladores de software están disponibles la mayor parte del tiempo preguntar dudas y sugerir cambios. Los orígenes de esta cultura se remontan hasta los años sesenta, en el seno de algunas universidades estadounidenses. Hasta la fecha existen numerosas prácticas y actividades con ese origen común.

Es difícil hablar de *hackers* que no son criminales. Para muchos esta idea resulta difícil de aceptar. Los medios de comunicación utilizan con demasiada frecuencia el término para hablar de delincuentes informáticos. Durante el transcurso de esta investigación monitoreé los medios para ver lo que se dice de los *hackers*. Casi el 90 por ciento de las noticias se relaciona con delitos en Internet. El cine tampoco pinta una realidad muy diferente. Películas como *The Matrix* o *Swordfish* nos cuentan historias de espías o "ciberpunks" capaces de hacer magia con las computadoras. Pensamos también en adolescentes que viven en un sótano frente a una computadora y entran ilegalmente a los sistemas de la CIA y la NASA. Se les comparara con espías, punks, ninjas, piratas, vaqueros y pandilleros.

Imposible negar que estos reportajes tienen un núcleo de verdad. Las intrusiones no autorizadas existen; los delitos informáticos están a la orden del día y jóvenes se dedican a la piratería de software que se autonombran "*hackers*". Todo eso existe, pero siempre hay una línea marcada entre los *hackers* llamados de "sombrero blanco" y otros frecuentemente denominados de "sombrero negro": aquellos interesados en la exploración ilegal y la piratería. Esta es una analogía tomada del western en cine, donde con frecuencia los villanos usan sombrero negro y los héroes un sombrero más claro.

Durante esta investigación, que duró aproximadamente dos años, conocí algunos *hackers* de sombrero negro. En un principio pensé que serían más interesantes que los desarrolladores de software libre, pero el tema presentó una multitud de dificultades. No muchos querían hablar sobre sus prácticas, poco éticas en muchos casos e ilegales en otros. Cuando se animaban a revelar información, la paranoia los invadía. Creían que serían identificados al publicarse esta investigación. Además, cada caso era muy diferente; encontrar elementos en común fue complicado.

Pronto conocí *hackers* de sombrero blanco y el tema tomó forma. Su abierta manera de expresión me reveló que sus valores son mucho más claros. Comparten valores particulares, hay una clara ética detrás de sus acciones. Comprendí que el software libre no sólo son programas que se descargan gratis de Internet. Es una práctica social que gana adeptos a lo largo del mundo.

En otra de las reuniones del grupo de programadores, asistió un nuevo participante. Al presentarse, comentó que se dedicaba al desarrollo de software desde hace más de 15 años y que llegaba al grupo para actualizar su conocimiento. En la segunda sesión a la que asistió, propuso una idea: "Oigan, yo veo que aquí hay muchas personas con habilidades diferentes y con mucha capacidad. ¿No han pensado en iniciar una empresa? Podríamos pensar en una idea para un producto, desarrollarla entre todos y venderla".

Algunas personas se entusiasmaron con la idea, pero uno en particular, Fernando Michel, se negó rotundamente: "Yo no estaría interesado en hacer eso; yo no vengo aquí por eso", dijo. Alguien le preguntó: "¿Entonces lo haces por hobby?". Fernando respondió casi ofendido: "no, tampoco es por hobby. Lo que yo hago de tecnología y el motivo por el cual estoy aquí es el de aportar a la comunidad". El programador que propuso la idea original se desconcertó visiblemente. Nos observaba como recién llegado a otro planeta. Ya lo volvimos a ver.

Ejemplos como ese hay muchos. ¿Qué es lo que motiva a estas personas a renunciar a la recompensa económica en aras del trabajo comunitario? Es fácil recordar a las grandes figuras de la informática como Bill Gates o Steve Jobs, quienes se enriquecieron vendiendo sus productos. Bill Gates se convirtió en el hombre más rico del mundo impulsando un sistema comercial de licencias para software. Estamos en un sistema capitalista, en la "era de la información" como afirma Manuel Castells (2006), las oportunidades de sacar provecho económico de la tecnología informática abundan. ¿Por qué los *hackers* de sombrero blanco no prefieren ser millonarios?

Así como existen estas dos figuras del software comercial, existen otras dentro del software libre. *Hackers* con la misma, o mayor, capacidad técnica que Gates y Jobs que

decidieron no vender sus creaciones. Richard Stallman, el inventor del software libre, es uno de ellos. Para él el software comercial es una trampa diseñada para mantener a los usuarios divididos e impotentes (Stallman, 2009). Por ello decidió nunca trabajar en proyectos de ese tipo. Incluso ha impartido conferencias en varios países, incluído el congreso de México, explicando la importancia de instalar software libre en las computadoras del gobierno y escuelas.

Linus Torvalds es otro ejemplo destacable. Desarrolló por gusto un sistema operativo llamado Linux. Dicho proyecto posee en la actualidad millones de usuarios en todo el mundo y es la base para infraestructuras tecnológicas muy complejas. El software se ofrece de forma totalmente gratuita y cualquiera puede aportar para mejorarlo. Nadie es dueño de Linux, ni el mismo Torvalds (Wayner, 2000; Torvalds y Diamond, 2001). Linus tiene su empleo de ocho horas, y durante su tiempo libre se encarga de ser el líder del proyecto.

Esta tesis de estudios socioculturales trata específicamente sobre la historia de tres *hackers* de sombrero blanco. Uno de Mexicali, otro de Ensenada, Baja California y otro de Sebeka, Minnesota, pero residente de San Francisco, California. Los tres poseen un conocimiento técnico muy avanzado, una curiosidad enorme por la tecnología y los tres aportan a proyectos de software libre. Pero antes de explicar el orden de este documento y lo que se puede esperar de este trabajo, debo hablar de un cuarto *hacker*: el autor de esta tesis.

En el transcurso de la maestría aprendí que en este tipo de investigaciones es importante que el autor se presente ante sus lectores. ¿Cuál es su historia? ¿Por qué escogió este tema de investigación? ¿Cómo fue su vida? Todo esto ayuda al lector a comprender el trabajo y de quién proviene. También cargo con mi propia cultura, vivencias, prejuicios y conocimientos. Con base en ellos se ha conformado mi visión del mundo, lo que influyó en cómo me aproximé a esta investigación; es una tarea responsable señalarlo. Es hora que cuente un poco de mi historia, antes de presentar el resto del trabajo.

Breve historia de un hacker *frustrado*

Hay algo en las computadoras que me cautiva. No sé muy bien qué es. Quizá su lógica y precisión. Desde la educación preescolar me atraían como imanes. Mis memorias de la época son bastante borrosas, pero recuerdo que cuando entré a la escuela primaria ya había utilizado el sistema operativo MS-DOS. Éste se maneja mediante comandos del teclado, sin interfaz gráfica. Sabía cómo insertar acentos dentro de los procesadores de texto de la época. No había muchos teclados en español: uno debía aprender códigos numéricos para introducir los caracteres especiales como la "eñe".

Mi papá es psicólogo e imparte clases en universidad. Cuando me llevaba a su trabajo yo usaba las computadoras, principalmente la de su oficina. Era tan chico que no tenía tareas importantes ni idea de cómo redactar documentos: las usaba para divertirme. En el laboratorio de cómputo de la universidad me copiaban juegos en disquetes de $5\,\frac{1}{4}$ de pulgada. Aprendí los pasos para encender la máquina y jugarlos. Las computadoras fueron para mí, antes que nada, un juguete.

Con el tiempo aprendí más o menos cómo funcionan, cómo guardan los archivos y procesan las instrucciones. Tomé cursos de computación para niños. Desde luego, seguía jugando en las computadoras, pero cada vez me divertía más aprender sobre la máquina y su software. Era un niño raro, lo admito. Cuando quería desesperadamente una computadora, a mis compañeros de clase no les divertían tanto como a mi. No veían mucha utilidad: "no la necesito", era la respuesta más común.

Quizá yo tampoco la necesitaba: la quería. La primera llegó a mi casa cuando cursaba sexto de primaria. Para los estándares actuales, no tenía siquiera la capacidad de un teléfono celular, pero me impresionó mucho. Esto fue a principios de los noventa, en el momento cuando se iniciaba uno de los varios *booms* de la computación personal. Los negocios de venta y reparación de computadoras surgieron por doquier y las cifras de ventas rompían records anualmente. El periódico local inició una sección semanal dedicada a la informática, llena de anuncios de computadoras. Las revistas sobre el tema tapizaban la mayor parte de

los puestos de revistas y surgían más cada mes.

La Internet llegaba a las masas y las especulaciones sobre su futuro eran muchas. Lo usé por primera vez en 1994, en la universidad en donde trabajaba mi papá. Escuchaba que era enorme, que era internacional y no tenía fronteras; que la información se encontraba disponible para cualquiera. No tenía duda en mi mente sobre lo que quería estudiar: mi vida se dedicaría a la informática.

Mi primer contacto con el mundo *hacker* se dio cuando cursaba segundo año de secundaria. Uno de mis amigos de otro grupo se acercó corriendo al salón con unas hojas impresas en la mano. Las sacudía emocionado y exclamaba que había entrado a la Internet a investigar "cómo *hackear*". "¡Vamos a ser *hackers*!" me dijo emocionado. Aún recuerdo que contenían información sobre dispositivos para modificar tarjetas de crédito y teorías de acceso no autorizado a satélites.

Aunque esto puede sonar a trama de película de ciencia ficción, ahora me divierte la información que contenían esas hojas. Era totalmente inofensiva: no había detalles técnicos ni una serie de pasos a seguir. La página intentaba vender esos dispositivos, y ahora que lo pienso, probablemente era una estafa. Lo impactante fue enterarme que, buscando bien, ese tipo de información podría encontrarse en la Internet. Esto alimentó un fuerte interés en mí para aprender lo que en ese entonces yo creía que era el *hacking*. Estoy seguro de que no era la primera vez que escuchaba de estos personajes expertos y misteriosos. Probablemente los vi en el cine y las noticias, o quizá en películas como *Mission: Impossible*.

Conocí el *underground* informático al buscar por mi cuenta y al obtener acceso a la Internet donde me fuera posible. Una gran fuente de información fueron las revistas electrónicas (*e-zines*) que publicaban digitalmente varios grupos de *hackers*. Se descargaban gratis (en ese entonces nada costaba en la Internet) y detallaban actividades tecnológicas, también se reflexionaban y comentaban sobre temas de interés para ellos. Se enviaban saludos entre ellos y buscaban crear vínculos de comunicación con otros *hackers*.

Entré a una escuela de computación cuando tenía 15 años y cursaba la secundaria.

Sólo aceptaban a estudiantes de preparatoria o adultos. Uno de los subdirectores dijo que me examinarían para verificar mis conocimientos y verificar si estaba listo para entrar en esa escuela. Fui al examen sin estudiar: ¿qué podría haber repasado? Si me preguntaban sobre cosas que no sabía, pues lástima. Pero en ese momento me sentía bastante seguro sobre mis conocimientos.

Al recordar creo que aplicaron el examen como una mera formalidad. Lo que pretendían demostrar es que yo carecía del nivel suficiente para entrar a la escuela. El examen fue relativamente avanzado para que fracasara y quitarse el problema de encima. Nunca esperaron que respondería todas las preguntas bien, que fue lo que sucedió. El subdirector lució un rostro de sorpresa que me sorprendió a mí aún más. A regañadientes y sin más pretextos para esgrimir, aceptó inscribirme en la escuela y ubicarme en un nivel más avanzado.

Quizá no fue buena idea. Utilizaba las computadoras de la escuela como mi laboratorio personal. Cada vez que conseguía un programa nuevo o aprendía una técnica que parecía insegura, la intentaba primero ahí. Varias veces dejé las computadoras del laboratorio inservibles por culpa de mis experimentos. Nada grave, todo se arreglaba borrando todo el disco duro y reinstalando el sistema (en ese tiempo Windows 95), pero ahora veo las increíbles inconveniencias y el tiempo que les hice perder a los laboratoristas. Obviamente nunca supieron que fui yo y si no me leen, nunca lo sabrán. Aprendí mucho de esta experiencia.

Veía que los *hackers* que yo admiraba tenían pseudónimos. Escogí uno para mi: "*BadBit*". Un bit es la unidad más pequeña de información que puede manejar una computadora. Representa un dígito binario: cero o uno. Estos dígitos se agrupan en paquetes de ocho para formar un *byte*, unidad que ya tiene significado. Los bytes se agrupan y conforman información más compleja. Pero si un bit está incorrecto, o erróneo, toda la información se corrompe. Por eso "*BadBit*", un bit "malo", erróneo, afecta un sistema mucho más grande. Era mi manera de expresar que una mínima participación afecta algo muy complejo.

Posteriormente aprendí una versión visual de BASIC desarollada por Microsoft, de nombre Visual Basic, que mencioné antes. Al ver las posibilidades del lenguaje, quise crear

mi propia *e-zine*. Amigos de mi ciudad se ofrecieron a ayudarme cuando tenía 16 años. Escribieron artículos y armé en Visual Basic una pequeña aplicación que sirviera como lector para los usuarios. La publicamos cada dos meses y continuamos durante dos años. En retrospectiva lo veo como un intento por emular a nuestros héroes y no tanto como un aporte significativo a la comunidad *hacker*.

Figura 1. **:** Imagen de "El Radiaktivo Newz", la *e-zine* para Windows que publiqué durante dos años. Este es el número de enero de 2000. El visor está programado en Visual Basic.

Estudié ingeniería en sistemas computacionales y descubrí que mi perspectiva laboral no lucía ni remotamente interesante en comparación a los proyectos que emprendía por gusto. Dentro de la carrera conocí el sistema operativo Linux y aprendí sobre el mundo del software libre. En un punto reemplacé todas las aplicaciones de mi computadora por software libre y me sentí muy satisfecho conmigo mismo. En mi universidad no habíamos muchos usuarios de Linux en ese entonces, recuerdo unos cinco o seis de mi generación.

Después de titularme en ingeniería impartí clases de informática en preparatoria. No me interesó trabajar en una maquiladora, la principal opción laboral en Mexicali. Los otros empleos me parecieron demasiado monótonos y poco creativos. Enseñar significaba, además, tiempo para seguir estudiando.

Los medios de comunicación también me interesan mucho. Por ello decidí iniciar mi propia *e-zine*. Dado que deseaba aprender más, me inscribí a la licenciatura en ciencias de la comunicación. El choque con un campo tan diferente me reveló la gran distancia cultural

con respecto a la ingeniería. Por ello no existen muchos estudios desde las ciencias sociales sobre tecnología. Mientras la ciencia computacional se enfoca en los algoritmos, los números, los procesos secuenciales y los sistemas de información, las ciencias sociales se preocupan por el individuo, la sociedad y los fenómenos que involucran al ser humano de manera central.

También tengo otros intereses. Me gusta la fotografía. Disfruté al fotografiar los eventos a los cuales asistí y las personas que conocí durante la investigación. La literatura me agrada, tanto leerla y escribirla. También disfruté al redactar y narrar este documento. En general me gusta aprender. Esa es la gran motivación detrás de este trabajo. Busqué durante mucho tiempo las mejores herramientas para obtener mayor conocimiento sobre este tema y compartirlo con los demás.

La relevancia del tema de los *hackers* desde una perspectiva social y cultural fue evidente después de leer dos libros: *Hackers* de Steven Levy y *The hacker ethic* de Pekka Himanen. Mencionaré ambos a mayor profundidad más adelante. Poco después de leerlos, supe de la maestría en estudios socioculturales. Ésta me pareció la oportunidad ideal para abordar el tema. Mi intención inicial era refutar las imágenes exageradas que aparecen en los medios, pero terminé encontrando una forma diferente de entender el mundo. En parte también me reencontré a mi mismo y volví al ámbito informático de manera indirecta, después de abandonarlo durante años.

Espero que esta breve introducción sirva al lector para comprender que no soy totalmente ajeno a la cultura *hacker*. Al hablar con ellos empatizaba mucho y por ello lograron abrirse más. El conversar con alguien que comparte algunos de tus ideales y tu historia de vida por lo general es más fácil que con alguien totalmente extraño. Esto también fue un reto para mí: constantemente corría el riesgo de estar demasiado convencido por lo que ellos me contaban. Mi tarea no consistió en convertirme en *hacker*, más bien describir la realidad de manera científica.

Sobre el orden de este documento

Dividí este trabajo en tres capítulos. En el capítulo I presento una revisión exhaustiva de los trabajos previos sobre *hackers* desde una perspectiva social o cultural. Inicié con este capítulo ya que, como mencioné, una de las principales dificultades al discutir mi tema es explicar qué es un *hacker*. Tengo que luchar contra las imágenes negativas de las películas y explicar que esas son, en su mayor parte, ficciones. Espero que este capítulo aclare la cuestión, incluso para aquellos sin conocimiento previo sobre el tema. También intento marcar la diferencia entre los *hackers* de sombrero negro y los *hackers* de sombrero blanco.

En el capítulo II explicaré en detalle el marco teórico en el cual basé mi investigación: los estudios socioculturales y la sociología de la tecnología. Mostraré cómo la tecnología tiene un importante componente cultural. De hecho, la tecnología es una de las manifestaciones culturales más antigüas. Es una de las características que nos distingue de los animales, y nos hace ser humanos. También, en base a ello, se explica la metodología que elegí para estudiar el fenómeno: el método etnográfico y el análisis del discurso. Además, expondré mis criterios para la selección de los informantes.

El capítulo III constituye la etnografía de los tres *hackers*. Mi descripción de cada uno responde a un objetivo específico de la investigación. En ese capítulo narro las circunstancias en que los conocí y por qué llamaron mi atención. Además, conformé una pequeña biografía de ellos en base a lo que me contaron en las diversas entrevistas.

Posteriormente presento las conclusiones: la descripción del mundo que ellos buscan y desean. Esta sección la basé en el análisis de los datos y de los resultados que obtuve de todo el trabajo anterior. Posteriormente introduzco la bibliografía, un índice de imágenes y otro índice alfabético con palabras clave que ayudan a navegar este documento con mayor facilidad. Espero que estos índices se encuentren lo suficientemente ordenados para auxiliar a todo aquel que quiera darse un panorama general de este trabajo, incluso llegar a un punto de particular interés.

Resulta un tanto extraño iniciar con un recuento de los trabajos anteriores sobre el

tema cuando el lector todavía no conoce los objetivos de investigación. Aunque sea extraño, creo que es la mejor manera para iniciar. De esta manera tendrán mucho más sentido las decisiones metodológicas que tomé. Por ello, presento aquí los objetivos de la investigación, para no mantener al lector en blanco.

Objetivo general

Comprender las prácticas sociales de los *hackers* para determinar su postura ética ante las tecnologías de la información.

Objetivos específicos

- Analizar las prácticas tecnológicas de *hackers*.
- Determinar las postura política frente a la tecnología de los *hackers* a través de sus discursos.
- Analizar los usos sociales de la tecnología desde una perspectiva ética.

Con estas consideraciones puestas por escrito, puedo comenzar a presentar la información.

Capítulo I - Sombreros blancos, sombreros negros

Durante mi juventud leí muchos textos sobre *hackers*, pero todo proveniente de revistas y libros creados por ellos mismos. Al iniciar mi investigación pronto me fue obvio que ese conocimiento sería insuficiente para un trabajo académico. No está sistematizado científicamente, no es analítico con respecto a la cultura y representaría únicamente una visión parcial.

Al buscar otra bibliografía, descubrí que abundan los libros sobre el *hacking* desde un punto de vista puramente técnico. En ellos, el término se utiliza de formas tan diversas que al revisarlos se dificulta obtener una idea clara de su significado. Otro trabajo de investigación podría centrarse en dichos textos para obtener un panorama de qué manifestaciones se engloban bajo este término, pero no era el camino que deseaba tomar. Existen pocos estudios sobre estas actividades desde una perspectiva social o cultural. A pesar de esto, las aproximaciones que existen sientan suficientes necesarios para emprender una nueva investigación sobre el tema.

En toda investigación se requiere conocer el trabajo anterior; la revisión del estado del arte es uno de los primeros pasos. Por ello, en este capítulo expondré la literatura existente en las ciencias sociales sobre los *hackers*. Esto colocando cada publicación en su contexto específico. Lo anterior permitirá conformar una idea de cómo el *hacker* ha sido comprendido a lo largo de los años.

Finalmente, en dos pequeños apartados exploro la diferencia entre los *hackers* de sombrero negro y los *hackers* de sombrero blanco: así defino más específicamente el objeto de estudio de esta tesis.

Estudios previos

El trabajo seminal, citado por todos los investigadores posteriores, es el libro de periodismo cultural *Hackers, Heroes of the Computer Revolution* (1984) de Steven Levy. El autor se introdujo al tema por órdenes de su editor, sin haber tocado una computadora en

su vida. Su libro se publicó cuando la guerra fría todavía era una realidad cotidiana; cuando las computadoras personales llegaban a los hogares de muchos usuarios por primera vez.

Se trata del primer trabajo descriptivo que intenta comprender al *hacker*. El autor rastrea los orígenes de los *hackers* a un club de ferromodelismo dentro del *Massachussets Institute of Technology* (MIT) en los años cincuenta. En dicho club había estudiantes expertos en crear circuitería para que los trenes a escala funcionaran. Las soluciones ingeniosas a problemas de este tipo se llamaban *hacks* dentro del club, y los miembros con más *hacks* eran llamados *hackers*.

Algunos *hackers* migraron al departamento de inteligencia artificial de la misma universidad, portando el término consigo. Esta pequeña comunidad comenzó a diferenciarse de otras debido a actitudes y valores particulares. Levy describe la ética compartida de la comunidad que se formó en ese primer momento y la enunció en seis puntos:

1. Toda la información debe ser de libre acceso.

2. El acceso a los ordenadores (y a todo aquello que nos pueda enseñar algo acerca de cómo funciona el mundo) debe ser ilimitado y total.

3. Desconfía de la autoridad. Promueve la descentralización.

4. Un *hacker* debe ser valorado por sus "hacks". Es decir, por la calidad de sus diseños y programas, no por criterios falsos y postizos como las titulaciones académicas, la raza o la posición social.

5. Un *hacker* puede crear arte y belleza con un ordenador.

6. Los ordenadores pueden mejorar nuestras vidas. (Levy, 1984; Contreras, 2004, pág. 33)

A partir de este planteamiento, Levy busca describir y entender a varios *hackers* específicos en sus situaciones particulares. Aclara que no busca los nombres más famosos; más bien a los *hackers* promedio. Fue así como personajes como Steve Jobs, fundador de Apple Inc. y

Bill Gates, fundador de Microsoft, ni siquiera aparecen mencionados. Steve Wozniak, desarrollador de la computadora personal, desempeña un papel muy pequeño en la narrativa del libro. Termina con un epílogo titulado "El último de los verdaderos *hackers*", en referencia a Richard Stallman, principal impulsor del software libre. Expondré este tipo de software más adelante; como mencioné, no tiene propietario y cualquier usuario puede modificarlo. Los pioneros del MIT se quejaban de los jóvenes adolescentes que cometían crímenes informáticos y se autodenominaban *hackers*.

La publicación de *Hackers* es muy cercana a dos eventos significativos. El primero es la aparición del primer número de la revista *2600: The Hacker Quarterly*. Se trata de una importante publicación de *hackers* para *hackers* que continúa en circulación hasta el día de hoy. El segundo evento es el estreno de la película *Wargames* (1983). En ella, un adolescente obtiene acceso no autorizado a un sistema militar de Estados Unidos. El ejército se sorprende al ver que sus sistemas más importantes se encuentran a merced de un intruso de 16 años. Para muchas personas, la película representó una introducción al *hacking*.

Si bien el tema no es nuevo, este accidentado panorama inicial introdujo a los *hackers* como una cultura diferenciada. Steven Levy aclara que no escribió una historia formal sobre los orígenes de esta cultura. Más bien esbozó varias viñetas con la esperanza de esclarecer las motivaciones y filosofía de los *hackers*. Si analizamos el libro desde las ciencias sociales, hay deficiencias y sesgos que saltan a la vista. Pero a lo largo de estos años, la obra detonó el interés que guía investigaciones como la que realicé.

Ya para 1989, la guerra fría llegaba a su fin. En este año Gordon Meyer publicó una tesis de criminología en la *Northern Illinois University* titulada *The Social Organization of the Computer Underground*. La criminología enfatiza el análisis y comprensión de las actividades criminales y transgresoras. Los *hackers* fueron categorizados como miembros de "grupos desviados" (*deviant groups*). Con el método etnográfico, Meyer analizó *Bulletin Board Systems* (BBS)[1]; pero también buzones de voz, correos electrónicos y conversaciones

[1] Un BBS es un sistema de mensajería electrónico similar a las redes sociales actuales. Los usuarios entraban a un servidor mediante un nombre de usuario y dejaban mensajes en tablones de anuncios temáticos. Otros

telefónicas con sus informantes.

El autor definió al *underground* informático como el espacio de interacción de tres actores principales. Cada uno de ellos con al menos una de las siguientes actividades:

- *Hacking*: Se refiere a los actos y métodos utilizados para obtener acceso no autorizado a un sistema computacional.

- *Phreaking*: Abarca las técnicas para evitar los mecanismos de cobro de las compañías telefónicas y utilizar los servicios sin costo. En otras palabras, permite llamar gratis y explorar el sistema telefónico sin autorización.

- Piratería: Consiste en copiar y distribuir, de manera no autorizada, software con *copyright*.

Para Meyer existe una división e incomprensión entre estos individuos, a pesar de ellos interactúan dentro del *underground* informático. Esta tesis afirma que su manera de trabajo se da mediante "redes" de colaboración. Ellos coexisten, comparten recursos e información pero funcionan por separado. Mientras que las actividades de *hacking* y *phreaking* se realizan individualmente, los participantes del *underground* informático:

> [Se] asocian unos con otros para discutir asuntos de interés común, como la práctica de técnicas, noticias, y resolución de problemas. Para facilitar este intercambio de información, han establecido una sofisticada red tecnológica [...]. La misma estructura que permite la asociación mutua entre los participantes del *underground* informático también invita a algunos a formar relaciones de trabajo, de tal manera que interactúan como iguales en la participación de las actividades del *undeground* (G. R. Meyer, 1989, págs. 65-66).[2]

Esta tesis describe el funcionamiento social del *underground*; revela que tales organizaciones

usuarios tenían la posibilidad de responder en público o privado al anunciante original.

[2] "Computer underground participants do associate with one another in order to discuss matters of common interest, such as performance techniques, news, and problem solving. To facilitate this informational exchange, they have established a technologically sophisticated network [...]. The very structure that permits mutual association among CU [computer underground] participants also encourages some to form working relation- ships, thus acting as peers by mutually participating in CU activities." (Nota: las traducciones que aparecen a lo largo de la tesis son mías).

son más sofisticadas de lo que antes se pensaba. Meyer encuentra que las prácticas *hacker* no siempre son criminales. Además, las agencias de seguridad exageraban las amenazas y peligros por incomprender el tema.

En 1989 se publicó una novela, basada en hechos reales, titulada *A Cuckoo's Egg*. La obra, escrita por Cliff Stoll en primera persona, narra su experiencia al rastrear un *hacker* que entró a los sistemas de la *University of California, Berkeley*. Después de meses de investigación y rastreo, detectó una compleja red de conexiones que conducían hasta una ciudad alemana. Ahí, un *hacker* al servicio de la KGB,[3] utilizó el sistema de Berkeley como intermediario para acceder a sistemas militares de Estados Unidos. De nueva cuenta, en el libro se asoció el *hacking* con el crimen y el espionaje. La novedad de estos incidentes, la guerra fría y el desconocimiento de la tecnología por el público en general, los señaló a los *hackers* como individuos temibles (Taylor, 1999; Thomas, 2002).

Un artículo de 1990 continúa la búsqueda de conocimiento sobre el tema: "The Baudy World of the Byte Bandit: A Postmodernist Interpretation of the Computer Underground" de Gordon Meyer y Jim Thomas. En él se busca un análisis más cultural que criminológico y se utiliza también el método etnográfico. El texto distanció al *hacking* del crimen. Enfatizó que la mayor parte de los *hackers* "evita destruir información o dañar el sistema deliberadamente". La destrucción de datos sería contraproducente para los objetivos del intruso: Alertaría al administrador del sistema quien cerraría el agujero por donde el *hacker* entró. Las posibilidades de explorar el sistema después quedarían frustradas.

La insistencia en esta distinción se debe a que durante los años ochenta el significado de la palabra *hacker* cambiaba constantemente. Se utilizaba cada vez menos como sinónimo de "programador experto" (Levy) y cada vez más como "criminal informático". Las agencias de seguridad de Estados Unidos y Europa tenían en la mira a los *hackers* como sospechosos delincuentes en potencia.

En 1992 se publicó el libro *Approaching Zero* (1992) de Paul Mungo. Este periodista

[3] La KGB era la agencia de seguridad nacional de la unión soviética.

inglés también retrató al *underground* informático. Dentro de su libro, cita el artículo "The secrets of the little blue box", publicado en la revista Esquire, como uno de los detonantes del *phreaking*. Este artículo de 1971 habla sobre un personaje pintoresco de apodo Captain Crunch, reconocido por llamar gratuitamente en teléfonos públicos gracias a un silbato que salió como regalo en un cereal del mismo nombre.

Otros capítulos de *Approaching Zero* abarcan periodos cubiertos también por otros autores. Conocemos, gracias al libro, más sobre el *Homebrew Computer Club*, un club de aficionados a las computadoras surgido en los años sesenta. También sobre el surgimiento del Apple Computer Inc. Nos explica cómo el término *hacker* cambió de significado de programador a criminal. Asimismo documenta los primeros procesos legales en contra de intrusos informáticos en Inglaterra. También indaga sobre las motivaciones de los *hackers* y los presenta, primordialmente, como personas curiosas. Como individuos apasionados por la tecnología que buscan la exploración (a veces ilegal) y el reconocimiento de sus iguales. Por último, el libro dedica una extensa sección a los virus informáticos.

Como ya mencioné, debido a la imagen criminal que comenzaba a popularizarse en los ochenta, el gobierno de Estados Unidos calificó a los *hackers* como grupo peligroso. Por ello, a finales de la década organizaron redadas donde se confiscó equipo; varios *hackers* terminaron arrestados. Un recuento parcial de estos sucesos se encuentra en *The Hacker Crackdown* (1992) de Bruce Sterling. Se trata de un recuento periodístico, aunque importante para comprender la representación de los *hackers* en ese momento específico. En la narrativa se detallan, entre otras cosas, los conflictos entre agencias de seguridad y grupos de *hackers* como *Legion of Doom* a finales de los ochenta. Especialmente relevante me parecen los recuentos de la llamada *Operation Sundevil,* una serie de operativos que coordinaron las agencias de seguridad de Estados Unidos:

> *Sundevil* parece haber sido una total sorpresa táctica [...] ejecutada con precisión y totalmente arrolladora. Al menos cuarenta "computadoras" fueron confiscadas durante el 7, 8 y 9 de mayo de 1990 en Cincinnati, Detroit, Los Ángeles, Miami,

> Newark, Phoenix, Tucson, Richmond, San Diego, San José, Pittsburgh y San Francisco. Varias ciudades tuvieron múltiples redadas, como las cinco [ocurridas] en los alrededores de Nueva York. En Plano, Texas [...] hubo cuatro confiscaciones de computadoras. Chicago, siempre a la delantera, tuvo su propia redada *Sundevil*, llevada a cabo por los agentes del servicio secreto Timothy Foley y Barbara Golden (Sterling, 1992, págs. 150-151).[4]

Sterling describe el origen del término "ciberespacio" para referirse a las redes de comunicación actuales. El término lo utilizó por primera vez por el novelista William Gibson en *Neuromancer* (1984). En el contexto de la novela, se refiere a un mundo virtual al cual los seres humanos acceden a través de una interfase física que se comunica directamente con el cerebro. Fue John Perry Barlow, poeta y ensayista involucrado en el *underground* informático, quien utilizó el término para describir la Internet. Afirmó que las redes digitales se "habían convertido en un *lugar*, el ciberespacio, que requería una nueva serie de metáforas, una nueva serie de reglas y comportamientos" (Sterling, 1992, pág. 236).[5]

Los autores de este periodo buscaban una visión al interior de la cultura *hacker*. Esto en respuesta a la avalancha de información que provenía del periodismo, el cine y la literatura: medios que no se encontraban al tanto de la vida cotidiana de los *hackers* y sus verdaderas motivaciones. Los académicos han simpatizado tanto con el movimiento *hacker* que en ocasiones parecen adoptar una postura de defensa. Debe comprenderse esta actitud dentro de su contexto. Recordemos que los discursos al interior de los grupos *hacker* eran en ese entonces desconocidos, un misterio casi total.

El artículo "Computer Hackers: Rebels with a Cause" (1994) de Tanja S. Rosteck posee

---

[4] "Sundevil seems to have been a complete tactical surprise [...] precisely timed and utterly overwhelming. At least forty 'computers' were seized during May 7, 8 and 9, 1990, in Cincinnati, Detroit, Los Angeles, Miami, Newark, Phoenix, Tucson, Richmond, San Diego, San Jose, Pittsburgh and San Francisco. Some cities saw multiple raids, such as the five separate raids in the New York City environs. Plano, Texas [...] saw four computer seizures. Chicago, ever in the forefront, saw its own local Sundevil raid, briskly carried out by Secret Service agents Timothy Foley and Barbara Golden."

[5] "It had become a place, cyberspace, which demanded a new set of metaphors, a new set of rules and behaviors."

esta misma actitud: "Ahora los *hackers* son procesados legalmente en una escala desproporcionada a comparación a la amenaza que representan"[6] (Rosteck, 1994). El autor nota que los pocos estudios que existían en ese entonces abordaron el tema desde dos perspectivas: la criminológica y la de las libertades civiles. Esta última porque en las redadas contra *hackers* se violaron muchos de sus derechos constitucionales, como se detalla en el libro de Sterling, *The Hacker Crackdown.*

Rosteck propone comprender al *hacker* como un colectivo revolucionario bajo la perspectiva de Stewart, Smith y Denton (1984) sobre los movimientos sociales. Para demostrarlo, Rostek utiliza el método etnográfico. Sus datos provienen de *e-zines* publicadas por *hackers*, disponibles libremente en redes digitales. El autor tomó este camino debido a la dificultad metodológica que presentaba el obtener acceso a BBS y foros *hacker.* Normalmente su acceso estaba restringido y utilizaban algún cuestionario o prueba para comprobar las habilidades técnicas y la valía de los nuevos miembros.

En retrospectiva y a la luz de trabajos posteriores, "Computer Hackers, Rebels with a Cause" posee varios puntos débiles. Para empezar, la definición de movimiento social que utiliza es muy simplista y generaliza demasiado en sus conclusiones. Por otra parte, revisar únicamente las publicaciones de los *hackers* implica un sesgo en la investigación: son discursos de *hackers* hacia el exterior. Aunque las *e-zines* se dirijan también hacia otros miembros del *underground* informático, son discursos basados en el "deber ser" y no en la práctica cotidiana de los *hackers*. Por otra parte, excluye la posibilidad de contrastar práctica con discurso. Esto lo afirmo por mi experiencia personal como editor de una *e-zine*, pero también por conocer *hackers* contribuyentes de otras.

Hay que comprender las carencias del artículo de Rostek en contexto: la falta estudios similares convierte a este trabajo en una de las primeras afirmaciones de que los *hackers* conforman una colectividad organizada. No son solamente individuos aislados trabajando desde el sótano de su casa en sus computadoras. Rosteck los presenta como "una cultura revolucio-

[6] "Hackers are now being legally persecuted on a scale disproportional to the actual threat they pose."

naria" que los medios han estigmatizado. Los matices vendrán en estudios posteriores.

Dorothy Denning amplió el panorama sobre los *hackers* al estudiar más allá del discurso. Su trabajo "Concerning Hackers Who Break into Computer Systems" (Denning, 1996) es un buen ejemplo de ello. La autora relata que en sus aproximaciones iniciales buscaba desmotivar a los *hackers* en sus actividades criminales. Para su sorpresa, les interesaba que la seguridad informática y la privacidad de la información aumentara: "Mis descubrimientos iniciales sugieren que los *hackers* son aprendices y exploradores que prefieren ayudar antes de causar daño, y que tienen altos estándares de comportamiento" (Denning, 1996, pág. 137). Al delimitar su estudio, tanto Denning como Rosteck dejan fuera aquellos *hackers* involucrados en actividades ilegales (Sterling, 1992).

Durante los años ochenta y noventa, como puede verse, la palabra *hacker* se asociaba de inmediato al crimen (Thomas, 2002). Las publicaciones académicas de este periodo, por su parte, minimizaron el aspecto criminal y lo dejaron fuera de sus objetos de estudio. Es posible que sus autores intentaran cambiar la percepción pública sobre los *hackers* y contrarrestar los efectos de los medios que los representaban negativamente.

Hackers (1999) de Paul A. Taylor es otro importante libro sobre *hackers* dsde un punto de vista cultural. El autor condujo entrevistas 1992 y añadió capítulos posteriormente. Taylor afirma que existe una amplia brecha cultural entre los *hackers* y sus supuestos adversarios. Éstos últimos caracterizados principalmente por las agencias de seguridad nacional y la industria del software. En uno de los capítulos establece que el *hack* es la base de esta cultura. Un *hack* tiene tres características principales, según el autor:

1. Simpleza: El acto debe ser simple pero impresionante.
2. Maestría: El acto involucra conocimiento tecnológico sofisticado.
3. Ilicitud: El acto se encuentra "fuera de las reglas" (Taylor, 1999, pág. 15).[7]

[7] "1. Simplicity: the act has to be simple but impressive.
2. Mastery: the act involves sophisticated technical knowledge.
3. Illicitness: the act is 'against the rules'."

La ilicitud puede comprenderse como "fuera de la ley", pero también como "fuera de las normas de ingeniería". Taylor encuentra normas éticas bastante claras entre los *hackers*. Indaga en el discurso contra ellos y encuentra, a grandes rasgos, dos bandos. A los primeros les llamó "halcones", en referencia a los que buscan sanciones severas para los intrusos. A los segundos los llamó "palomas", los que buscan dialogar con los *hackers* y aprender de ellos para mejorar la seguridad informática. Taylor aporta de manera enorme a la comprensión del tema.

El libro filosófico de Pekka Himanen, *La ética del hacker y el espíritu de la era de la información* (2001), también marca un parteaguas. El autor recupera la obra *La ética protestante y el espíritu del capitalismo*, donde Max Weber afirma que uno de los valores del protestantismo es el sentimiento personal de que el trabajo es un deber.

Weber argumentó que los valores de esta religión favorecieron el desarrollo del capitalismo (M. Weber, 2008). Mientras que en otras, como la católica, el trabajo es una carga y el cielo es el descanso eterno, en el protestantismo el cielo es la recompensa por un trabajo bien hecho (Himanen, 2001). De esta forma, el trabajo duro en la tierra es el medio para alcanzar la salvación eterna.

Para el *hacker* esto es diferente. Su ética de trabajo es más parecida a las éticas preprotestantes y esto repercute en su uso del tiempo y en sus motivaciones económicas. Para afirmarlo, Himanen observa a los *hackers* más visibles y representativos de la ética del *hacker* original propuesta por Levy. Por ello, Himanen ignora por completo a aquellos intrusos informáticos, *phreakers*, piratas de software y prácticamente todo el *underground* informático.

Desde mi punto de vista, Himanen habla de *hackers* de "sombrero blanco", aunque nunca utiliza el término en su obra. Nos presenta los grandes nombres del movimiento: Eric Raymond, Linus Torvalds, Steve Wozniak y Richard Stallman: todos ellos programadores convertidos en figuras públicas sin antecedentes penales. Este uso de la palabra *hacker*, remitiéndose de nueva cuenta a su significado original, ha obligado a investigaciones posteriores a marcar claramente la diferencia entre el intruso y el programador.

¿Qué distingue a un *hacker* de sombrero blanco? Linus Torvalds, como mencioné, desa-

rrolló el núcleo base para el sistema operativo GNU/Linux. A diferencia de empresas que contratan a cientos de programadores bien pagados para que trabajen en ello, Linus lo desarrolló en su tiempo libre. Cada cierto tiempo mostraba su trabajo en la Internet para aceptar sugerencias de cualquier persona interesada en mejorar el sistema. Después de sus ocho horas laborales, trabajaba en Linux sin recompensa monetaria alguna:

> Los *hackers* optimizan el tiempo a fin de que haya más espacio para el ocio: el modo de pensar de Torvalds es que, en medio del trabajo duro y serio que supuso el desarrollo de Linux, siempre debe haber tiempo para ir a la piscina o llevar a cabo algunos experimentos de programación que no responden a metas inmediatas. La misma actitud ha sido compartida por los *hackers* desde el MIT de los años sesenta. En la versión *hacker* del tiempo flexible, las diferentes áreas de la vida, como el trabajo, la familia, los amigos, las aficiones y demás, se combinan con mucha menor rigidez, de modo que el trabajo no siempre se halla en el centro del mapa (Himanen, 2001, pág. 51).

En cuanto a la motivación económica, resulta sorprendente el modelo de desarrollo de estos proyectos de software. En un modelo corporativo tradicional un proyecto pueden costar millones de dólares, pero en el modelo *hacker* proyectos similares los emprenden aficionados sin sueldo. Himanen, al respecto, afirma:

> En realidad, no es posible comprender por qué algunos *hackers* dedican su tiempo libre a desarrollar programas que acaban distribuyendo gratuitamente a los demás, sin percibir los fuertes motivos sociales que tienen para hacerlo. [Eric] Raymond afirma en este sentido que estos *hackers* se hallan motivados por la fuerza que tiene para ellos el *reconocimiento de sus iguales* [...]. La diferencia que les distingue decisivamente de la ética protestante es que para los *hackers* reviste especial importancia el hecho de que el reconocimiento de sus iguales no es un sustituto de la pasión, si no que debe producirse como resultado de la acción

> apasionada, de la creación de algo que sea desde un punto de vista social valioso para esta comunidad creativa (Himanen, 2001, pág. 71).

La visión de Himanen resultó tan reveladora que reivindicó la imagen de los *hackers* a partir de la publicación de su obra. Aún así, es importante repetir que esta imagen la desarrolló analizando exclusivamente casos de programadores de software libre. Himanen presenta los grandes ideales del movimiento, pero quizá encontraría conflictos con la práctica si realizara antropología. Esto debido a que muchos *hackers* se apegan en menor medida a ellos.

Otro libro de igual de importancia se publicó poco tiempo después. Por las referencias bibliográficas y fechas mencionadas dentro del texto, se nota que fue escrito antes de que se publicara de *La ética del hacker*. Por tanto, no incorpora todavía las ideas de Himanen. Se trata de *Hacker Culture* (2002) de Douglas Thomas, una colección de ensayos desde los estudios culturales. Los tópicos que aborda Thomas se dividen en tres:

- La evolución del *hacker*: Thomas rastrea los orígenes del *hacking* desde los sesenta hasta los noventa. Se aproxima de manera mucho más crítica que Levy, y afirma que este fenómeno debe comprenderse como un derivado de la guerra fría. Durante los sesenta, principalmente, el gobierno de Estados Unidos se interesó en el desarrollo de tecnologías de la información. El financiar universidades, particularmente en áreas de ciencia computacional, tenía como objetivo generar herramientas para la guerra. Durante este periodo llegaron las primeras computadoras a las universidades y, gracias a esto, surgió la cultura *hacker*.

- La representación del *hacker*: Thomas revisa las representaciones mediáticas y como los *hackers* se presentan ante el público en varios ejemplos. Thomas revisa minuciosamente los elementos simbólicos de la película *Hackers* (1995).

- La ley del *hacking*: Analiza cómo se construye al *hacker* jurídicamente, especialmente a través del controvertido caso de Kevin Mitnick. Él fue capturado durante los noventa y su caso es tan notorio que se discutirá en extensión más adelante. Su juicio sentó precedentes y se caracteriza por irregularidades en su contra durante el proceso criminal. Este evento es sumamente revelador para comprender la representación social del *hacker*.

En 2002, Jorge Lizama publicó en México el artículo "Hackers: de piratas a defensores del software libre". El artículo ya retoma la perspectiva del *hacker* propuesta por Himanen y en una de sus secciones contextualiza el tema en el caso mexicano. Especula acerca de la apropiación tecnológica que hacen los *hackers* y cómo podría beneficiar al país. Critica especialmente la iniciativa e-México, donde el gobierno de Vicente Fox pretendía instalar un gran número de computadoras con el sistema operativo Microsoft Windows en diferentes comunidades. Si el sistema operativo fuera software libre, como Linux, el costo se reduciría drásticamente:

> Sabemos que México en materia de inversiones en tecnologías de la información ocupa un lugar poco destacado en el contexto de América Latina (debajo de Brasil, Chile y Argentina) y muy rezagado en relación con el contexto internacional. Hoy en día, con la llegada del programa e-México existe la posibilidad de cambiar al menos una parte del panorama tecnológico del país. Sin embargo, desde mi punto de vista para que este escenario no rinda frutos sólo en el apartado de las estadísticas y los logros sexenales si no también en rubro de lo social, es necesario recurrir a algunos de los puntos defendidos por los *hackers*, a los que entiendo como impulsores de proyectos de colaboración y beneficios colectivos al interior de la Internet y no como piratas de las redes (Lizama Mendoza, 2002, pág. 105).

Lizama retomaría el tema en su tesis doctoral "Hackers en el contexto de la sociedad de la información"(2005). Es la única tesis que he encontrado sobre el tema en Latinoamérica. El enfoque parte de la construcción social de la tecnología, una perspectiva crítica que afirma que la tecnología no evoluciona linealmente. La tecnología se crea a partir del "cierre" que se le da a problemas que afectan a diferentes "grupos relevantes" afectados por un nuevo artefacto tecnológico (Bijker y cols., 1987). Desde este punto de vista, se deja atrás la noción de que cada nuevo artefacto es un avance automático con respecto al anterior, y que todos los inventos o innovaciones que no funcionaron simplemente no eran óptimos. Existen razones sociales que llevaron ese artefacto al fracaso.

La práctica *hacker*, para Lizama, se caracteriza por una "flexibilidad interpretativa" que

cuestiona los artefactos tecnológicos y logra modificarlos. La combinación de este hallazgo junto con la "ética *hacker*" ayuda a análizar los *hackers*. Lizama se limitó a los desarrolladores de software libre y a los *hacktivistas,* activistas de la Internet que utilizan prácticas *hacker* como medio de protesta. El autor afirma: "Los *hackers*, mediante la producción de software libre y la defensa de una Internet donde impere el libre acceso y distribución de la información, han generado un modelo de uso social de la tecnología [...] que impulsa directamente el desarrollo de la cultura de la información" (Lizama Mendoza, 2005, págs. 194-195).

Un libro filosófico tomó una postura sumamente radical sobre el tema. *A Hacker Manifesto* (2004) de McKenzie Wark aborda el fenómeno desde una perspectiva marxista. Para Wark el conflicto de los *hackers* es una lucha de clases y su disputa se encuentra en la esfera de la propiedad intelectual. El mundo actual genera abstracciones (información, objetos inmateriales) y los propietarios de éstas son equiparables a los burgueses que describía Karl Marx. La clase *hacker* posee las herramientas para crear una nueva revolución y liberar la información. El libro de Wark se desarrolla totalmente en el mundo de las ideas. No menciona el nombre de ningún miembro de la comunidad y tampoco proporciona evidencia empírica. El autor reinventó el término *hacker*, basándose en el texto de Himanen y otros de Marx. La obra de Wark es un ejercicio de filosofía pura.

Muy diferente es *Me llamo Kohfam* (2004) de Pau Contreras. Se trata de un estudio antropológico sobre la identidad de *hackers* de televisión digital en foros españoles. En estos foros se discute cómo "piratear" tarjetas electrónicas para ver gratis la televisión de paga. Los individuos involucrados publicaban sus técnicas en los foros sin fines de lucro. La pregunta de investigación de Contreras se enfocó hacia la motivación personal de los *hackers* para hacerlo. Utilizó el método ciberetnográfico y creó tres identidades virtuales con las que participó en los foros, obtuvo ayuda e interactuó con diversos *hackers*. El principal informante y centro del estudio tiene por seudónimo Kohfam. A través de entrevistas con él, Contreras obtuvo los elementos suficientes para llegar a conclusiones sobre la identidad *hacker* en ese contexto:

Para la identidad física a la que Kohfam pertenece, las prácticas *hacker* son mucho

> más que una simple actividad lúdica realizada de manera esporádica. Kohfam ha sido uno de los maestros venerables de la comunidad, un *hacker* respetado, capaz de diseñar ingenios técnicos que han mantenido en jaque a grandes empresas de TV digital y que siempre ha compartido sus hallazgos con el resto del grupo. [...] La identidad de Kohfam tiene una dimensión social, construida en el juego de relaciones necesarias para determinar su nivel de prestigio y su estatus dentro del grupo (2004, pág. 139).

Contreras detecta también la existencia de una comunidad. La que denomina "inteligencia-red" y la describe como una organización que permite compartir conocimiento y formar redes de colaboración. Una comunidad que se nutre de conocimiento y genera también productos tecnológicos.

Para terminar este recuento quisiera llamar la atención en torno al trabajo de Gabriella Coleman, una antropóloga radicada en Nueva York. Ella ha tomado el tema de los *hackers* y le ha inyectado continuidad durante la mayor parte de la década pasada hasta el día de hoy. Su trabajo se enfoca principalmente a la ética *hacker* a través de la programación y el *hacktivismo*.

En "Hacker Practice" (2008), Gabriella Coleman y Alex Golub encuentran elementos liberales en la práctica *hacker*. En *Code is speech* (2009), Coleman compara el código fuente, es decir, las instrucciones para la computadora, con un texto escrito en inglés o español. De esta forma, el código podría estar clasificado dentro de las mismas categorías que los libros, revistas, y periódicos. Esto implica que debe ser protegido por la libertad de expresión, como muchas otras manifestaciones de comunicación.

Importante resulta también su texto *The hacker conference* (2010) donde la investigadora se acerca a las conferencias *hacker* etnográficamente, buscando interpretar los significados y la interacción cara a cara, contrastada con la actividad en línea. Su primer libro *Coding Freedom: The Ethics and Aestetics of Hacking* (2012) está a punto de publicarse en la editorial Princeton University Press. El trabajo de Gabriella Coleman es el más reciente y consistente

de los autores revisados en este apartado.

Este panorama general no sólo ayuda a recopilar lo que la ciencias sociales han averiguado acerca de los *hackers*. También permite encontrar descripciones detalladas y diferentes intentos por aproximarse al tema. Los investigadores se acercan cada vez más y encuentran más facetas del fenómeno. Espero que este apartado también haya servido para introducir al lector sobre lo que es un *hacker*, o las diferentes maneras de serlo. Quizá en esta revisión de trabajos previos sea notable una distinción. Por una parte se encuentran los intrusos informáticos y por la otra, los programadores de software libre. Sin embargo, para dejar mucho más clara dicha distinción, en las siguientes secciones intentaré especificar aún más la diferencia.

Kevin Mitnick y los hackers *de sombrero negro*

Cuando cursaba la secundaria no me perdía la sección de informática que se publicaba los lunes en el periódico local. Siempre contenía notas sobre los últimos productos de Microsoft, sobre desarrollo en el hardware y rumores sobre nuevas tecnologías. A veces el periódico publicaba noticias sobre el *hacking*. Nunca olvidaré cuando leí un artículo de título "El chacal de la red". Trataba sobre un *hacker* recientemente capturado y que permaneció como fugitivo varios años: Kevin Mitnick.

La historia era sensacional. Según el artículo, Mitnick había sido responsable de crímenes informáticos y tuvo problemas con la ley en varias ocasiones. Entre sus crímenes se encontraba el robo de software a empresas de telefonía celular. El 24 de diciembre de 1994 accedió sin autorización a la computadora de un experto de seguridad de San Diego, California. Tsutomu Shimomura, el afectado, tomó la intrusión como una "cuestión de honor". Mediante su conocimiento y experiencia en informática, Shimomura auxilió al FBI a capturar al *hacker*, arresto que ocurrió en 1995.

Parecía una película. La trama cautivó mi imaginación y, como *hacker* principiante, moría de ganas por saber cómo Mitnick había logrado todas sus "hazañas". No fui el

único interesado en la historia. El caso ganó notoriedad al grado que ahora es paradigmático. En el periodismo frecuentemente se menciona a Kevin Mitnick como el *hacker* más famoso del mundo y su nombre casi siempre aprece cuando se habla de historias similares. Incluso Manuel Castells se dejó seducir por la historia:

> En enero de 1995, Tsutomu Shimomura, un experto en seguridad informática del San Diego Supercomputer Center, reveló que habían accedido a sus archivos de seguridad y los habían trasvasado a los ordenadores de la Universidad de Rochester, y que otros archivos protegidos de varios emplazamientos se habían visto sometidos a ataques similares, proporcionando pruebas contundentes acerca de que las medidas de seguridad de la red de [la] Internet eran inefectivas contra la invasión del *software* avanzado. Shimomura se vengó de esta ofensa profesional. Prosiguió trabajando en seguir la pista al pirata y, utilizando estrictamente medios electrónicos, unas pocas semanas después condujo al FBI al complejo residencial donde detuvo a Kevin Mitnick, un legendario bandido de la frontera de la red. No obstante, este suceso tan divulgado subrayó la dificultad de proteger la información en [Internet] (Castells, 2008, pág. 389).

Quisiera tomar su caso para ejemplificar a los *hackers* de sombrero negro, aunque existen

muchos otros similares. Todos los *hackers* entrevistados para esta tesis conocen el caso y tienen una postura al respecto. Es un referente importante para comprender la cultura *hacker* en general. Me parece notable que Castells lo llame un "legendario bandido de la frontera de la red". Como menciona Taylor (1999), los *hackers* han sido comparados con vaqueros e Internet con el viejo oeste. Además, el primer artículo que leí lo denominaba "chacal de la red". Las metáforas abundan, pero desde que se divulgó el caso caso hay más evidencia que desmiente la imagen de Mitnick manejada por los medios.

Después de su arresto, Shimomura publicó un libro sobre el incidente en coautoría con el periodista John Markoff. El título es *Takedown: The Pursuit and Capture of Kevin Mitnick, America's Most Wanted Computer Outlaw -by the Man Who Did It* (1996). En él, Shimomura narra los incidentes desde su perspectiva. El libro tiene una extensión de aproximadamente quinientas páginas y Mitnick aparece muy poco. A lo largo de la trama sólo se presentan pocas pistas sobre sus crímenes, pero la mayor parte del libro se centra en la vida de Shimomura. Aún así, se convirtió en un bestseller.

La historia provocó que muchos *hackers* defendieran a Mitnick. Argumentaron que el trato legal y mediático que se le daba a Mitnick era injusto. Estaban en contra de las etiquetas de "chacal", "ciberterrorista", y "Darth Vader" que se utilizaban en las noticias para referirse a él. Por otra parte, los *hackers* alegaban que los medios aprovechaban la historia para recibir beneficios económicos y divulgar información falsa o sin pruebas. El contrato de Shimomura y Markoff para escribir el libro, por ejemplo, fue de $750,000 dólares (Goldstein, 2008; Littman, 1996; Thomas, 2002). Otro de lo puntos del debate era que se acusaba a Kevin Mitnick de robar software, cuando en realidad no había privado a las empresas de su posesión, más bien lo copió. Estos debates se encontraban sumamente polarizados y, al final del *Takedown*, Shimomura añade una nota:

> Hoy, mientras recapitulo sobre los eventos de los últimos dos años, todavía estoy angustiado. Justo después de su arresto, la leyenda de Mitnick siguió creciendo. Durante muchos meses posteriores, se discutió acaloradamente en la Internet sobre

> sus acciones. Algunos argumentaban como Kevin Mitnick nunca dañó a nadie físicamente, lo que hizo fue inocuo. El hecho es que es el caso de un hombre que tuvo 16 años y seis arrestos para distinguir entre el bien y el mal. A finales de los ochenta, un juez federal se esforzó por darle una segunda oportunidad. Para mi, el verdadero crimen de Kevin Mitnick fue violar el espíritu original de la ética del *hacker*. No es bueno leer el correo de otras personas. Creer que el software y otras tecnologías de cómputo deben compartirse libremente no es lo mismo que creer que está bien robarlas (Shimomura y Markoff, 1996, pág. 493).[8]

En 1996 también se publicó otro libro sobre Mitnick, con una perspectiva muy diferente. *The Fugitive Game* (1996) de Jonatan Littman. El periodista entrevistó a Mitnick por teléfono mientras se encontraba fugitivo. Su objetivo original era escribir un artículo para la revista *Playboy*, pero al ver que la noticia se hizo tan importante, lo expandió hasta convertirlo en libro. Mitnick comentó que nunca hubiera hablado con él de haber sabido que escribiría un libro completo (Mitnick y Simon, 2002).

Littman acertó al corroborar las fuentes y entrevistar a muchas de las personas involucradas en el caso. Esto reveló que la participación de Shimomura en la captura de Mitnick fue mínima y que basó mucha de la información de su libro en fuentes no confiables. Por otra parte, contiene más datos de la vida cotidiana de Mitnick, obtenidos a través de entrevistas telefónicas. Pero no aportó gran información sobre las técnicas utilizadas por el *hacker* para copiar sin autorización el software de grandes empresas. A pesar de esto, hasta hace poco era el recuento más completo de los eventos que culminaron con su arresto.

Douglas Thomas dedica todo un capítulo de su libro *Hacker Culture* (2002) a Kevin

[8] "Today, as I look back on the events of the past two years, I'm still troubled. In the wake of his arrest in February 1995, Mitnick's legend continued to grow. For many months after his arrest, heated discusions about his actions took place on the Internet. Some people continued to argue that because Kevin Mitnick never physically harmed anyone, what he was doing was innocuous. The fact is that this is also the case of a man who has had sixteen years and six arrests to figure out what is right and wrong. In the late 1980s, a Federal judge made a special effort to give him a second chance. For me, Kevin Mitnick's real crime is that he violated the original spirit of the hacker ethic. It's not okay to read other people's mail, and to believe that software and other computer technologies should be freely shared is not the same as believing that it's okay to steal them".

Figura 3. **:** El movimiento "Free Kevin", organizado por la revista *2600: The Hacker Quartely*, repartía gratuitamente calcomanías como ésta para difundir su causa.

Mitnick. Él también llamó la atención a las exageraciones de los periodistas al reportar la historia y que las fotografías utilizadas para representarlo en los periódicos eran generalmente las menos favorables.

> Lo que ha condenado a Mitnick ante los ojos tanto del público como la ley no es su *hacking*, es su personalidad. La caracterización de Mitnick se construyó casi por completo a partir de testimonios de segunda mano de gente que participó como informante en su contra o tiene algún interés en vilificarlo para servir a sus propósitos. Convertir a Mitnick en el "arquetípico *hacker* 'del lado oscuro"' es una jugada que ha servido a muchas agendas, recientemente la de Shimomura y Markoff (Thomas, 2002, pág. 203).[9]

La revista *2600: The Hacker Quarterly* de Nueva York siguió el caso. Su editor es un *hacker* de seudónimo Emmanuel Goldstein, nombre de un personaje de la novela *1984* de George Orwell. En esta publicación trimestral se encuentran artículos técnicos y otros de corte más político y social. Los artículos normalmente los envían los mismos lectores y los selecciona el editor.

El juicio de Mitnick se posponía y pronto se dio la noticia de que el libro *Takedown*, con todas sus inconsistencias, se convertiría en película. Algunos *hackers* consiguieron ilegalmente

[9] "What has damned Mitnick in the eyes both of the public and law enforcement is not his hacking, but his personality. That characterization of Mitnick is built almost entirely on secondhand accounts from people who had either served as informants against him or had an investment in vilifying him to suit their own agendas. Turning Mitnick into the "archetypal 'dark-side' computer hacker" is a move that has suited a number of agendas, most recently Shimomura's and Markoff's."

un libreto de la película que sería producida por Miramax. Descubrieron que en la trama el juicio de Mitnick ya había ocurrido y había sido encontrado culpable. Además, contenía una escena donde Mitnick agredía físicamente a Shimomura (Thomas, 2002). Emmanuel Goldstein y el grupo editorial de la revista lanzaron el movimiento *Free Kevin* para difundir información sobre el caso. Su objetivo era lograr que se hiciera un juicio justo para Mitnick y que la película no afectara el proceso legal.

Durante este periodo los *hackers* de la revista *2600* registraron sus manifestaciones en vídeo y entrevistaron a personajes relevantes para el caso. Estas grabaciones fueron materia prima para el documental *Freedom Downtime* (Goldstein, 2001), sobre las irregularidades en el proceso legal contra Mitnick. La película es, hasta donde tengo noticia, la primera en donde los *hackers* se documentan a sí mismos. Las manifestaciones tuvieron como efecto que el libreto original se cambiara y se apegara más a los hechos. La película se estrenó en 2000 bajo el nombre de *Track Down*.

Kevin Mitnick salió libre ese mismo año. Los términos de su liberación fueron sumamente estrictos. Exigían, por ejemplo, que no utilizara una computadora durante varios años. Escribió su primer libro *The Art of Deception, Controlling the Human Element of Security* (Mitnick y Simon, 2002) en una máquina de escribir. Es un manual para que empresas pse protejan de ataques como los suyos. Fundó su empresa de consultoría de seguridad informática basándose en esos conocimientos.

Otro de los términos de su libertad condicional exigía que no contara la historia de sus crímenes durante siete años. Esto provocó que muchos de los mitos y rumores sobre su caso continuaran su curso sin que nadie los detuviera. Cuando el plazo se cumplió, Mitnick escribió su autobiografía, publicada en 2010 bajo el nombre de *Ghost in the Wires* (2010).

Esta obra es sumamente reveladora e importante por muchos motivos. En primer lugar, conocemos por su propia voz a uno de los *hackers* de sombrero negro más famosos del mundo. Alguien que fue sujeto de especulación controversial en el pasado. Su historia se toma como ejemplo del poder de los intrusos informáticos y cómo la tecnología es inconfiable

para proteger información importante.

La gran sorpresa es que la mayor parte de sus intrusiones no son tecnológicas. Mitnick cuenta que de niño era aficionado a leer libros que enseñaban cómo crear identidades falsas, abrir candados y manipular gente. Aplicó muchas de esas técnicas a lo largo de su vida. Para conseguir el código fuente de software de grandes empresas no penetraba directamente su sistema informático. La mayor parte del tiempo llamaba a la empresa por teléfono, fingía ser otra persona y pedía por favor que se lo enviaran. La estrategia funcionaba la mayor parte del tiempo.

Consiguió su identidad falsa al visitar el registro civil de una ciudad de Dakota del Sur. Como ese estado tenía registros abiertos al público, buscó actas de nacimiento de personas que nacieron en un año aproximado al suyo. En un momento de descuido de los empleados, una vez que había ganado su confianza, robó hojas membretadas de acta de nacimiento. Cuando la máquina para sellar quedó sin vigilancia, las selló todas para que parecieran auténticas. Después de algunas semanas, agradeció las atenciones de todos y se retiró. Imprimió las actas él mismo, con las identidades que encontró en el registro.

Así fueron la mayor parte de sus intrusiones. Mitnick me recuerda mucho a Frank W. Abagnale, el famoso falsificador, impostor y autor del libro *Catch Me if You Can* (2000), que posteriormente se convirtió en una película de Steven Spielberg. Mitnick, como Abagnale, era un impostor: tiene una habilidad muy grande para mentir, engañar y manipular. A esta forma de obtener información o beneficios de la gente se le conoce en el mundo del *hacking* como "ingeniería social" (Thomas, 2002). Las experiencias narradas en su autobiografía nos demuestran que, a pesar de tener un sistema informático muy seguro, el elemento humano es siempre el más inconfiable.

Decidí hablar de este caso debido a su importancia, pero también por lo que se aprende de él para esta investigación. Si Mitnick es el mejor ejemplo de un *hacker* de sombrero negro, es probable que los ataques que aparecen en las noticias se realicen de manera similar. Quizá no sea a través de la tecnología como los sistemas quedan expuestos a ataques. Durante

Figura 4. **:** Kevin Mitnick (izquierda) presentando su autobiografía en el programa "The Colbert Report". El episodio fue transmitido por el canal Comedy Central el 18 de septiembre de 2011.

años las empresas atacadas se preguntaron cómo hizo Mitnick para romper sus esquemas de seguridad. ¿Qué conocimiento técnico tan avanzado tenía ese hombre para no dejar rastro?

No fueron los sistemas informáticos los más vulnerables: fue el ser humano. La mayor parte de las veces los mismos empleados, por amabilidad y servicio, permitieron a Mitnick entrar sin autorización a un edificio o copiar información confidencial. Con el tiempo, este *hacker* de sombrero negro se volvió muy hábil para ganarse a las personas y obtener lo que quería. Para Mitnick, esto era una "debilidad" de los otros. Un psicólogo podría pensar diferente.

El caso de Mitnick no es aislado. Se pueden contar muchos más, como el de Kevin Poulsen, capturado el 21 de junio de 1991 por crímenes informáticos (Littman, 1997) y el de "Mafiaboy" arrestado en 2000 por participar en un ataque coordinado contra grandes sitios de Internet como Yahoo! (Verton, 2002). Sin embargo, todos los casos siguen líneas similares a las ya descritas por autores que han escrito sobre el tema: jóvenes inquietos e inteligentes

que por curiosidad y malicia invaden sistemas sin autorización (Sterling, 1992; Mungo, 1992; Thomas, 2002). Al escribir sobre todos estos casos, los autores generalmente mencionan el nombre de Kevin Mitnick.

Software libre y los hackers *de sombrero blanco*

En un mundo casi aparte, se encuentran otro tipo de *hackers* que acaparan mucho menos tiempo en las noticias, películas y periodismo. Steven Levy termina su libro *Hackers* con un epílogo sobre Richard Stallman. En 1984 el autor lo promueve como uno de los últimos *hackers* del MIT. Esta etiqueta de no es válida por varios motivos. Uno de ellos es que actualmente existen muchos más *hackers* que se apegan a la ética original del MIT, no sólo Stallman. Por otra parte, ahora hay que aclarar si un *hacker* es de sombrero blanco o sombrero negro. A pesar de esto, Stallman es crucial para comprender las motivaciones de los *hackers* de sombrero blanco.

En los inicios de la computación, el software no tenía la misma importancia que el día de hoy. Las grandes computadoras, o *mainframes*, tenían un costo tan elevado que había muy pocas de ellas en el mundo. El software que se escribía era especializado y gestionado únicamente por algunos expertos en el tema. La posibilidad de que este código fuese robado era sumamente pequeña. La miniaturización, abaratamiento y simplificación de las computadoras logró que el panorama cambiara. Con la computación personal llegó la posibilidad de compartir programas libremente, al copiar cintas perforadas o cintas magnéticas. La naciente industria del software se preocupó por restringir la copia por medios legales o tecnológicos (Levy, 1984).

En 1976, Bill Gates escribió una carta abierta a los aficionados de la computación. En ella, pedía a la comunidad que comprara su software y no lo copiara ilegalmente ya que esto afectaba la industria. La carta se publicó en el boletín del Homebrew Computer Club y causó controversia al momento de su aparición. El argumento era principalmente económico:

> ¿Por qué sucede esto? Como la mayor parte de los aficionados sabe, casi todos

> roban su software. El hardware se debe pagar, pero el software es algo para compartir. ¿A quién le importa si la gente que trabajó en ello recibe compensación? [...] ¿Quién puede permitirse trabajar profesionalmente a cambio de nada? ¿Qué aficionado puede invertir tres años en la programación, encontrar los errores, documentar su producto y distribuirlo gratis?[10] (Gates, 1976, pág. 2).

La carta suscitó varias respuestas en números posteriores, tanto de apoyo como de oposición. Dejó bien en claro la postura de la industria del software acerca de la piratería de software. Las empresas de software concebían su código como secreto comercial y los empleados firmaban acuerdos de confidencialidad al trabajar para ellas.

Durante este periodo, Richard Stallman se unió al departamento de inteligencia artificial del MIT, en 1971 para ser precisos. Le sorprendió la manera en que los usuarios compartían libremente sus propias creaciones de software dentro de la comunidad del laboratorio (Wayner, 2000; Levy, 1984; S. Williams, 2002). Esto sucedía de la misma forma como otras personas compartían recetas de cocina, cambiándolas a su gusto y compartiendo también los cambios.

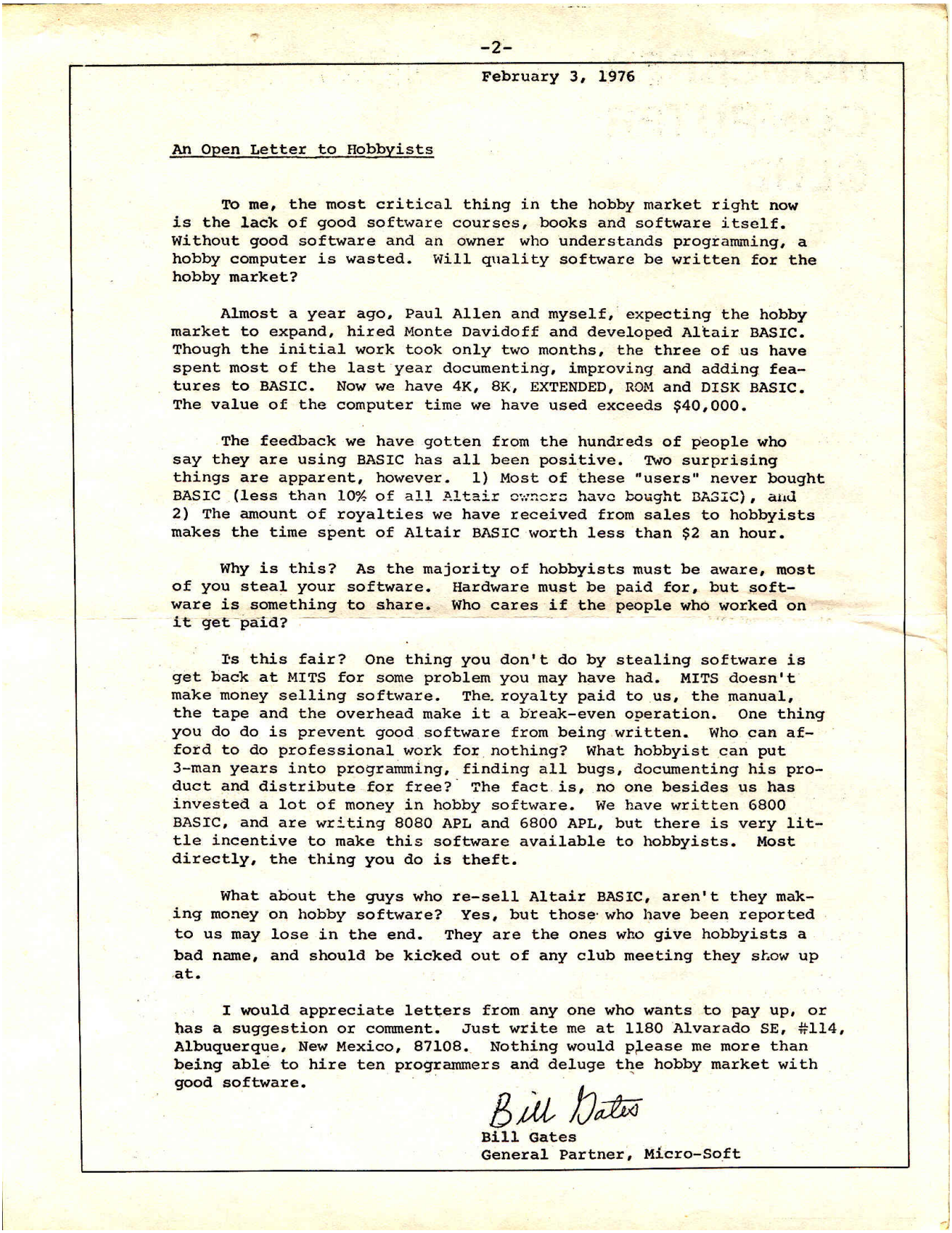
> No llamábamos a nuestro software "software libre" porque el término todavía no existía; pero eso era. Cuando la gente de otra universidad o alguna compañía quería portar y usar nuestro programa, se lo permitíamos con gusto. Si veías a alguien usando un programa poco familiar o interesante, siempre podías pedirle ver el código fuente para leerlo o reutilizar partes de él para hacer un nuevo programa (Stallman, 2009, pág. 7).[11]

Algunas empresas de la incipiente industria del software vieron en esta forma de trabajar una oportunidad para obtener beneficios.

---

[10] "Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid? [...] Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free?"

[11] "We did not call our software 'free software,' because that term did not yet exist; but that is what it was. Whenever people from another university or a company wanted to port and use a program, we gladly let them. If you saw someone using and unfamiliar and interesting program, you could always ask to see the source code, so that you could read it, or cannibalize parts of it to make a new program."



-2-

February 3, 1976

An Open Letter to Hobbyists

To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

Almost a year ago, Paul Allen and myself, expecting the hobby market to expand, hired Monte Davidoff and developed Altair BASIC. Though the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4K, 8K, EXTENDED, ROM and DISK BASIC. The value of the computer time we have used exceeds $40,000.

The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however. 1) Most of these "users" never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent of Altair BASIC worth less than $2 an hour.

Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?

Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software. We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists. Most directly, the thing you do is theft.

What about the guys who re-sell Altair BASIC, aren't they making money on hobby software? Yes, but those who have been reported to us may lose in the end. They are the ones who give hobbyists a bad name, and should be kicked out of any club meeting they show up at.

I would appreciate letters from any one who wants to pay up, or has a suggestion or comment. Just write me at 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.

Bill Gates

Bill Gates
General Partner, Micro-Soft

Figura 5. **:** Imagen de "An Open Letter to Hobbyists" escrita por Bill Gates el 3 de febrero de 1976. Se publicó en el boletín del Homebrew Computer Club.

> Esta filosofía era la razón principal por lo cual compañías como Xerox establecieron políticas de donación de máquinas y software a lugares donde los *hackers* se congregaban. Si los *hackers* mejoraban el software, las compañías podían tomar prestadas las mejoras e incorporarlas en las versiones actualizadas para el mercado comercial. En términos corporativos, los *hackers* eran una ventaja comunitaria, una división auxiliar de investigación y desarrollo disponible por un costo mínimo (S. Williams, 2002, pág. 9).[12]

Para finales de los ochenta la comunidad disminuyó en tamaño y en actividad. La principal causa, según Stallman, fue que la compañía Symbolics contrató a la mayor parte de los *hackers* del laboratorio del MIT (Stallman, 2009; S. Williams, 2002). Otro de los motivos fue que las licencias de nuevas computadoras adquiridas restringían grandemente su uso. No se permitía cambiar nada ni distribuir copias libremente. Para Stallman este tipo de restricciones son "antisociales y antiéticas" (Stallman, 2009) ya que mantienen divididos a los usuarios. Al ver a la comunidad del departamento de inteligencia artificial prácticamente deshecha, Stallman consideró sus opciones. Para él, entrar al ámbito del software comercial era impensable:

> Podría haber ganado dinero de esta forma, y posiblemente me hubiera divertido escribiendo código. Pero sabía que al final de mi carrera miraría atrás, hacia mis años de construir murallas para dividir a la gente, y sentiría que he desperdiciado toda mi vida haciendo del mundo un peor lugar (Stallman, 2009, pág. 9).[13]

Argumenta en contra de este tipo de contratos a través de una experiencia que tuvo con una impresora. Ésta carecía de varias características que fácilmente se podrían añadir a través de software (S. Williams, 2002). El dueño del controlador se rehusó a mostrar el código fuente

[12] "Such a philosophy was a major reason why companies like Xerox made it a policy to donate their machines and software programs to places where hackers typically congregated. If hackers improved the software, companies could borrow back the improvements, incorporating them into update versions for the commercial marketplace. In corporate terms, hackers were a leveragable community asset, an auxiliary research-and-development division available at minimal cost."

[13] "I could have made money this way, and perhaps amused myself writing code. But I knew that at the end of my career, I would look back on years of building walls to divide people, and feel I had spent my life making the world a worse place."

para que los *hackers* del MIT la implementaran, así que tuvieron que tolerar la falta de funciones. "No podía convencerme a mi mismo de que los contratos de confidencialidad son algo inocente" (Stallman, 2009, pág. 9). Para continuar programando y no faltar a su ética, Stallman creó un nuevo sistema operativo con código fuente visible por todos. Se inspiró en un sistema operativo preexistente llamado UNIX[14]. Esto debido a la popularidad del mismo y a que era fácil crear aplicaciones compatibles para él. La compatibilidad lograría que los usuarios migraran al nuevo sistema con menor esfuerzo.

Bautizó al proyecto como GNU, abreviatura de *Gnu's not UNIX*[15]. El desarrollo de un nuevo sistema operativo es una tarea muy compleja, por lo que Stallman la abordó por partes: Reemplazó las aplicaciones más utilizadas por los usuarios, con el fin de integrarlas en un sistema completo y totalmente libre. Esto, por supuesto, requirió tiempo y ayuda de voluntarios que se unieran a la causa. En 1984 renunció a su trabajo en el MIT y se dedicó de lleno al desarrollo de GNU. Stallman considera que su renuncia fue "necesaria", ya que de haber permanecido ahí, la universidad podría haber alegado propiedad del nuevo software, ya que utilizó para su trabajo los laboratorios del MIT (Stallman, 2009).

La cuestión de la libertad es central en el pensamiento de Stallman. Para él, el software libre no es una cuestión de precio. En inglés el término "free software" es ambiguo: Puede significar "software gratis" o "software libre", por lo que en este idioma normalmente se debe aclarar el sentido. Las libertades que ofrece este tipo de software son cuatro:

1. La libertad de ejecutar el programa, para cualquier propósito.

2. La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que

[14]UNIX es un sistema operativo multiusuario y multitarea desarrollado en 1969 por los laboratorios Bell en AT&T. Sus principales creadores fueron Ken Thomson y Dennis Ritchie. A pesar de haber sido creado hace más de 40 años, es un sistema tan sólido y actualizable que se sigue utilizando para una gran gama de tareas y muchos otros sistemas operativos, como Linux, imitan su estructura y funcionalidad. Dentro del mundo *hacker* se le tiene considerado como uno de los mejores sistemas operativos y normalmente se duda muchísimo de las capacidades de un *hacker* que no esté familiarizado con algún sistema UNIX. Como se verá a lo largo de este texto, es tan importante que constantemente aparece en el discurso de los *hackers*.

[15]En español la mejor traducción de esta frase sería: Ñu no es UNIX. Lo que Richard Stallman pretendía era encontrar un acrónimo "recursivo". Es decir, que el acrónimo se encontrara incluído dentro de la definición del mismo. Este tipo de acrónimos recursivos son muy comunes en el mundo *hacker*. Por ejemplo, otros dos paquetes de software con nombres similares son WINE, que significa Wine Is Not an Emulator y PINE, que significa Pine Is Not Elm.

usted quiera (el acceso al código fuente es una condición necesaria para ello).

3. La libertad de redistribuir copias para que pueda ayudar al prójimo.

4. La libertad de distribuir copias de sus versiones modificadas a terceros. Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios (el acceso al código fuente es una condición necesaria para ello). (Stallman, 1996)

En cuanto al precio del software libre: los desarrolladores tienen la libertad de venderlo. Sin embargo, no pueden impedir a otros usuarios que compartan copias. De hecho, Stallman vendía cintas magnéticas con las primeras versiones de programas GNU que desarrollaba junto con los colaboradores que se unieron a su proyecto. Entre las primeras aplicaciones desarrolladas se encontraba el editor de texto Emacs y un compilador para el lenguaje de programación C.

Aún cuando el sistema se encontraba todavía incompleto, las aplicaciones por separado funcionaban dentro de un entorno UNIX. Para Stallman esto sólo era parte del trabajo, ya que UNIX es un sistema comercial, o "privativo", como él lo nombra. Es decir, un sistema que priva a los usuarios de las libertades que ofrece el software libre.

Cabe destacar que, a pesar de ser ateo, en la visión de Stallman abundan las alegorías religiosas. Habla de "evangelizar" acerca del software libre y "convertir" a más usuarios. Se disfraza como un personaje llamado "Saint Ignucious" y hace exorcismos simulados donde elimina el software comercial de una computadora y lo reemplaza por soft-

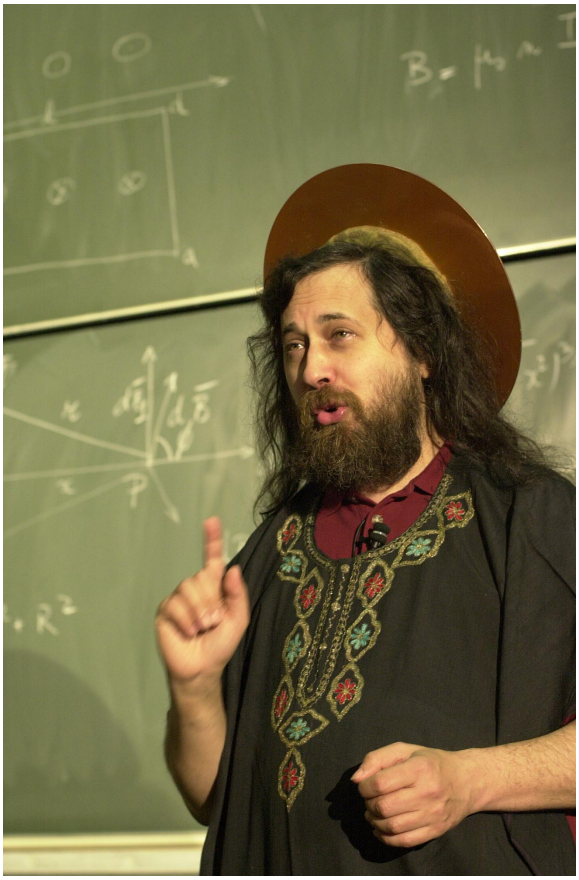


Figura 6. **:** Richard Stallman disfrazado de *Saint Ignucious*.

Figura 7. **:** La laptop Yeelong Lemote es la computadora actual de Richard Stallman y, por el momento, la única totalmente libre. El resto de las computadoras en el mercado poseen por lo menos algunos componentes propietarios.

ware libre (Himanen, 2001). Stallman afirma

que las características atractivas del softwa-

re privativo son "tentaciones" para que los usuarios renuncien a sus libertades. Incluso su computadora actual no posee ningún componente "privativo".

Otra discurso evidente en el discurso de Stallman es el marxista. De cierta manera, la historia que cuenta sobre el laboratorio de inteligencia artificial del MIT remite al comunismo primitivo del que hablaban Marx y Engels (Marx y Engels, 1966). La comunidad de laboratorio compartía libremente el software sin que hubiera ningún propietario. Posteriormente llegó la industria del software, particularmente la empresa Symbolics, e irrumpió la armonía del lugar. Por ello, Stallman busca liberar todo el software de manera sistemática para llegar

a una nueva fase en el mundo de la informática. Por eso tomó la iniciativa de crear GNU.

Una de las partes más importantes de un sistema operativo es el núcleo. Es el componente encargado de administrar tanto el hardware como el software, por lo que su función es vital. Es también una de las partes más complejas y su desarrollo requiere de mucho tiempo. Stallman y su equipo emprendieron el desarrollo de un núcleo libre de nombre Hurd. Antes de concluirlo, conocieron el proyecto Linux, iniciado por un estudiante de Finlandia.

Linus Torvalds siempre destacó en matemáticas y física; las computadoras para Linus eran solo un pasatiempo. Pero un libro lo inspiró a indagar mucho más: "*Operating Systems: Design and Implementation*" de Andrew Tanenbaum. Es un libro obligatorio en muchos cursos de ingeniería. Yo también lo revisé cuando cursé la materia de sistemas operativos en mi propia universidad. En él se publica el código fuente de un sistema operativo sencillo llamado Minix. El sistema era una herramienta que utilizaba Tanenbaum como componente didáctico para introducir a sus estudiantes a los sistemas tipo UNIX. Frustrado por las características faltantes de Minix, Linus decidió programar su propia versión (Torvalds y Diamond, 2001).

Su objetivo nunca fue crear un nuevo sistema operativo, más bien practicar programación y tener un pasatiempo. Linus llamaba juguetonamente "Linux" a su nuevo sistema, una mezcla entre su propio nombre y el de UNIX, sistema al que intentaba emular. Cuando terminó una versión medianamente utilizable, publicó el código en la Internet para que cualquier curioso lo utilizara. Pronto varias personas encontraron errores en el sistema. Algunos de ellos le escribieron para comentárselo y otros más tomaron la iniciativa y enviaron parches de corrección. Linus los revisaba, evaluaba si valían la pena, y en caso afirmativo los integraba al sistema. Publicó las nuevas versiones:

> Fue en enero [de 1992] cuando los usuarios de Linux se incrementaron de cinco, diez, veinte personas -tipos a quienes yo podía enviar un correo electrónico y cuyos nombres conocía- hasta convertirse en cientos de personas anónimas. Yo no conocía a todas las personas que usaban Linux, y eso era divertido (Torvalds y

Diamond, 2001, pág. 92-93).[16]

Cuando Stallman encontró Linux, el sistema ya se encontraba con la suficiente madurez como para convertirse en el núcleo de GNU. Pidió permiso para integrarlos a su propio proyecto y Linus aceptó. Este acuerdo informal ocurrió en 1992 y la unión de ambos proyectos se conoce como GNU/Linux. (Stallman, 2009) A pesar de esto, coloquialmente se le conoce simplemente como Linux, lo cual enfurece a Stallman quien siente que con ese nombre se ignora el origen del sistema.

Una de las preocupaciones de Linus era la propiedad intelectual del sistema. No quería lucrar con él, pero tampoco quería que alguien más lo hiciera, beneficiándose con el trabajo de otros. El *copyright* ofrecía protección, pero Linus no creía correcto registrar el software a nombre de una sola persona o institución cuando. Su argumento fue que no era producto de uno solo, había una comunidad muy grande que colaboraba con el proyecto. Además, Linus creó su sistema con herramientas disponibles libremente en la Internet, como el compilador de C de Richard Stallman.

Para proteger el trabajo de la comunidad, Linus recurrió a la licencia creada por Stallman, la Generic Public Licence (GPL). Esta licencia ofrece los beneficios del software libre y obliga a todos aquellos que reutilicen el programa a ofrecer las mismas libertades. Esto convirtió a GNU/Linux en el primer sistema operativo completamente libre. Es interesante que los motivos de Linus para liberar su software fueron muy diferentes a los de Stallman. Este último buscaba liberar todo el software del mundo; Linus sólo quería retroalimentación por parte de sus usuarios:

> Hablando generalmente, veo los *copyrights* desde dos perspectivas. Digamos que hay una persona que gana $50 dólares al mes. ¿Deberíamos esperar que pague $250 dólares por un software? No creo que sea inmoral de esa persona el copiar el software ilegalmente y gastar su salario de cinco meses en comida. Ese tipo de

[16]"It was in January [1994] that Linux users grew from five, ten, twenty people -folks who I could email and whose names I knew- to hundreds of unidentifiable people. I didn't know everybody using Linux, and that was fun."

Figura 8. **:** Linus Torvalds posando irónicamente en Japón frente a los estantes con Windows 7, cerca del día de lanzamiento.

> violación de *copyright* está moralmente bien. Y es inmoral -por no decir estúpido- el intentar capturar al "violador". Con respecto a Linux, ¿a quién le importa si un individuo no respeta totalmente la GPL si está utilizando el programa para sus propios fines? Es cuando alguien intenta ganar dinero rápido cuando me parece inmoral, aunque suceda en los Estados Unidos o África. E incluso entonces es un asunto de niveles. La avaricia nunca es buena (Torvalds y Diamond, 2001, pág. 97).[17]

[17]"Generally speaking, I view copyrights from two perspectives. Say you have a person who earns $50 a month. Should you expect him or her to pay $250 for software? I don't think it's immoral for that person to illegally copy the software and spend that five months' worth of salary on food. That kind of copyright infringement is morally okay. And it's immoral-not to mention stupid-to go after such a "violator". When it comes to Linux, who cares if an individual doesn't really follow the GPL if they're using the program for their own purposes? It's when somebody goes in for the quick money-that's what I find immoral, whether it happens in the United States or Africa. And even then it's a matter of degree."

Linus alega que existe un vínculo entre el creador y su creación que nunca se borrará. Por ello, no debe imponerse a todo el software la misma licencia como pretende Stallman. Hay casos legítimos en los que un programador no debe compartir su código fuente o permitir que los demás lo copien sin problema. Para Linus, el asunto de la propiedad intelectual tiene que ver con eso mismo que nos hace humanos: la creatividad.

> Ese tipo de creatividad -en forma de pintura, música, escultura, escritura o programación- debe ser sagrada. El creador y su creación tienen un vínculo que no puede romperse. Es como el vínculo entre una madre y su hijo, o entre la mala comida china y el glutamato monosódico. Pero al mismo tiempo es algo en lo cual todo mundo debería participar, por que *es* humano (Torvalds y Diamond, 2001, pág. 205).[18]

En su visión del mundo, Linus entiende el software como si fuera una pieza artística: un producto de la creatividad humana. Aunque admite que no se aplican exactamente los principios del software libre a los objetos de arte. En esto coincide con Stallman, quien argumenta que lo que se aplica para el código fuente no debe aplicarse para la literatura, por ejemplo. Es decir, no es lo mismo que las personas tengan libertad de modificar un software a que cualquiera pueda modificar una novela.

A finales de los noventa, Steve Jobs se interesó en el trabajo de Linus Torvals. Le envió un correo electrónico pidiéndole una o dos horas de su tiempo. El motivo era explicar el trabajo de Apple Inc. en su nuevo sistema operativo basado en UNIX de nombre Mac OS X. Linus accedió y lo escuchó atentamente. Jobs intentó convencerlo de que el futuro residía en los productos de Apple. Según él, sería inteligente que gente de Linux trabajara en ellos para vencer a Microsoft como el sistema operativo más popular.

Pero la idea no agradó a Linus en absoluto. Su pregunta fue: "¿por qué debería intere-

[18]"That kind of creativity -wheter it be in the form of a painting, music, sculpture, writing, or programming- should be sacred. The creator and the thing he or she created have a bond that cannot be severed. It's like the bond between a mother and child, or between bad Chinese food and MSG. But at the time it's something that everybody in the whole world should be able to be part of, because it *is* human."

sarme?" No creyó que hubiese nada atractivo en Apple y tampoco concordaba con el aspecto técnico del nuevo sistema: "[Jobs] asumió que [su propuesta] me interesaría. No tenía idea, no podía concebir que hubiera segmentos enteros de la raza humana totalmente despreocupados por incrementar el mercado de la Mac -o en el tamaño del mercado de Microsoft" (Torvalds y Diamond, 2001, pág. 151).[19]

Otro *hacker* importante como líder intelectual es Eric S. Raymond, también programador. Su postura difiere con la de Stallman al preferir el término código fuente abierto (*open source*) en lugar de software libre (Raymond, 1998). El argumento principal es que si el software libre pretende entrar al mundo comercial, debe adaptarse a las necesidades corporativas. Esto implicaría que el software podría tener propietarios, pero aún así los usuarios tendrían la posibilidad de analizar el funcionamiento interno del programa y sugerir modificaciones.

Raymond explicaría más a detalle su postura frente al software en su libro *The Cathedral and the Bazaar* (2001). Raymond ejemplificaría cómo el código fuente abierto rinde mejores resultados, también es más seguro ya que su fortaleza se encuentra en que todos los usuarios pueden sugerir cam-

Figura 9. **:** Eric S. Raymond.

[19]"He just took it for granted that I would be interested. He was clueless, unable to imagine that there could be entire segments of the human race who weren't the least bit concerned about increasing the Mac's market share. I think he was truly surprised at how little I cared about how big a market the Mac had -or how big a market Microsoft has."

bios para mejorarlo. Es aquí donde Raymond formula la llama "Ley de Linus": "con muchas miradas, todos los errores saltarán a la vista" (Raymond, 2001).

A mediados de los noventa surgieron empresas dispuestas a lucrar con el software libre. Algunas estas ofrecen "distribuciones" de Linux. Es decir, compilan aplicaciones útiles para los usuarios y las distribuyen armadas de antemano. Esto hace que aquellos con menos habilidades técnicas puedan instalar el sistema sin armarlo por si mismos. Las primeras surgieron en 1994. Entre ellas se distinguen Pacific Hi-Tech, Suse, Red Hat y Caldera (S. Weber, 2005, 107-108). Esta última causó controversia al combinar software libre con software privativo con el afán de ganar más usuarios, por lo que fue criticada principalmente por la Free Software Foundation (S. Weber, 2005; Stallman, 2009).

Ya para finales de los noventa, Microsoft se enfrentaba a un jucio antimonopolio por competencia desleal contra Apple y Netscape. Una de las mayores pruebas en su contra eran los denominados "Documentos Halloween", que consistían en cartas y memorandos internos filtrados hacia el exterior (Torvalds y Diamond, 2001). En ellos se revelaba que Microsoft consideraba al software libre como una amenaza para su estrategia de negocios. Durante el juicio alegaron que Linux era un legítimo competidor de Windows (Wayner, 2000).

El brinco de un proyecto personal iniciado en el dormitorio de un estudiante hasta convertirse en un competidor del sistema operativo más usado del mundo se dio en menos de siete años. Linux se convirtió en una opción seria para empresas y usuarios domésticos. A pesar de esto, continuó más o menos el mismo estilo de desarrollo desde sus inicios. Linus Torvalds trabaja en el núcleo durante su tiempo libre, pero en horarios de oficina tiene su empleo de programador en una empresa (Torvalds y Diamond, 2001).

Miles de desarrolladores de todo el mundo envían sugerencias y parches a Linux. Actualmente son tantas que Linus no tiene tiempo para revisarlas todas. Existe una fundación encargada de mantener el núcleo, con muchos voluntarios que se reparten el trabajo. A pesar

de esto, Linus tiene la última palabra en cuanto a los cambios. Durante un tiempo, los desarrolladores utilizaron el software comercial BitKeeper para organizar el trabajo distribuido. Linus creó un nuevo sistema para reemplazarlo de nombre Git en 2005. Este sistema permite a programadores de todo el mundo colaborar en el mismo proyecto a través de la Internet.

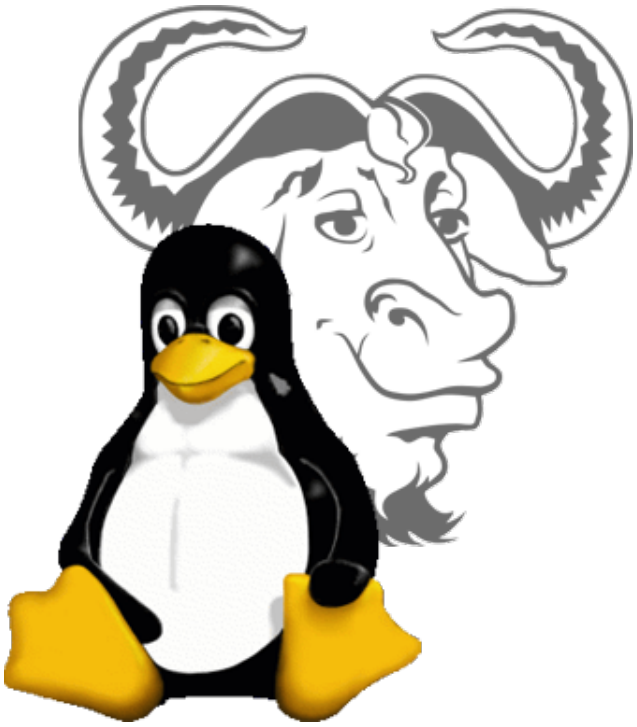


Figura 10. **:** El logotipo de Linux es un pingüino y el de GNU un ñu.

Para E. G. Coleman y Golub (2008) la práctica del *hacker* se encuentra guiada por principios liberalistas. Aunque el liberalismo se encuentre presente en el discurso de los *hackers* y éstos mencionen la libertad constantemente, los valores de éste se encuentran de diferente manera en cada práctica. Un último componente a tomar en cuenta para comprender la práctica *hacker* es el aspecto lúdico de ésta. Este es el tema tratado por el filósofo Pekka Himanen (2001): Lo que distingue a la práctica *hacker* es que se realiza con *interés*, *entusiasmo* y *gozo* de manera que se realiza en el tiempo libre. Himanen contrasta esta ética de trabajo con la ética protestante descrita por Weber para afirmar que el *hacker* posee horarios más flexibles y una motivación diferente. Mientras que el protestante buscaba ganarse el cielo como recompensa de un trabajo bien hecho, y su motivación es económica, el *hacker* mira al trabajo como un juego, como un reto. Esto implica horarios de trabajo diferentes, el desarrollo de la colaboración y la creatividad.

Esta es, a grandes rasgos, la historia de los *hackers* de sombrero blanco y su filosofía. Espero que durante el transcurso de estas secciones haya quedado bien definida la gran diferencia entre “sombreros negros” y “sombreros blancos”. Éstos últimos están interesados en

la programación como un fin en sí mismo y buscan estrategias de colaboración sin jerarquías. Con estas extensas aclaraciones, pasemos a la sección teórica y metodológica de esta tesis.

Capítulo II - La receta

Durante el transcurso de la maestría uno de mis profesores, el Dr. Manuel Ortiz, presentó uno de sus libros. El título es *Cómo hacer una rica sopa con la metodología* (2010). El título llamó mucho mi atención, al leerlo me gustó la comparación de la metodología con una receta de cocina. En toda investigación se requiere una metodología y en cada investigación ésta será diferente. Una de las primeras lecciones del libro es que no hay recetas prediseñadas, depende del talento de cada investigador el encontrar la mejor, la que mejor responda a la pregunta de investigación. Debe ser como un traje diseñado a la medida, que se ajuste lo mejor posible.

Obviamente el diseño metodológico también es parte de mi trabajo. La metodología es, a grandes rasgos, el procedimiento racional para responder a mi pregunta de investigación y cumplir mis objetivos. Esta tesis es de estudios socioculturales, por lo que la metodología surge desde esta disciplina, ya con una tradición larga y respetable.

Los estudios socioculturales buscan incorporar una visión interdisciplinaria en sus objetos de estudio. Su tradición abreva principalmente de la sociología, la antropología y la crítica literaria (Alasuutari, 1995). Como intuye por el nombre, los estudios socioculturales dependen en gran medida de un concepto sistematizado y científico de cultura.

Este concepto surgió, en primera instancia, dentro de la antropología (Giménez, 2005). Pero dentro de los estudios culturales se utilizó de manera diferente, con una fuerte influencia de la literatura. Richard Hoggart, en 1966, afirmó que los estudios culturales debían tener una sensibilidad literaria, de lo contrario no podrían "leer" apropiadamente a la sociedad (Hebdige, 1979, pág. 8).

Autores como Raymond Williams y Clifford Geertz trabajaron durante el siglo XX para formar un concepto claro y definido de cultura. De esta forma, las diferentes disciplinas podrían utilizarlo también para sus estudios. Pensar así en la cultura implica abandonar las nociones de sentido común que normalmente atribuimos a esta palabra. Hablo, por ejemplo, de la idea de que "cultura" representa únicamente "alta cultura" o bellas artes, como significó

en otros momentos históricos. Para Williams, de la cultura está constituida por los significados y las prácticas del hombre común. El lenguaje constituye elementos materiales significativos y estructura los significados que son presentados en un contexto determinado (Barker y Galasinski, 2001).

La concepción de la cultura que Clifford Geertz acuñó es la llamada "concepción simbólica". Geertz afirma que la cultura es el conjunto de "estructuras de significado socialmente constituidas" (Geertz, 2005). Esto es, los significados compartidos por grupos de individuos. Su ejemplo más famoso es el de un guiño, aunque en todos los contextos un giño es un movimiento que se hace con el párpado, significa diferentes cosas de acuerdo al contexto en donde nos encontramos. Esto implica que el análisis de la cultura no compete a las ciencias que pretenden explicar la realidad y buscar leyes; una realidad compete a aquellas que buscan interpretarla y encontrar pautas de significado.

Interpretar la cultura es similar a interpretar un texto. Es como leer "un manuscrito extranjero, borroso, plagado de elipsis, de incoherencias, de sospechosas enmiendas y de comentarios tendenciosos y además escrito, no en las grafías convencionales de representación sonora, si no en ejemplos volátiles de conducta modelada". (Geertz, 2005, pág. 24) Es por ello que estudiar la cultura es complejo, pero existen técnicas y métodos para lograrlo. Lo que busca la etnografía, entonces, es la "descripción densa": una descripción que no limitada a lo directamente observable de la realidad social, y que también interprete los significados simbólicos, aquello que no puede verse pero puede leerse (Giménez, 2005).

Una de las dificultades al estudiar a los *hackers* desde un punto de vista cultural es la de conciliar diferentes visiones sobre el tema. Las actividades de los *hackers* transitan despreocupadamente entre lo tecnológico, lo legal, lo social, lo económico y lo cultural sin que los individuos sean totalmente conscientes de ello. Es necesario complejizar la visión científica y armarse con las herramientas necesarias para estudiarlos desde la perspectiva de los estudios socioculturales. Lo que busco en este apartado teórico-metodológico es explicar y fundamentar cómo construí dicha herramienta para este estudio.

De las perspectivas teóricas que revisé en el transcurso de esta investigación, la sociología de la ciencia y la tecnología aportó las herramientas compatibles con este objeto de estudio. Ésta rama de la sociología surge como consecuencia de las críticas a la modernidad. Desde la revolución francesa y particularmente a partir de la revolución industrial, los valores de progreso y razón se incorporaron en su mayor parte de las actividades de las sociedades occidentales. Durante este periodo, se reemplazó a Dios con la ciencia y la razón (Touraine, 2000; Morley, 1998).

La supremacía de los valores modernos provocó que la ciencia se transformara en una verdad aceptada hegemónicamente. La ciencia se encontraba protegida de cualquier cuestionamiento sobre su validez, ya que sus propios métodos servían para validarla (Alexander, 1991). Diversos autores cuestionarion los principios de la modernidad, dando paso a lo que algunos denominan "postmodernismo" (Morley, 1998). Autores como Michel Foucault y Bruno Latour han cambiado esta perspectiva. El primero afirma que la verdad científica es un mecanismo de control social que se impone coercitivamente (Foucault, 1992). Latour afirma que la ciencia y la tecnología, consideradas como "verdad" por la modernidad, pueden estudiarse como cualquier otra práctica humana. Además, Latour añade que la división disciplinaria muchas veces dificulta la comprensión de fenómenos que no pueden ser entendidos desde una sola disciplina.

> Es evidente que nuestra vida intelectual está muy mal hecha. La epistemología, las ciencias sociales, las ciencias del texto, cada una tiene su casa propia, pero a condición de ser distintas. Si los seres que a ustedes les interesan atraviesan las tres, dejan de ser comprendidos (Latour, 2007, pág. 20).

Este antropólogo francés ha basado su trabajo en el estudio de los científicos, los legisladores y la tecnología, tomando varios casos como ejemplos paradigmáticos para confirmar sus teorías. Sus hallazgos demuestran que la práctica científica no es totalmente "neutra" cultural socialmente, esta práctica está cargada de significados y subjetividades y responde a preopaciones de las sociedades donde se origina. Latour es uno de los desarrolladores, junto

con Michel Callon y John Law, de la teoría del actor-red. Ésta pretende asociar elementos "humanos" con "no-humanos", creando una red que va más allá de lo social y provee una perspectiva diferente a la que se obtiene únicamente con las ciencias sociales (Latour, 2008).

Esta perspectiva ha sido utilizada en una ocasión para estudiar a los *hackers*: en la tesis doctoral "Hackers en el contexto de la era de la información" (2005) de Jorge Lizama Mendoza. En ésta Lizama se propuso a los *hackers* como a un "grupo social relevante" desde la corriente del constructivismo tecnológico (Bijker, Hughes, y Pinch, 1987), corriente que se explicará a continuación.

Sociología de la tecnología

La tecnología ha existido desde tiempos inmemoriales, desde antes incluso de que existiera la escritura o la ciencia (Basalla, 1994). En ocasiones el término se refiere a artefactos materiales o a procesos que resuelven problemas. La sociología de la tecnología pretende comprender el uso y cambio tecnológico desde una perspectiva amplia que recupere visiones de diferentes disciplinas. Las propuestas de la sociología de la tecnología intentan contrarrestar visiones simplistas del desarrollo tecnológico. Una de ellas se denomina "determinismo tecnológico" y asume que los inventos e innovaciones son producto de mentes geniales cuyas ideas crean una discontinuidad con respecto al pasado. Es decir, los nuevos inventos son innovadores por que no tienen relación con lo que se ha hecho en el pasado.

La sociología de la tecnología parte de la premisa de que la tecnología es una construcción social. Al ser producto del ser humano, debe comprenderse en relación a su contexto e historia. George Basalla, en su libro *La evolución de la tecnología (1994)* afirma que se puede comparar la teoría de la evolución de Darwin con respecto a la tecnología. Es decir, todos los artefactos actuales provienen de uno anterior, y la adaptación de los mismos se pude explicar por múltiples factores.

Basalla propone como ejemplo el hacha, cuya forma actual proviene de antiguas piedras afiladas, pero hay muchos más. Las computadoras actuales remiten a antiguas máquinas de escribir del siglo XIX, por ejemplo: "Ninguna sociedad está aislada o es autosuficiente de

forma que nunca haya tomado prestados al menos algunos aspectos de su tecnología de una fuente exterior" (Basalla, 1994, pág. 104). El autor rastrea los motivos del cambio tecnológico y concluye que son tanto sociales, económicos, ambientales, psicológicos, culturales y migratorios. Pero siempre existe un precedente, una base previa para ese artefacto novedoso: "Para que se dé un cambio evolutivo, la novedad ha de hallar una forma de afirmarse en medio de la continuidad" (Basalla, 1994, pág. 88).

Existen varios análisis en esta corriente. Pueden dividirse en tres categorías: Constructivismo social, metáfora de sistemas y la teoría del actor-red (Bijker y cols., 1987). Las tres parten de supuestos similares y comparten características en común. En primer lugar, suponen que la tecnología se construye socialmente. En segundo lugar, que los constructores de sistemas no necesariamente respetan las categorías de conocimiento de la ciencia ni los límites disciplinares. Al interrogar a un ingeniero si su práctica es tecnológica o científica, posiblemente no sepa qué responder, aunque su práctica cruce e involucre varias disciplinas (Latour, 2007; Bijker y cols., 1987).

Es por ello que la perspectiva de metáfora de sistemas busca analiza prácticas y la tecnología no como entes separados de la vida social, más bien como parte de ellos. La sociedad no como algo aparte de la naturaleza, como sistemas que interactúan y pueden modificarse mutuamente, siguiendo de cerca las ideas de Latour:

> La tarea de la antropología del mundo moderno consiste en describir de la misma manera el modo en que se organizan todas las ramas de nuestro gobierno, inclusive la de la naturaleza y las ciencias exactas, y explicar de qué manera y por qué esas ramas se separan, así como los múltiples arreglos que las reúnen (Latour, 2007, pág. 35).

Para Layton (1977, citado por Bijker y cols., 1987) es necesario comprender la tecnología "desde dentro", en vez de tener una perspectiva de "caja negra". Pensar la tecnología desde esta perspectiva implica pensarla como parte del sentido común, obviando su lógica. El libro *The Social Construction of Technological Systems* (1987) surge como reszpuesta a la falta de

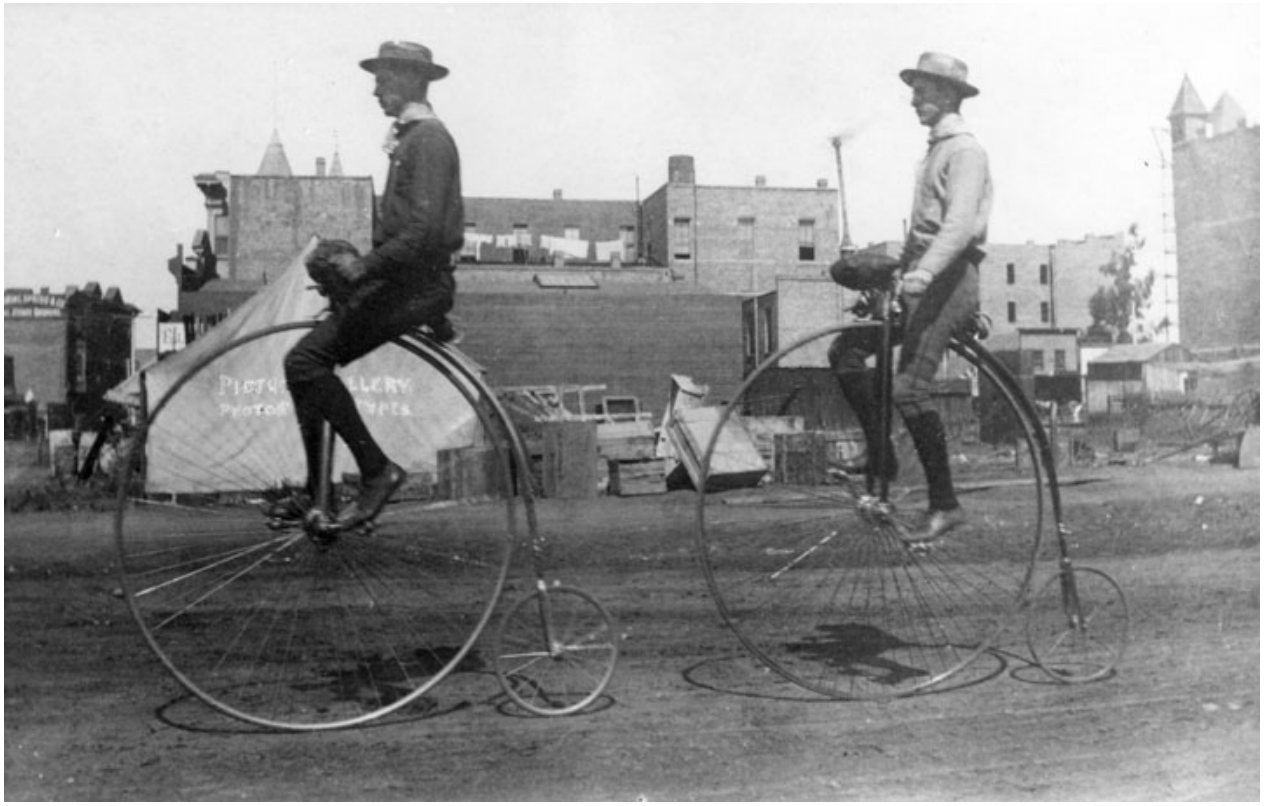


Figura 11. **:** Bicicleta Penny Farthing, con una gran rueda frontal. Esta permitía a los ciclistas mantener la velocidad suficiente para no perder el equilibrio.

estudios de este tipo. Es una colección de ensayos de diferentes autores (entre ellos Thomas Hughes, Michel Callon y John Law) donde pretenden ilustrar esta perspectiva ilustrando la teoría mediante ejemplos paradigmáticos. En esta obra se estudian varios casos de desarrollos tecnológicos haciendo uso de las herramientas de la sociología de la tecnología, entre ellos la bicicleta, el sistema eléctrico, las carabelas y los misiles teleguiados.

Un ejemplo que particularmente me gusta es el de la bicicleta, en parte porque me transporto frecuentemente en bicicleta. Trevor Pinch y Wiebe Bijker analizan este caso en el artículo "The Social Construction of Facts and Artifacts" (Bijker y cols., 1987). Se remontan a la bicicleta Penny Farthing, la cual poseía una gran rueda frontal. Los pedales se encontraban directamente en la llanta frontal, lo cual permitía un rodado suficiente para no perder el equilibrio. Estas bicicletas eran difíciles de montar.

Los autores afirman que diversos grupos sociales poseían interés en este modelo de

bicicleta. Bajo la perspectiva de la construcción social de la tecnología, es requisito que este grupo social relevante comparta el mismo conjunto de significados asociados con un artefacto (Bijker y cols., 1987, pág. 30). Uno de estos grupos sociales fueron las mujeres, quienes poseían un interés por utilizar la bicicleta, pero tenían un impedimento cultural: La falda.

La bicicleta Penny Farthing surgió durante la segunda mitad del siglo XIX. En las sociedades donde se popularizó, era impensable para una mujer no utilizar falda. Por este motivo, no era una buena candidata para montar en este tipo de bicicletas. Pero los ingenieros y productores vieron en las mujeres potenciales ciclistas, así que varios modelos fallidos se desarrollaron pensando en este grupo social relevante. Aunque técnicamente hubiera sido más fácil que las mujeres vistieran pantalón para montar bicicleta, culturalmente era inaceptable, por lo que el artefacto sufrió modificaciones.

Durante un periodo de casi veinte años, los modelos de bicicleta cambiaron hasta que se encontró una solución al problema. Este modelo se conocía en un principio como "bicicleta de seguridad". Ésta poseía una llanta frontal del mismo tamaño que la llanta trasera. Esto se consiguió cambiando la posición de los pedales, y vinculándolos con la llanta trasera mediante engranes y una cadena. El resultado fue una bicicleta menos alta, y que era compatible con la forma de vestir de las mujeres de la época.

A través de este interesante ejemplo, Pinch y Bijker ilustran cómo funciona la construcción social de la tecnología. Los artefactos no son sólamente avances con respecto a los anteriores, responden a demandas específicas por parte de grupos sociales relevantes. La cultura de las sociedades en donde estos artefactos se desarrollan son cruciales para comprender su desarrollo.

Ética

La ética es una rama de la filosofía que estudia la bondad y maldad de los actos humanos. Sus raíces son tan profundas que llegan al menos hasta Aristóteles y se ramifican a lo largo de toda la tradición helénica. La ética es un concepto retomado por muchos autores que hablan sobre los *hackers* y ciertamente es uno con una larga historia. ¿Qué es actuar correctamente

Figura 12. **:** El modelo conocido como "bicicleta de seguridad" surgió 19 años después de la Penny Farthing y permitía que mujeres con falda pudiesen montarla.

para un *hacker*? La pregunta es interesante a la luz de una revisión histórica del fenómeno. Este tipo de consideraciones es tan importante que según algunos autores, como Hegel, forma nuestra identidad (Blackburn, 2003).

Aristóteles escribió un tratado sobre ética como parte de una obra más grande sobre la conducta humana. En ésta, describe qué virtudes y cualidades debe tener un buen ciudadano, también explica cómo ellas pueden inculcarse (Aristóteles, 2007). Aristóteles pensaba que los comportamientos correctos y deseables son aquellos que conducen a la felicidad. Esto se logra mediante la razón y apegándose a la naturaleza. Para él, los valores eran formas abstractas y perfectas que se aplicaban universalmente y por cada valor existía un vicio que se le opone (Aristóteles, 2007; Ritzer, 2005).

> Las cualidades éticamente relevantes en nosotros son nuestros hábitos, y [...] cada uno es el conjunto de los hábitos que forman su carácter (su *ethos*), es decir, su "personalidad" empírica, moral y psicológica potencial, la cual [...] se expresa en la actitud o en la disposición que adoptamos hacia nuestros afectos en general, y que se traduce o se actualiza en la realidad de la acción (*praxis*) (Sinnott, 2007, págs. XXXII-XXXIII).

El concepto de la ética ha sido tratado por muchos autores y en general se buscaba un conjunto de valores y comportamientos que fueran buenos y se aplicaran universalmente. Esta fue una de las bases de la ética que propuso Emmanuel Kant durante el siglo XIX.

> Kant describe sus sistema ético, basado en la idea de que la razón es la autoridad última de la moral. Afirmaba en sus páginas que los actos de cualquier clase han de ser emprendidos desde un sentido del deber que dictase la razón, y que ningún acto realizado por conveniencia o sólo por obediencia a la ley o costumbre puede considerarse como moral (Rutiaga, 2004, pág. 7).

Las críticas a la modernidad provocaron que se cuestione esta visión universal. El intentar asumir la visión de Kant significó para ciertos filósofos imponer valores a individuos y culturas que podrían no compartirlos. Por otra parte, los códigos éticos poseen, por su propia naturaleza, contradicciones internas que no pueden resolverse.

> El código ético a prueba de tontos -con fundamentos universales e inamovibles- nunca se encontrará y, después de habernos quemado los dedos con demasiada frecuencia, ahora sabemos lo que no sabíamos entonces, cuando nos embarcamos en ese viaje de exploración: que una moral no aporética, no ambivalente, una ética universal y con "fundamentos objetivos" es una imposibilidad práctica, quizás incluso un oxímoron, una contradicción (Bauman, 2006, pág. 17).

Varios autores que han tratado el tema de las prácticas tecnológicas y los *hackers* han utilizado la ética como categoría de análisis. Comenzando por el mismo Steven Levy, quien veía un

código ético que denominó "la ética del *hacker*". A partir de entonces, diferentes autores (G. Coleman, 2005; G. R. Meyer, 1989; Taylor, 1999; Himanen, 2001) han encontrado que una de las características que definen al *hacker* es una ética determinada. Ya comprobamos que tanto Stallman, Torvalds y Raymond poseen claras posiciones éticas con respecto al software y la propiedad intelectual.

Raymond publicó en Internet un archivo conocido como *The Jargon File*. Después de varios de existencia en línea, se editó en papel bajo el nombre de *The hacker's dictionary* (1999). Se trata de un glosario semi-humorístico con términos utilizados coloquialmente en el lenguaje *hacker*. Se enfoca casi totalmente a los *hackers* tradicionales, no a los intrusos o al *underground* informático. En este diccionario se encuentran definiciones para "Good Thing" y "Bad Thing" (Cosa Buena y Cosa Mala, con mayúsculas). Estos términos se utilizan con frecuencia para describir valores deseables para la comunidad *hacker*, sobre todo dentro de la programación.

En cuanto al uso de la tecnología, Taylor (1999) dedica un capítulo completo al tema de nombre "The Construction of Computer Ethics". En él, se buscan las diferentes opiniones sobre la tecnología que tienen tanto los *hackers* como la industria de Internet y la seguridad informática. Taylor encuentra que uno de los principales conflictos ente ellos son los valores éticos, y esta diferencia se da debido a la discrepancia de edad, de tradiciones y de objetivos que se encuentran en ambos casos.

La distinción entre "sombreros negros" y "sombreros blancos" es también ética. Ya que tanto a unos como a otros les interesan las mismas tecnologías, aunque posean una ética diferente. Por esto me parece importante tomar la ética como uno de los componentes importantes para comprender a los *hackers* presentados en esta tesis. La ética puede analizarse a través del discurso, como demostraré más adelante. Sin embargo, también importa analizar la *praxis*, o la práctica de los *hackers*: las acciones concretas que arrojan luz sobre la ética de los *hackers*. Por ello el concepto de "práctica social" también es relevante.

Práctica social

El analizar prácticas que involucran computadoras y sistemas computacionales parece una tarea para la ingeniería. Pero coomo ya argumenté, la sociología de la tecnología nos revela que existe una importante dimensión cultural en todo artefacto tecnológico. Quisiera llamar la atención a un brevísimo punto que en particular me parece relevante para entender esto. Durante su artículo "The Social Organization of The Computer Undergound", Gordon Meyer menciona un punto al que no concedió mucha importancia, supongo que se salía de su objeto de estudio. Por tanto lo añadió como nota al pie:

> El equipo básico [del *hacker*] consiste de un módem, una línea telefónica y una computadora - todos ellos dispositivos que se encuentran disponibles a través de medios legítimos. Lo que distingue a los *hackers* de otros usuarios es la forma en que se utiliza este equipo y el conocimiento que para ello se requiere.(G. R. Meyer, 1989, pág 34)[20]

En pocas palabras, los *hackers* tienen a su disposición la misma tecnología que el resto de los usuarios. Cualquier consumidor promedio puede comprar un módem, una computadora o dispositivos electrónicos cualquiera. Como he reiterado en varias ocasiones, la diferencia entre el *hacker* y otras personas es el uso que dan a la tecnología. Y esa diferencia puede entenderse particularmente en el campo de lo simbólico.

Existe un artículo de Jeffrey Alexander que fortalece el argumento. Tiene como título: "Sociología cultural de lo sagrado y lo profano en el discurso tecnológico" (Alexander, 1991). El autor parte de los cuestionamientos del postmodernismo y se pregunta: ¿es verdaderamente posible un mundo de racionalidad técnica total, que deje de lado el pensamiento mágico? Alexander concluye con la premisa que, mientras que la tecnología es un objeto tangible y que puede medirse y razonarse, forma parte del sistema cultural. Es un signo con significante

[20]"The basic equipment consists of a modem, phone line, and a computer – all items that are available through legitimate channels. It is the way the equipment is used, and the associated knowledge that is required, that distinguishes hackers from other computer users."

y significado al cual no se le puede retirar totalmente su carga subjetiva, o simbólica. "Esto quiere decir que debemos aprender a ver la tecnología como un discurso, como un sistema de signos sujeto a fuerzas semióticas y que responde a demandas sociales y psicológicas" (Alexander, 1991, pág. 296).

Posteriormente, Alexander presenta un ejemplo ilustrativo: la representación de la computadora en la publicidad. A través de anuncios y reportes periodísticos de varias décadas, Alexander demuestra que, a pesar de hablar de un artefacto tecnológico objetivo, las connotaciones utilizadas oscilan entre la salvación y la destrucción. En un momento las computadoras son dispositivos que solucionan todos nuestros problemas y resolverán hasta los más pequeños detalles de nuestras vidas, y en el siguiente podrían destruir y arruinar a toda la civilización. Ante la perspectiva de un futuro totalmente tecnologizado, Alexander concluye: "Sólo entendiendo la conformación omnipresente de la conciencia tecnológica en el discurso, podemos esperar obtener control sobre la tecnología en su forma material" (Alexander, 1991, pág. 306).

Ésta es una de las premisas con las que quisiera pensar la práctica social: la tecnología no es únicamente racional, también se encuentra imbricada por aspectos culturales y sociales. La comprensión de estas es importante como contraste a las visiones racionalistas y deterministas de la tecnología. Los componentes de la práctica social se explicarán más adelante.

Me limité a estos conceptos para entender el fenómeno de los *hackers*, pero otros también serían adecuados. Especialmente el de "capital social" que proviene de teorías desarrolladas por Pierre Bourdieu, quien se basó en teorías económicas para entender lo social (Lin, 2002). Los cuatro componentes del capital social (información, influencia, credenciales sociales y refuerzos) podrían ayudar a comprender las relaciones de los *hackers*. ¿Cómo obtienen validación en su campo? ¿Cómo se dan las redes de colaboración? Sin embargo, he decidido, como ya mencioné, en la ética de los *hackers* dado a que me parece un tema menos explorado y con mayores posibilidades de aportar conocimiento nuevo. Con estas puntualizaciones tomadas en cuenta, presentaré a continuación la pregunta y los objetivos de mi investigación.

Pregunta de investigación. Ya con el tema delimitado, quiero señalar cuál es la pregunta de investigación. Esta pregunta es el motor del trabajo determina todas las decisiones que tomé posteriormente: ¿Cuál es la postura ética que los *hackers* tienen ante la tecnologías de la información?

Objetivos. Para responder a la pregunta de investigación, es necesario plantear un objetivo general, del cual se desprenden objetivos específicos más finos. Esto divide a la investigación en partes manejables que pueden resolverse de manera más sencilla y comprensible. Los objetivos de esta investigación son los siguientes.

Objetivo general

Comprender las prácticas sociales de los *hackers* para determinar su postura ética ante las tecnologías de la información.

Objetivos específicos

- Analizar las prácticas sociales de *hackers*.
- Determinar las postura política de los *hackers* frente a la tecnología a través de sus discursos.
- Analizar éticamente los usos sociales de la tecnología.

Metodología

Toda investigación, del enfoque que sea, tiene dos pasos fundamentales: recoger toda la información necesaria para cumplir los objetivos y estructurar la información en un todo coherente (Martínez Miguelez, 2006). Los objetivos planteados y el giro cultural de la tesis nos llevan a los territorios de la investigación cualitativa. Este tipo de investigación científica se aleja de la investigación cuantitativa y, en vez de de cuantificar los datos, intenta valorarlos. Los datos se consideran una totalidad y no pueden aislarse de su contexto. En investigación cualitativa los investigadores son agentes sensibles a los cambios que ellos mismos producen en sus objetos de estudio (Alasuutari, 1995; Martínez Miguelez, 2006). Existen varios paradigmas para emprender este tipo de investigaciones. El interaccionismo simbólico, la teoría

fundamentada y el análisis conversacional, para nombrar algunos. Para esta investigación utilicé el método etnográfico.

Etnografía. La etnografía es un método desarrollado en el seno de la antropología. Mediante este método, "el investigador se propone interpretar/describir una cultura para hacerla inteligible ante quienes no pertenecen a ella" (Guber, 2001). Antropólogos como Bronislaw Malinowski desarrollaron principios mediante los cuales se sistematizó la observación de la cultura. De esta forma surgió el rol del "etnógrafo", el individuo que convivía directamente con los miembros de la cultura que pretendía estudiar (primordialmente indígenas) y obtenía datos mediante observación directa, interacciones cotidianas y otros métodos. El resultado de ello era una etnografía, una descripción de los usos y costumbres de una cultura ajena (Malinowski, 1986; Mead, 2000).

Con el paso de los años surgieron críticas al método etnográfico tradicional. Una de las objeciones afirmaba el que los antropólogos viajaran a las sociedades denominadas "salvajes" para explicarlas científicamente posee una fuerte carga colonizadora. Por otra parte, ¿por qué se consideraban aquellas culturas como exóticas, primitivas y salvajes tan sólo por no apegarse a las normas occidentales? ¿Por qué no también se hacían estudios de esas sociedades "avanzadas"? (Guber, 2001). Paulatinamente, estos cuestionamientos culminaron en posturas más críticas y reflexivas sobre el método. La etnografía en la actualidad intenta desligarse de sus implicaciones coloniales y, por supuesto, es saludable que el investigador tome en cuenta sea reflexivo en torno a estos temas (Fabian, 1983; Evans-Pritchard, 1990; Harris, 2001).

La etnografía se concreta a partir de varias técnicas. Una de ellas, la técnica por excelencia, es la observación participante (Martínez Miguelez, 2006). A través de dicha observación, el investigador interactúa con los individuos en escenarios específicos y busca "implicarse en actividades concernientes a la situación social a estudio, y observar a fondo dicha situación" (Valles, 1999, pág. 150).

El observador se encuentra en la curiosa situación de ser miembro del grupo y a la vez un extraño, por aplica su introspección como instrumento de investigación social. El objetivo es

interactuar con ellos como un miembro más y conectarse con el mundo empírico y utilizar su propia subjetividad para obtener datos que respondan a la pregunta de investigación (Guber, 2001).

Cuando el antropólogo tradicional hacía etnografía de los pueblos "salvajes", normalmente actuaba en el campo. Es decir, una aldea, la jungla, una isla. En pocas palabras, un lugar alejado de la modernidad. Los cuestionamientos de la antropología también cambiaron la noción de "campo". Hoy es mucho más difícil definirla claramente (Clifford, 1999). El trabajo de campo hoy en día podría implicar no viajar grandes distancias. El campo podría estar localizado en la misma ciudad del investigador. Podría accederse públicamente como un centro comercial o un cine. También un laboratorio científico o una escuela. Incluso podría no estar en un espacio físico y trabajo de campo en el ciberespacio, o ciberetnografía. Ésta vertiente de la etnografía es similar, pero contempla la interacción humana a través de mediaciones tecnológicas (E. G. Coleman, 2010; Rybas y Gajjala, 2007; Ruiz Torres, 2008).

En esta investigación consideré que la ciberetnografía no es apropiada para responder a mi pregunta de investigación. En cambio, utilicé técnicas más tradicionales como la observación participante, en la que el investigador asume un rol dentro de las actividades de los agentes a estudiar. Otra técnica importante fue la entrevista, que es "una estrategia para hacer que la gente hable sobre lo que sabe, piensa y cree" (Guber, 2001, pág. 30). La etnografía ha sido utilizada en trabajos académicos anteriores para estudiar a los *hackers* (G. R. Meyer, 1989; Contreras, 2004; E. G. Coleman, 2010; G. Coleman, 2010) debido a que los *hackers* poseen valores, costumbres y ritos, como muchos grupos sociales. En algunas ocasiones, incluso, las investigaciones han tenido un giro ciberetnográfico.

Selección de los informantes. He decidido privilegiar las interacciones presenciales en vez de las "virtuales" por diversos motivos. Los *hackers*, a diferencia de lo que podría pensarse, por lo general buscan los encuentros cara a cara en medida de lo posible. Evidencia de esto puede encontrarse en las múltiples reuniones *hacker* que se celebran en todo el mundo. Desde reuniones locales, con asistentes regulares de comunidades cercanas, hasta globales en donde

se encuentran personas de todo el mundo.

Un ejemplo paradigmático para ilustrar esto son las reuniones 2600. Éstas son convocadas por la revista neoyorkina *2600: The Hacker Quarterly* los primeros viernes de cada mes. Se realizan en diferentes partes del mundo y son autogestionadas. Los lugares elegidos normalmente son céntricos, abiertos al público y con acceso a Internet y teléfonos públicos. Ciertas reuniones, por ejemplo, se realizan en La Jolla, una colonia adinerada en donde se encuentra la UCSD (University of California, San Diego). En Phoenix, Arizona las reuniones se efectúan en un café de la Central Avenue. En Los Ángeles en el *downtown* y en México, D.F., cerca del Zócalo. La página oficial[21] incluye recomendaciones para iniciar una reunión 2600. Entre ellas la siguiente:

> Aunque las reuniones no se limitan a grandes ciudades, la mayoría ocurre en grandes áreas metropolitanas que son fácilmente accesibles. Puede convenir una reunión en tu pueblo, pero invitamos a la gente a asistir a reuniones donde conocerán gente del área más amplia posible. Así que si hay una reunión a una hora o dos de distancia, asiste a esa en vez de tener dos reuniones pequeñas bastante cerca una de la otra. Siempre tendrás la oportunidad de reunirte con los "*hackers* de tu pueblo" en el momento que gustes.[22]

Como puede apreciarse, los *hackers* no buscan en absoluto reemplazar las interacciones presenciales, todo lo contrario. Incluso en mi primera visita al *hackerspace* Noisebridge, en San Francisco, comenté que muchos académicos creen que los *hackers* prefieren comunicarse a través del ciberespacio. Mi interlocutor se llevó la mano a la barbilla pensativo y replicó: "Es interesante que alguien pudiera pensar eso", como si fuera una posibilidad que no hubiese pasado por su cabeza.

---

[21] `http://www.2600.com/meetings/guidelines.html`

[22] "While meetings are not limited to big cities, most of them take place in large metropolitan areas that are easily accessible. While it's convenient to have a meeting in your home town, we encourage people to go to meetings where they'll meet people from as wide an area as possible. So if there's a meeting within an hour or two of your town, go to that one rather than have two smaller meetings fairly close to each other. You always have the opportunity to get together with 'home town hackers' any time you want."

Gabriella Coleman afirma que las reuniones *hacker* sirven como "rituales de condensación" de la interacción. Es decir, en ellas se vive una interacción acelerada e intensificada debido a que los individuos interactúan casi siempre en línea y sólo unos cuantos días al año se encuentran en persona (G. Coleman, 2010). Esta afirmación contradice la mirada de algunos autores, incluyendo a Manuel Castells, que afirman que los *hackers* "serios" habitan exclusivamente en línea (Castells, 2003; Contreras, 2004).

Pero esto no niega la existencia de redes más ocultas de comunicación *underground*, en donde se comparta información posiblemente ilegal. Varios autores se han estudiado este tipo de redes y prácticas (Gordon, 1996; G. R. Meyer, 1989; G. Meyer y Thomas, 1990; Oré, 2007; Denning, 1996). Sin embargo, todos ellos enfrentaron la dificultad metodológica del difícil acceso a esa información y a la reticencia de los informantes de hablar sobre el tema. Además, un principio *hacker*, como lo mencioné al principio, es compartir la información libremente. En su mayor parte, las dinámicas de participación de los *hackers* funcionan para permitir la colaboración (Levy, 1984; Himanen, 2001). Un sistema que prohíba la participación o que se encontrara tan herméticamente cerrado sería lento y contraproducente para los propósitos de los *hackers*. Por ello normalmente no se restringe el acceso a las reuniones y convenciones *hacker*.

Un primer paso para conseguir a los informantes es la aproximación exploratoria a eventos y reuniones frecuentadas por *hackers*. La observación participante es una herramienta adecuada para recopilar datos para el análisis y conocer la cultura *hacker* actual (Valles, 1999; Flick, 2004; Barker y Galasinski, 2001). En la investigación cualitativa no se muestrea estadísticamente, a diferencia de la cuantitativa. Esto por dos motivos: Martínez Migueles (2006) afirma que hay situaciones en donde se pueden buscar "casos representativos". Es decir, individuos que posean las características que se están buscando, casos particularmente reveladores y que presenten mayores ventajas para los objetivos de nuestra investigación (Martínez Miguelez, 2006; Valles, 1999).

Para encontrarlos, visité los escenarios frecuentados por los *hackers*. Especialmente las

reuniones 2600 y *hackerspaces*. Dentro de ellas, busqué individuos dispuestos a hablar, ya que muchos buenos candidatos a informantes no estaban dispuestos a hacerlo. También busqué que los individuos a participar en la investigación contribuyeran activamente a proyectos libres, tanto de software como de hardware, pero guiados por los principios propuestos por Richard Stallman, Linus Torvalds y Eric Raymond, principalmente.

Análisis del discurso. Para analizar los datos utilicé la perspectiva del análisis crítico del discurso. Esta perspectiva provee las herramientas y la comprensión necesaria para analizar el lenguaje y tomarlo en cuenta en la constitución y regulación del mundo social y puede ir muy de la mano con la etnografía, que lidia en gran parte con discursos y textualidades (Barker y Galasinski, 2001). Para esta perspectiva, el significado dentro del lenguaje se genera a través de relaciones de diferencia entre signos. Como consecuencia, el lenguaje no genera "verdad", debe ser interpretada en relación directa entre palabra y realidad. En pocas palabras, el discurso debe ser intepretado. Aunque la etnografía y el discurso entran en el territorio de la interpretación y el análisis cualitativo, hay que basar el análisis en evidencia "dura".

Esta perspectiva se enriquece mediante los beneficios de introspección lingüística. Este tipo de análisis revela marcos ideológicos en prácticas discursivas y se basa en estructuras léxico-gramaticales del lenguaje (Barker y Galasinski, 2001). A través de lo que los *hackers* afirmaron en entrevistas, se puede hacer un análisis enfocado en las categorías que se desprenden de los conceptos centrales elegidos.

A continuación presentaré tres ejemplos representativos en los cuales, a través del discurso de los informantes, atisbaremos las posibles respuestas a mi pregunta de investigación. Cada informante responde particularmente a uno de los objetivos específicos planteados en esta investigación, los cuales se detallarán a continuación. También presentaré el caso del informante en cuestión.

Capítulo III - Tres hackers

En este capítulo se plasman los datos obtenidos gracias a los tres *hackers* que decidieron colaborar en esta tesis. Es interesante que en el transcurso de esta investigación, mucha gente me preguntaba si no sería demasiado difícil encontrar *hackers* que quisieran hablar conmigo. Y ya si accedían, ¿estarían dispuestos a mostrar su nombre real? ¿Su rostro? ¿Contarían la verdad o inventarían hazañas fantásticas?

Creo que ellos estaban pensando en *hackers* de sombrero negro. El mundo de los sombreros blancos es muy diferente, como quizá ya se vio en la revisión de literatura. Como Linus Torvalds, los tres *hackers* entrevistados no tienen ningún interés en ocultarse, no utilizan seudónimo y no están interesados en explorar sistemas ilegalmente. El software y las tecnologías libres, en cambio, les llaman muchísimo la atención y son parte integral de sus vidas.

Los tres me autorizaron publicar su nombre real. El caso de cada uno responde particularmente bien a uno de los objetivos específicos. Antes de hablar sobre cada *hacker,* explicaré brevemente lo que se intento lograr con el objetivo. Presentaré luego una breve biografía de cada *hacker*, resaltaré los datos importantes que responden la pregunta de investigación. Llevé a cabo la mayor parte del trabajo de campo en 2011, con tan sólo algunas entrevistas en 2012.

Analizar las prácticas sociales de los hackers

Como ya mencioné varias veces, la tecnología durante los inicios de la modernidad fue considerada herramienta neutral y completamente racional (Feenberg, 2002; Basalla, 1994). Los cuestionamientos del postmodernismo lograron que la ciencia lograra admitir conocimientos y significados subjetivos al uso y representación de la tecnología (Alexander, 1991). Bajo este punto de vista, es posible pensar en las prácticas que involucran a la tecnología como una práctica social. Me gustaría retomar el concepto de práctica social como ha sido explicado por Andreas Reckwitz (2002). El distingue varias características propias de una práctica social, las cuales listaré a continuación para explicar también como se asocian con

las prácticas de los *hackers* presentados en esta investigación.

- Cuerpo: Las prácticas consisten en actividades corporales rutinizadas, en actitudes y comportamientos recurrentes. Esto no quiere decir que en todos los casos sean físicas, también pueden incluir actividades mentales y emocionales que, a final de cuentas, son corporales. En el caso de las prácticas tecnológicas, y sobre todo las concernientes a la computadora, pareciera que no hay grandes movimientos corporales. Esto no es así, siempre hay una interfase humano-máquina que requiere movimiento del cuerpo. Mediante un teclado, un ratón, una pantalla táctil, o inclusive un sensor cerebral.

- Objetos: La mayor parte de las prácticas sociales consisten en relaciones entre varios agentes y objetos. En la cuestión de la tecnología, esto significa artefactos tecnológicos tanto de software como de hardware.

- Conocimiento: Las prácticas sociales contienen formas específicas de conocimiento. El conocimiento es una manera particular de entender al mundo, lo cual incluye una forma de comprender a los objetos, incluyendo a los objetos abstractos, y a las personas, incluso al sí mismo. En la cuestión de la tecnología, por supuesto se encuentra el conocimiento científico, los principios que rigen al software, pero también una gran cantidad de conocimiento sobre el funcionamiento y uso del software y hardware.

- Discurso/lenguaje: Es la forma en como el mundo se construye a través del lenguaje. En el caso de esta tesis, al utilizar un análisis del discurso, el lenguaje tiene un lugar muy privilegiado en la investigación. Las prácticas de los *hackers* no sólo son observables, también tienen discursos e ideas asociadas. A través de sus discursos podemos enterarnos de sus motivaciones, valores y subjetividad general, a pesar de estar lidiando con artefactos y conocimientos supuestamente objetivos.

- Estructura/proceso: Son las estructuras que surgen de la rutinización de las prácticas y que son observables a través de estas. Todas las prácticas tanto de programación, como de electrónica y otras más se encuentran altamente estructuradas. De la misma forma, llevarlas a cabo implica seguir o no estándares establecidos y normas aceptadas para el uso de esa

tecnología.

• Agente/individuo: En la teoría de la práctica los individuos no son totalmente autónomos ni totalmente restringidos por las estructuras. En este caso los individuos son los *hackers* que realizan las prácticas. No sólo los informantes seleccionados para esta tesis, también aquellos otros usuarios de tecnologías cuyas prácticas difieren de la de los *hackers*. Quisiera presentar como ejemplo el caso de un *hacker*. Es un usuario de tecnología y tiene a su disposición exactamente las mismas tecnologías que el usuario promedio. Sin embargo, a través de sus prácticas y lo que dice de ellas, podremos darnos cuenta de los usos y significados que le da a éstos.

Carlos Iván Sosa. El último jueves de cada mes se realiza en Mexicali una reunión de programadores de *open source*. En esta reunión fue donde surgió la idea para el recetario de mi mamá. Se iniciaron a principios de 2010 y continúan gracias a la iniciativa de programadores entusiastas. Con el tiempo se integraron más asistentes, entre ellos yo. Presencié varias reuniones cuando iniciaba la investigación para enterarme de lo que se hace ahí y ver si encontraba algún informante. En una sesión promedio se dan pláticas sobre tecnologías específicas con ejemplos prácticos que se muestran en un proyector o en un monitor grande. Escuchamos pláticas sobre Blueprint CSS, *Test driven development* y *Behaviour driven development* y el *framework* Symphony, entre otras tecnologías libres.

Aporté a las reuniones con algunas pláticas sobre software libre y cuestiones no muy técnicas. Comparado con los demás, soy un verdadero principiante en programación. Cuando recién me uní, me encontraba muy poco actualizado. Mis clases de programación y desarrollo en ingeniería habían ocurrido muchos años antes y había perdido habilidad al interpretar código fuente o aprender nuevas tecnologías rápidamente. Las reuniones despertaron mi interés y empecé a utilizar herramientas de software más avanzadas para facilitar mis tareas cotidianas, como LaTeX, que utilicé para dar formato a esta tesis.

Conocí a Carlos Iván durante estas reuniones. Hizo algunos parcos comentarios en una de mis pláticas, complementando con datos aislados lo que yo decía. No le presté mucha

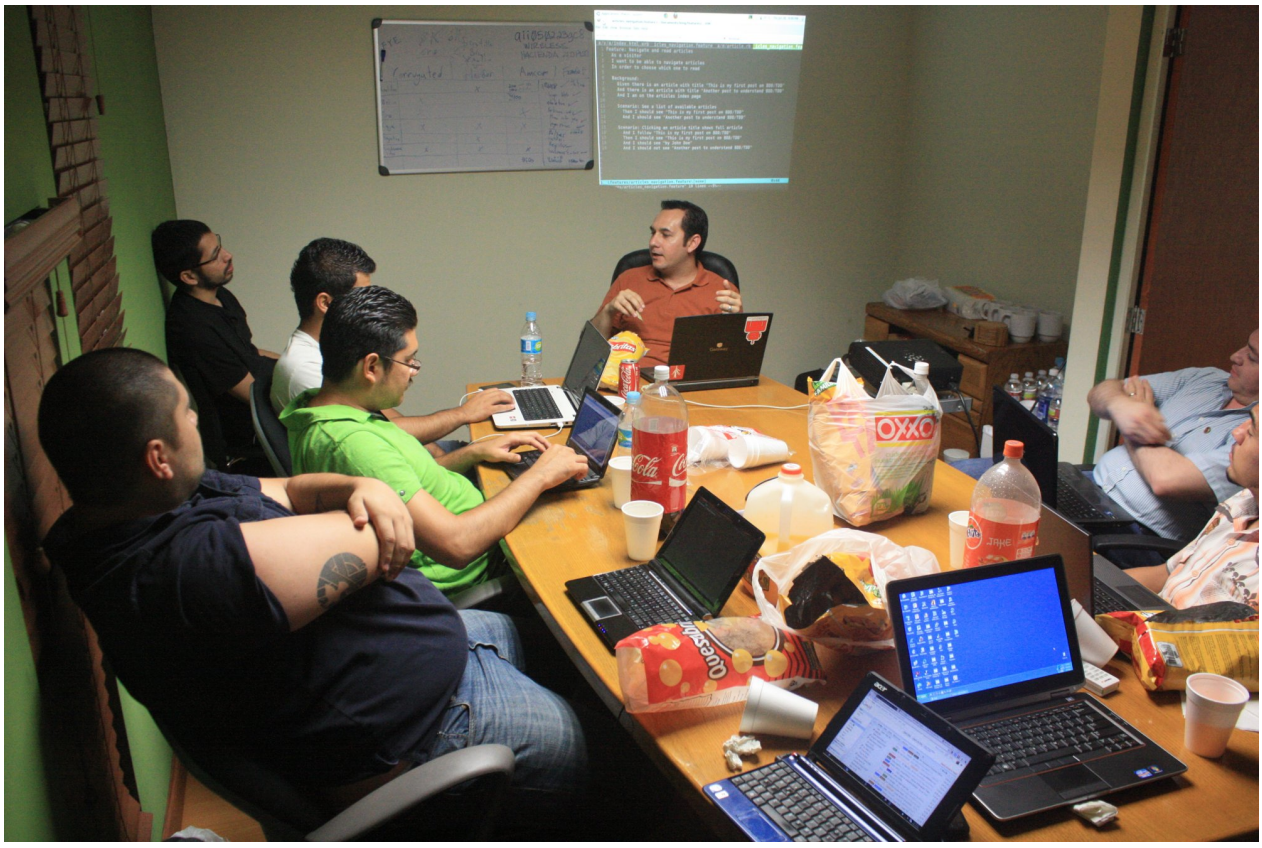


Figura 13. **:** Una reunión típica de mxlOpenSource. Esta ocurrió en las oficinas del periódico industrial Siglo XXI en Mexicali el 28 de julio de 2011. El tema de la plática era *Test driven development* y *Behaviour driven development*. Carlos Iván se encuentra reclinado en su asiento en la extrema izquierda.

atención al principio. Me comentó que trabajaba en Imperial, del otro lado de la frontera. Llamó mi atención como posible informante el día que dio un taller sobre Vim, un editor de texto para sistemas UNIX.

Vim es un editor que puede parecer primitivo a simple vista, comparado con otro tipo de programas. A diferencia de software comercial como Microsoft Office o la suite iWork de Apple, Vim es un editor libre que no posee una interfaz gráfica. Se maneja exclusivamente a través del teclado.

Antes de comenzar el taller, Carlos sacó una *netbook* de su mochila y un teclado que conectó a la computadora. Esto me pareció extraño y le pregunté cuál era la ventaja de conectar un teclado aparte. Me respondió que se trataba del teclado "Happy hacking." Este

teclado, me explicó, intenta imitar los teclados de las terminales UNIX de los años sesenta y setenta. Me mostró que no posee la tecla "Caps Lock" y es reemplazada por la tecla "Control".

En un principio no vi mucha utilidad a este cambio. Carlos me explicó: "¿Qué tanto usas la tecla 'caps lock'?". Contesté que casi nunca. Añadió: "¿Vez? Esa es una tecla inventada por los huevones de Windows. La tecla 'control' se usa mucho más y está en un lugar de difícil acceso". Me demostró en un teclado normal que la tecla "control" se tiene que presionar con el dedo meñique incómodamente y que en el teclado "Happy hacking" se encuentra en un lugar mucho más ergonómico.

Carlos inició su presentación con la historia de Vim, que desciende de otros editores de UNIX como ed y ex. Argumentó que "vim no es para todos" y explicó algunos de los casos donde conviene utilizarlo: En la programación y edición de archivos de configuración principalmente. Explicó algunos de los principios en los cuales se basa el diseño de este programa y algo llamado el "zen de Vim".

"En UNIX todo es como un río, todo es como un flujo de texto", dijo Carlos. Los comandos más básicos y poderosos de UNIX trabajan únicamente con texto. "El minimalismo es bello", es preferible que el software sea feo, pero simple. "Controla tu energía". Vim se maneja exclusivamente con el teclado. Las teclas *h*, *j*, *k* y *l*, sirven de la misma forma que las flechas de dirección. Todo el teclado posee funciones programadas. El objetivo de esto es nunca requerir un ratón y no perder tiempo retirando las manos del teclado para activar una opción y luego volver a él.

Al ver los ejemplos de Carlos comprobamos que sus recomendaciones funcionan. En ningún momento requirió un *mouse*. Mediante comandos del teclado dividió la pantalla en dos y colocó código fuente de programación en ambas partes. Brincaba de una a otra y navegaba por los textos justamente como él decía: Brincando, no recorriendo. Con dos o tres teclazos llegaba al punto donde quería estar. Carlos me comentaría posteriormente: "Si ya soy rápido en el teclado, ¿para qué me voy a mover del teclado para usar el *mouse*? [...] Si es de código, comunicarme, usar la red y todo eso, rara vez uso el *mouse*. El *mouse* lo uso para

Figura 14. **:** Teclado "Happy Hacking" y la *netbook* de Carlos con un *sticker* de Google Summer of Code 2011.

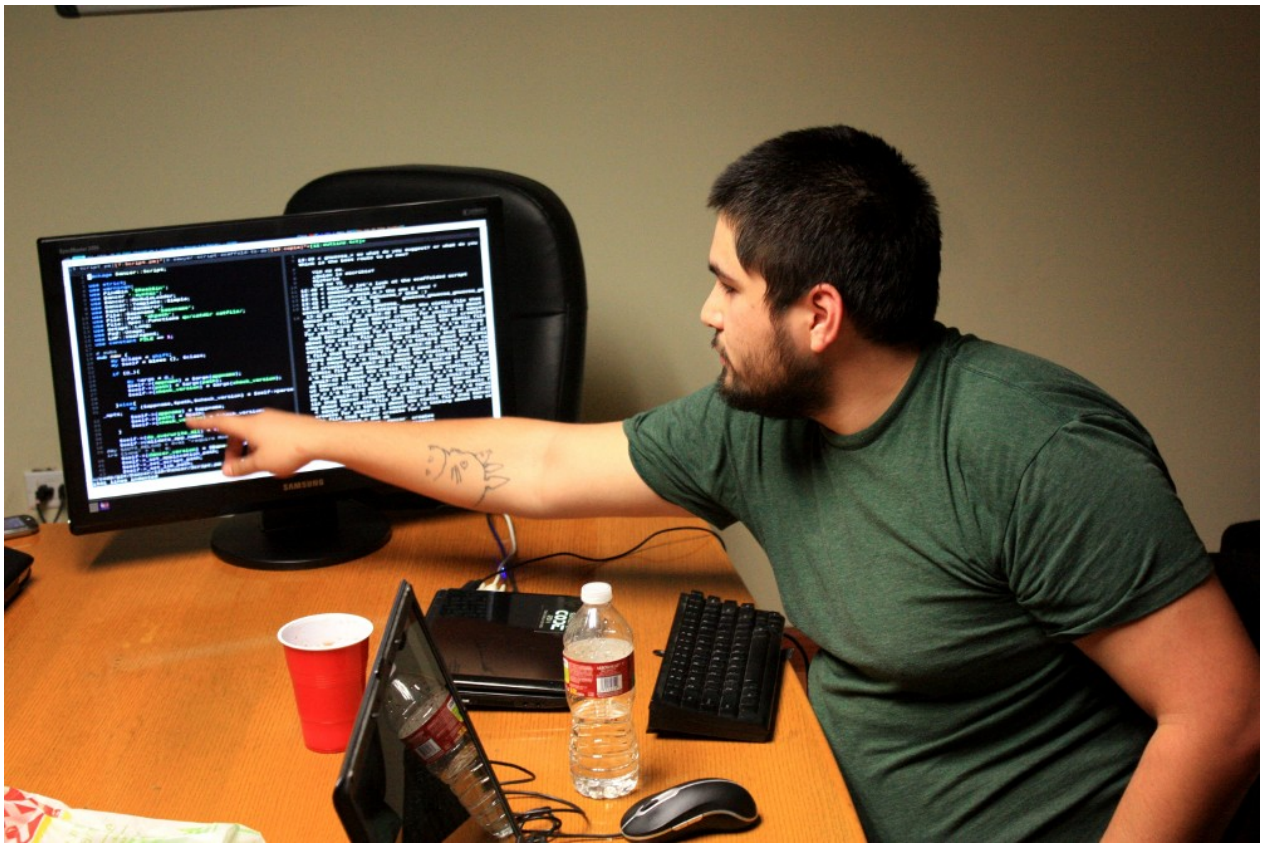


Figura 15. **:** Carlos durante el taller de Vim, explicando algo en la pantalla.

ocasiones bien específicas, que básicamente es la red".

Como ejemplo de malas prácticas de edición, nos remitió a cualquier usuario de Microsoft Word. Muchas veces, para buscar información en un documento dejan presionada la flecha hacia abajo mientras ven el texto avanzar por la pantalla. Posiblemente no lo encontraron, tienen que regresar un poco y estar probando para ver dónde se encuentra la información. Para Carlos esto es ineficiente no sólo por el tiempo invertido en ello, también porque el usuario no está haciendo que la computadora trabaje por él. Las computadoras tienen enormes capacidades de búsqueda y ordenamiento, ¿por qué no aprovecharlas en vez de perder tiempo viendo cómo pasa el texto por la pantalla?

Al ver este taller, me di cuenta de que Carlos es un *hacker* en el sentido propuesto por Eric S. Raymond. Los principios expuestos en su taller se encuentran en textos escritos por *hackers* como Richard Stallman y Eric S. Raymond, particularmente este último. Además,

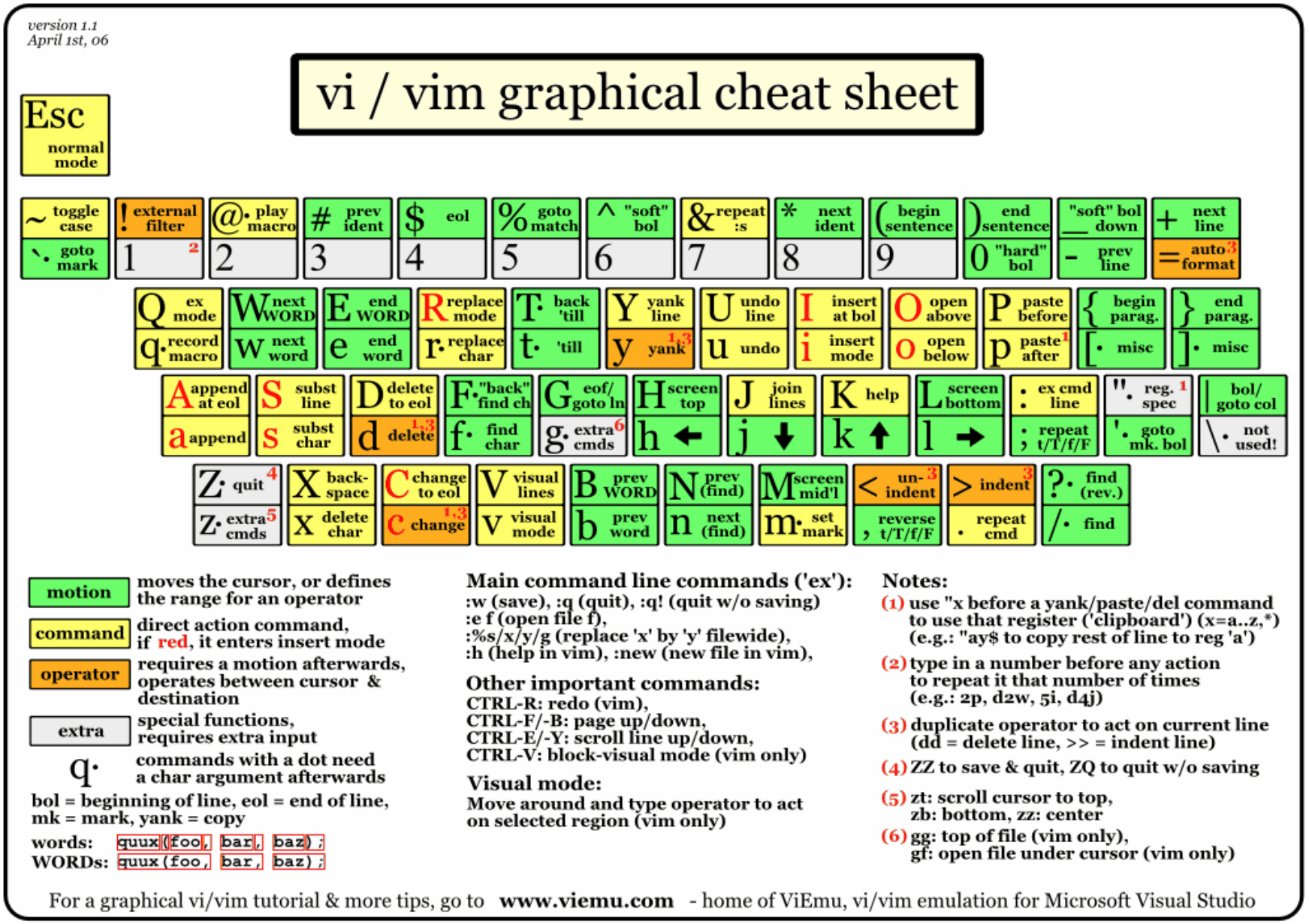


Figura 16. **:** Esta hoja de referencia muestra todas las funcionalidades que Vim asigna a cada una de las teclas. Como puede verse, tanto las letras minúsculas como las mayúsculas poseen una función. Por ejemplo, la *p* minúscula sirve para pegar información después del cursor, mientras que la *P* mayúscula sirve para pegar información antes del cursor. Un uso eficiente de Vim requiere memorizar todas estas funciones, cuando menos.

me comentó que había aportado a proyectos de software libre como Dancer, un módulo para el lenguaje de programación Perl, y para LibreOffice. Había encontrado un *hacker*, como quedará claro más adelante.

Cuando decidí entrevistar a Carlos y propuse un día para vernos, me comentó que estaba a punto de partir a San Francisco. Consiguió empleo como programador de Perl y en unos días más se iría de Mexicali. Aceptó la entrevista un día antes de marcharse, pero debido a un problema de última hora no se presentó a la cita. La solución que encontré fue seguirlo y entrevistarlo allá. La primera entrevista ocurrió en un *hackerspace* llamado Noisebridge. Este lugar se discutirá en detalle más adelante. Presento a continuación una breve semblanza de la vida y trabajo de Carlos, en base a lo que me contó.

Biografía. Carlos nació en Mexicali, Baja California. Su papá es ingeniero electrónico, por lo que estuvo en contacto con las computadoras desde una edad muy temprana. Recuerda cómo aprendió de niño a operar una computadora Commodore 64 y que entendió la programación. "Agarré la idea desde chamaco, pues, de cómo se puede controlar una computadora. La idea nomás, no muy a fondo."

Durante quinto año de primaria se mudó con su familia a Torrance, California, una ciudad cercana a Los Ángeles. Ahí su papá compró una computadora Packard-Bell y contrató el servicio de Internet con America On-Line.[23] "Tonces empecé a leer como lo de los emuladores[24] y los juegos, y como todo morrillo empecé por los juegos... Pues de mi generación".

Su interés por la tecnología se alimentó por la curiosidad de saber cómo funcionaba: "Mi papá recibía fax. [...] Tonces a mí siempre se me hizo bien increíble cómo sabía cuándo iba a llegar un fax. [...] Por que el fax, pues, ¡nomás salía, güey! Nomás salía y ya. Nunca te avisaba ni nada". Cuando le preguntó a su papá cómo sabía que llegaría, respondió que era

[23] America On-Line es un servicio de Internet pionero que ofrecía servicios de fácil acceso a usuarios domésticos. Una de las ideas principales eran los servicios exclusivos, que limitaban las opciones de los usuarios pero facilitaban el uso (Zittrain, 2008). Por ello, y otras prácticas de marketing, el servicio no es bien visto dentro de la comunidad *hacker*.

[24] Los emuladores son programas que intentan replicar el funcionamiento de computadoras. En el caso de los juegos, existen emuladores de NES (Nintendo Entertainment System) que tienen la capacidad de ejecutar juegos de Nintendo en una computadora de escritorio.

Figura 17. **:** La computadora Commodore 64 se conectaba a una televisión para mostrar la salida, de esta manera los usuarios se ahorraban el costo de comprar un monitor de computadora. Representó la puerta de entrada al mundo de la informática casera para muchos usuarios. (Polgár, 2008) La posición de la tecla "control," se encuentra en el lugar donde los teclados actuales tienen la tecla "tab". Esto me parece similar a la posición de su actual teclado Happy Hacking.

gracias al correo electrónico, recibía un mensaje que le avisaba que un fax estaba por llegar.

"Empecé a programar cuando mi papá una vez llevó una computadora de UNIX. [...] Y en realidad era una versión de Linux, ponle, el 2.4, pero de los más viejos". Pero cuando manejar esta máquina con Linux también buscó juegos. Carlos sintió afinidad por esa manera de trabajar: "Yo tenía Windows y ponle que podía jugar y todo eso, pero se me hacía bien limitado, pues. [...] O sea, como dije, esto ya lo había visto pero esto es como el futuro".

La primera distribución de Linux que instaló fue Red Hat. Ésta gozaba de mucha popularidad durante los años noventa: "De Red Hat aprendí pues que no hay gran diferencia entre Linux y Windows. Es un sistema operativo, lo instalas y ya". Pensó que la situación

se repetiría en otras distribuciones, hasta que encontró Debian, una distribución que le parece más "elitista" por que no permitía código privativo. Esto complicaba el uso a usuarios convencionales, quienes instalaban el sistema y encontraban que su monitor no funcionaba bien, o que los altavoces no reproducían sonido. Esto debido a la falta de controladores libres, muchos fabricantes de hardware no estaban interesados en crearlos.

Pero valía la pena enfrentarse a estas dificultades ya que la alternativa era usar Windows: "Todos saben que Windows, o sea, es una gran batalla para trabajar. Pero la gente trabajaba con Windows. Tonces yo desde chamaco dije: 'No, esto no puede ser'. Empecé a buscar, a buscar, y empecé a programar en Visual Basic[25] en Windows, empecé a tratar de hacer lo que debí haber hecho en el comienzo con Linux, pues". Una de las dificultades de utilizar Debian era la poca documentación que existía en ese momento:

> Y pues en la máquina Debian lo instalé pero no sabía cómo correr Internet, no sabía cómo meterle lo del ethernet[26] y todo eso. [...] Básicamente [la documentación] era como: "¿Sabes soldar? Ok, solda todo esto." [...] Nomás te daban la herramienta y te decían cómo llegar ahí, pero no te decían qué tenías que ver o qué tenías que modificar para llegar ahí. [...] La documentación era bien orientada, pues, a gente que ya sabía de UNIX.

Con Debian instalado, Carlos aprendió a escribir scripts en bash y otras habilidades útiles en sistemas UNIX.[27] En ese momento Carlos se interesó por la cultura *hacker*. "Tuve la suerte de no quererme adentrar en la porquería que es ser un *cracker*". Para Carlos hay una clara diferencia entre un *hacker*, es decir, un programador, y un *cracker*, un explorador ilegal de sistemas.

A pesar de estar en desacuerdo con esas prácticas, concede que pueden comprobar la

[25] Visual Basic era un lenguaje de programación desarrollado por Microsoft que se enfatizaba en la facilidad de uso. Carlos se expresa negativamente de él posteriormente.

[26] Ethernet es un estándar para la conexión de dispositivos en red. Carlos aquí se refiere principalmente a configurar la tarjeta de red de la computadora.

[27] Los scripts de bash son pequeños programas para automatizar tareas rutinarias en Linux.

seguridad de un sistema para mejorarla: "Era una porquería... O sea, si lo usas para *trollear*[28] no sirve. Pero si lo usas para estar constantemente mejorando algo, sí sirve. Siempre es bueno quebrar algo y volverlo a armar y así vas aprendiendo". Se opuso a todas las técnicas del *cracking* cuando conoció personas que las aplicaban. Incluso él mismo incursionó en estos territorios:

> Yo conocí gente que se metía al sistema Banamex, o sea, conocía gente, pues que se metió a bancos. Estamos hablando antes del 2000. [...] Me metí a los sistemas de Telnor[29], eh... Llegué a hacer llamadas gratis, o sea... Aprendí ciertos comandos hexadecimales que puedes mandar por *command prompt*[30]. ¿No te acuerdas del NetBIOS? Ah, pues, pues podías tronar abanicos.[31]

Carlos no ahondó mucho en el tema a pesar de mi insistencia. Sin embargo, explicó el por qué estos ataques eran posibles: "El *kernel* de Windows como es monolítico, o sea, básicamente todo está pegado. Podías tronarlo. Ya no puedes hacer eso". Abandonó el interés por el *cracking* cuando descubrió que le gustaba más programar: "Me aburrió el hecho de tener que usar otras cosas de otras personas y yo no aprender nada". Volcó su interés a la programación.

Llegó a la página de Eric S. Raymond y leyó con interés su texto titulado *How to become a hacker*[32] y posteriormente *La catedral y el bazaar*[33]. "Ajá, a la hora de terminar leer qué era realmente un *hacker* dije: ah, pues, está bien, no tiene nada de increíble, pues. Todavía yo creía que los *crackers* estaban curada". Además, llegó a la conclusión de que "gran porcentaje

[28] Molestar a otros usuarios en Internet.

[29] Teléfonos del noroeste fue durante muchos años el único proveedor de servicio telefónico residencial de Baja California y parte del noroeste de Sonora. El propietario es el multimillonario Carlos Slim.

[30] Command prompt se refiere a la interfase de modo texto que poseen sistemas como MS-DOS.

[31] NetBIOS significa Network Basic Input/Output System, era un sistema de red que era comúnmente utilizado por máquinas con Windows y que poseía muchas fallas de seguridad. Entre ellas, el que atacantes externos enviaran información y alteraran el funcionamiento de sistema casi sin esfuerzo.

[32] Este ensayo presenta algunos principios que guían las actitudes y las habilidades de *hackers* de sombrero blanco. Aunque el espacio aquí es muy reducido para desarrollar extensamente los puntos del ensayo, a grandes rasgos Raymond recomienda mantenerse curioso y abierto a nuevas ideas, aprender cómo programar y usar sistemas operativos libres y escribir software libre o conocimiento asociado con éste. El texto completo puede encontrarse en la siguiente dirección URL: `http://www.pavietnam.net/`

[33] Este texto puede encontrarse íntegro en la página de Eric. S Raymond que se encuentra en la siguiente dirección: `http://catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/`

es como de ingeniería social. [...] Así es como truenas cualquier cosa, pues. Buscas fallas que tenga la persona y casi siempre esas fallas se reflejan en sus programas o en su código".

A inicios de la preparatoria, a los 16 años, desarrolló habilidades serias de programación. Primero empaquetó software, que consiste en preparar el código fuente para instalarlo en diferentes sistemas. Después de sus exploraciones en Visual Basic migró a tecnologías libres "por el hecho de poder compartir herramientas". Para Carlos, la principal ventaja del software libre es la posibilidad de ver y comparar el trabajo de los demás sin costo alguno: "No tenía mucho dinero y [...] aprendes mucho viendo el código de los demás. O viendo sus ejemplos o cómo hizo tal cosa".

Su primer trabajo relacionado con la informática fue en un café internet que necesitaba un servidor. "Básicamente era administrar la red y que no estuvieran descargando pornografía y todo eso, pues. Básicamente ahí aprendí lo del Squid, del IP Tables[34], todo lo de red, pues, todo lo Linux de red". Consiguió el empleo gracias a sus amistades.

> Un amigo le dijo a su papá, de: "Este güey usa esto" y, o sea, ya me habló y yo fui. Y básicamente fue como en el rancho, pues: "¿Tú sabes hacer esto?", "Sí", "Ok, te voy a dar el dinero y has esto". Pero nunca fue como que realmente técnico en el de: "Ah, ¿cuántos años tienes [trabajando en esto]?" Simplemente fue: "Hazlo correr y si la chiva saca leche, pues dale". [...] Sí sabía cómo es una red, sé lo de TCP/IP[35], sé como todo eso, pero no sé como correr un servidor, como todo... Ahí fue cuando lo aprendí todo. Y de hecho el servidor lo tuve como una semana antes y se lo cobré como si fuera un mes, algo así, pues. Pero como yo me di cuenta que el señor no tenía ni puta idea, pues.

El usar software libre fue una experiencia muy positiva para Carlos, sobre todo cuando avisó al dueño del cibercafé que no necesitaría pagar licencias de ningún software, nunca. Ni siquiera

[34] Ambas son herramientas libres, Squid para la creación de proxies e IP Tables es un firewall. Ambos son sistemas intermediarios entre redes locales e Internet que sirven para controlar el tráfico que entra y sale.

[35] Tranfer Control Protocol/Internet Protocol. Es un protocolo de comunicación de red en el cual se basa la comunicación por Internet. Su desarrollo se dio durante los años setenta. (Rosenzweig, 1998; Zittrain, 2008)

por futuras actualizaciones.

> Ahí fue mi primera experiencia de software libre, venga, pues. [...] Tonces el señor desde ahí empezó... Aprendió, pues, que utilizar el software libre y el *open source* te abre las puertas a que la gente crezca y se pueda ir, pero tu puedas conseguir otro chango. [...] Una de las cosas buenas del software libre o del *open source*, lo que me gusta, pues, que puedes iniciar a cualquier cosa y cualquier persona puede tener un trabajo en específico.

Con el software libre no sólo sacó adelante un trabajo, también obtuvo conocimiento aplicable a otros problemas sin atarse a un proveedor. Además, dejó el campo libre a otra persona que lo reemplazaría. Para Carlos, este conocimiento y manera de trabajar es más útil que lo que se enseña en las facultades de ingeniería:

> Lo que te enseñaron en la universidad, lo de ciencias computacionales y todo eso, es cómo pensar, pues. O sea, cómo usar algoritmos, cómo trabajar... Y luego ya tu utilizas lo práctico, los lenguajes de programación, las herramientas que tienes de software y las pasas ahí. O sea, los grandes genios no fueron a la universidad, pues, ¿por qué? Porque, pues, lo aprendieron con documentación, lo que quieras.

A la par que resolvía el problema que le propusieron en el cibercafé, aprendía las tecnologías necesarias para implementar una solución. Aclaró a su jefe que lo resolvería con tecnologías libres y su jefe aceptó. "La verdad a esa edad, y creo que todavía hasta la universidad, yo miraba los trabajos como aprendizaje". El dinero también era motivador, pero gran parte de ese dinero lo invirtió en libros y computadoras que le pudieran aportar más conocimiento.

> Con ese dinero realmente lo único que hacía era... O sea, no me voy a comparar con Kevin Mitnick, pues, pero igual que todos los *hackers*, conseguían dinero y compraban algo para aprender de eso. [...] Sí, por que yo me acuerdo que compré un libro de C, compré un libro de Java y los... todos los perdí en ese café cuando se incendió, de hecho.

Mientras tanto, estudiaba en la preparatoria Xochicalco, la cual tenía una capacitación de informática y de que expresó no tener muy buenos recuerdos:

> Antes no nos dejaban llevar laptop, acuérdate, era distracción. [...] Pinchi sistema educativo es una estupidez, bueno. No nos dejaban eso... Nomás te dejaban cuando ibas a hacer presentaciones lo cual es una pinchi denigración bien cabrona a las computadoras. Es como: "Nomás para eso sirven". O sea, y es cuando te preguntas: "Bueno, tú eres una persona educada y todo eso, ¿tú crees que estás capacitada para enseñarme?" O sea, ¿de qué ching...? O sea, a mí todavía no me entra en la cabeza, pues, cómo esas personas que las consideraban adultas y cómo los consideraban personas con pensamiento didáctico.

Aprendieron programación en BASIC[36]. Durante una de las primeras clases, los estudiantes admitieron no saber nada de programación. En cada lección, la profesora escribía los ejercicios en el pizarrón y el que terminara salía de clase. Las primeras cuatro clases, todos terminaron en menos de diez minutos. "Porque, o sea, la maestra bien buena onda de... nos decía, acá pues: 'Terminen el programa' y ya se podían ir, pues. [...] Y si, realmente nomás yo y otros cuatro güeyes que ahorita están en ingeniería [...] sabían hacer los programas, pues. Pero todos los demás batallaban".

Carlos guardaba los ejercicios resueltos en la red y decía a sus compañeros: "Enséñenle esto a la maestra". La profesora pronto detectó la artimaña, al darse cuenta de que todos terminaban rápido y con soluciones idénticas. Prefirió exentarlo a tenerlo dentro del salón de clase, con la mente demasiado desocupada y acceso a las computadoras del laboratorio.

Después de su trabajo en el cibercafé, Carlos trabajó en el CEART Mexicali. Como la institución apenas iniciaba en la ciudad, el trabajo fue más de monitoreo e instalación de equipo, aunque de nueva cuenta fue una experiencia de aprendizaje:

> Pero realmente ahí más que nada levanté servidores, vi como estaba la red, puse

[36] Beginner's All-purpose Symbolic Instruction Code (BASIC) es un lenguaje de programación de alto nivel dirigido a principiantes.

> impresoras, pues realmente, ahí lo que si realmente hice fue monitorear. Pero como el CEART apenas estaba empezando, no había mucho qué monitorear, pues, o sea. Estaba monitoreando y tenía los *logs*[37], pero ahí fue cuando empecé a practicar todo eso de la seguridad y demás.

Poco tiempo después surgió la oportunidad de estudiar ingeniería en Ensenada gracias a que su papá conocía a ingenieros de la facultad. Él lo motivó a estudiar una carrera para obtener un mejor empleo, pero a Carlos no le interesaba mucho entrar a la universidad.

> El punto es que mi ruco, pues, se empezó a piratear y: "No, si, que vas a perder tiempo". Y yo pa' no hacerla de tos y por vivírmela concha dije: "Pues me van a dar de comer y de todas formas voy a ser feliz, pues... Pues sí, vamos a intentar ingeniería." Pero pues, fue... creo que es el peor error de mi puta vida, pues, haber gastado mi tiempo, ¿qué fue? Un semestre nomás, en la facultad de ingeniería.

Las opiniones de Carlos sobre la facultad no son nada positivas. Dentro de las aulas no encontró lo que buscaba y se decepcionó decepcionado con algunos de los manejos administrativos y académicos: "Es pésimo. Es un pinchi nido de corrupción. [...] Realmente te enseñan a ser ingeniero en cuanto a la actitud y lo que es un pinchi ingeniero, no la profesión, si no como persona o en la sociedad."

Yo había escuchado buenos comentarios de la ingeniería de la UABC en Ensenada, así que pregunté entonces qué esperaba de una escuela de ingeniería. ¿Qué es para Carlos un ingeniero ideal?

> Alguien que sabe aplicar muy bien lo que sabe hacer [...]. De hecho es una persona que trabaja más que una persona que se dedica al software o a la computación, pues. O sea, un ingeniero básicamente, en México [...] está manejado como un güey que es L.A.E.[38], o sea, que es administrador, que estudió administración,

[37] Los "logs" son las bitácoras donde se registran los eventos más recientes del sistema. Estos normalmente son útiles para el administrador en caso de una falla en el sistema o una intrusión no autorizada.

[38] Licenciado en administración de empresas.

> y que sabe de tecnología. Es lo único. O sea, un ingeniero en México ya no es alguien que se dedica a desarrollar, pues, o a investigar, pues. [...] Básicamente, así como en el Big Bang Theory[39], los ingenieros son los Oompa Loopmas[40] de la ciencia, pues. Y es la realidad, pues. Y está bien, porque debe haber ese tipo de gente. [...] No que te enseñen a ser un pinchi güey que maneja maquilas, pues. Esa es una putada, o sea...

No toda su experiencia fue negativa. Admite que aprendió estructura de datos, refinó sus habilidades de programación en C y practicó Solaris[41] en el laboratorio de UNIX. En éste último trabajó monitoreando y administrando sistemas, de manera similar al café internet. Sin embargo, se enfrentó con el administrador oficial de los laboratorios.

> Es un típico güey que va a la universidad y que se mantiene actualizado pero que paga un chingo de dinero para ir a conferencias... Que no tiene nada de malo, que pague... Que aprenden una que otra cosita y se cree capacitado, pues. Y sigue poniendo o utilizando pinches tecnologías bien viejas. Hay un güey que estaba usando IPWall y yo como de: "No me chingues, cabrón, ya está el IP Tables". Pero no les puedes respingar por que tienen su pinchi título pendejo y pues de dónde venimos nosotros, tú y yo[42], el pinchi título remarca más de lo que haga la gente, pues. Práctica totalmente estúpida.

Las personas de las que más aprendió fueron físicos. Yo había escuchado de esta asociación entre la física y las ciencias computacionales pero no la entendía. Tim Berners-Lee, por ejemplo, inventor de la World Wide Web, es físico, y el tema surgió en las entrevistas con todos los informantes. Así que pregunté a Carlos al respecto:

---

[39] Se refiere a la serie de televisión cómica "The Big Bang Theory" sobre un grupo de físicos.

[40] Los Oompa loompas son los ayudantes del fabricante de dulces Willy Wonka en la película "Willy Wonka and the chocolate factory."

[41] Solaris es un sistema operativo basado en UNIX desarrollado por Sun Microsystems y actualmente por Oracle.

[42] Esta entrevista ocurrió en San Francisco, lugar de residencia actual de Carlos, y varias veces dijo "de donde venimos" para referirse a México en general.

> Los mejores programadores son físicos. [...] Porque la física te enseña cómo funciona todo. Y por todo es todo, o sea, todo se rige por las fuerzas electromagnéticas y gravedad, o sea. Física tiene muchas matemáticas, es mucho resolver cosas. [...] Y es a lo que voy, pues, o sea, a la hora, al final, no es qué tanto sabes del lenguaje de programación, sino qué tanto puedes resolver un problema.

Según Carlos, obligaron a los físicos de Ensenada a usar UNIX ya que usar Windows para aplicaciones en física "es una estupidez". También me comentó que en la sierra de San Pedro Mártir, en un importante observatorio astronómico, le tocó ver cómo usaban Solaris. "Pero obviamente, estoy segurísimo que ya brincaron a Linux, eso te lo puedo asegurar". El por qué de esta afirmación tiene que ver con un suceso sobre el cual escuché mucho en el ámbito de los programadores y gente de software libre: La compra de Sun Microsystems por parte de Oracle Inc.

> Pero ese es el problema con Solaris, pues, que desde que lo compró Oracle ya no es lo mismo. Nada de los productos de Sun desde que Oracle compró a Sun, ya no es lo mismo. [...] Sun estaba lleno de *hackers* y estaba lleno de gente de la comunidad, estaba lleno de gente que le gustaba. Oracle es *suits*, Oracle es Novell. O sea, si tienes gente que es así [*hackers*], pero está manejada, pues.

Durante su estancia en Ensenada tuvo la oportunidad de explorar la biblioteca de la UABC, la cual se encontraba muy actualizada desde su punto de vista: "Tenía un chingo de libros que mandaba la SDSU, la UCSD".[43] Leyó libros y manuales sobre tecnologías que le interesaban: "Ahí fue donde realmente aprendí, pues, lo que era como ser autodidacta. Aunque suena remamón y siempre lo digo así, pues. Si, o sea, aprendí que realmente uno puede aprender por sí mismo. Y empecé a agarrar libros, y práctica, libros, ejemplos, libros y así".

[43]Se refiere a dos universidades: San Diego State University y University of California, San Diego.

Un día en el laboratorio de Solaris vio que un físico utilizaba un software bastante inusual. En la pantalla se movían grandes cantidades de texto y el físico parecía brincar en los archivos. Imagino que su impresión fue similar a la que yo obtuve la primera vez que vi a Carlos usar vim.

> Y le dije: "Oyes, ¿qué estás usando?" Bien me acuerdo: "Estoy usando vim". Me vio con cara de loco. Y yo: "Ah, Ok". Pero nunca se me va olvidar, pues, porque nunca nadie cuando yo preguntaba algo, pues, nunca nadie en mi vida, de hecho, me había hecho una cara de loco como me hizo este güey, ¿qué pedo, pues? [...] Y yo así de: "Ay, güey. Algo no ando haciendo bien", me quedé.

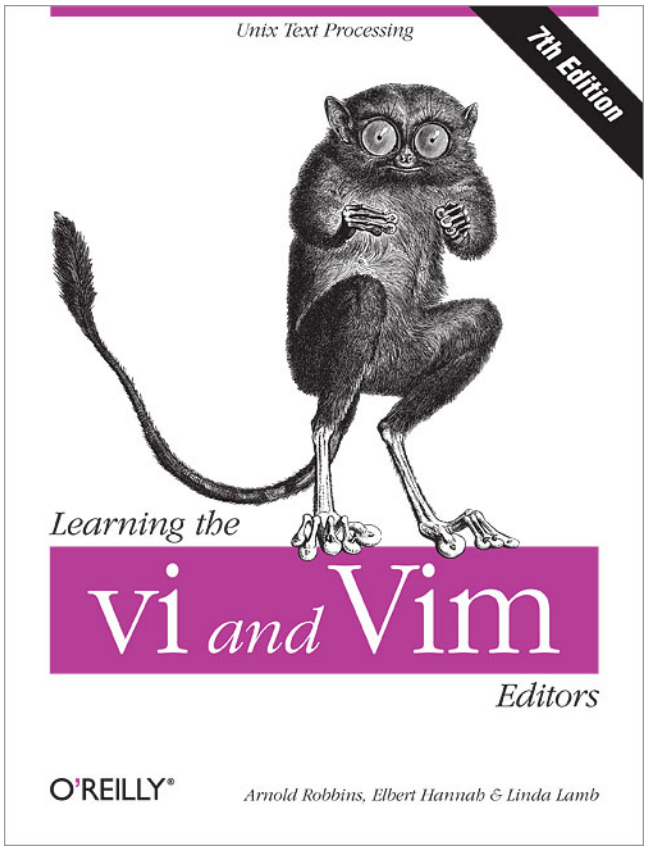


Figura 18. **:** Portada del libro "Learning the vi and vim editors" de la prestigiada editorial O'Reilly.

Esto lo motivó a investigar al respecto y en la biblioteca del CICESE se encontró con el libro "Vi and vim editors" de la editorial O'Reilly en una edición nueva. Esta biblioteca, según Carlos, se encuentra sumamente actualizada.

> Te digo, me topo el libro y lo empiezo a leer bien y me doy cuenta de un chorro de cosas así, o sea, vilmente me brillaban los ojos así de: "Ay, güey". Y de hecho llegó al punto donde me quería robar el libro, o sea. [Ríe]. Pero pues no, preferí comprarme mi copia.

En Ensenada también contactó al grupo ELUG, que significa "Ensenada Linux User Group". Se trataba, como su nombre lo indica, de un grupo de usuarios de Linux que se reunían regularmente para ayudarse o trabajar en proyectos conjuntos. Carlos no se impresionó mucho con las habilidades de algunos de los asistentes a quienes considera "desarrolladores de software", entre comillas. Esto debido a que el rango de problemas que pueden resolver el muy limitado: bases de datos sencillas manejadas mediante botones e interfases gráficas. Programadores similares al que mi mamá quería contratar para desarrollar su programa de recetas de cocina. El cliente pide lo que quiere ver en pantalla:

> Si es un desarrollador, pues, pero prefiere que especifique un *UI developer*[44] porque, o sea, conlleva algo de diseño y saber cómo la gente maneja las cosas, no conlleva hacer algo que funcione, como un *software developer* [...]. Eran honestos que: 'Tengo una familia, tengo esto, tengo lo otro y lo que sé lo he aprendido pero ya de ahí no le muevo porque funciona.' O sea, le sacan leche a la cabra.

Después de abandonar la universidad encontró trabajo en Torrance, California. En esta ocasión trabajó en un proyecto que lo emocionaba: Un *startup*. Una de tantas empresas tecnológicas californianas recién inauguradas, buscando éxitos como los de Microsoft, Apple o Google. Su nombre era Blinker, que desarrollaba un sistema de geolocalización de taxis utilizando la placa Arduino.[45]

Blinker buscaba que los usuarios pidieran taxis a través de sus teléfonos celulares. En un mapa en la pantalla del teléfono se mostrarían los taxis más cercanos. El usuario seleccionaría uno y éste llegaría pronto. Es obvio que esto requiere un sistema de geolocalización, pero implementarlo con GPS (Global Positioning System) era muy caro en ese tiempo. Crearon un sistema con la misma funcionalidad de GPS, pero con la placa Arduino como base, la cual es más barata. Carlos se encargó particularmente de escribir código en Perl para Arduino.

---

[44] User interface developer. El encargado de crear las interfases gráficas mediante las cuales interactúa el usuario para trabajar.

[45] Sobre esta placa electrónica se discutirá en detalle más adelante.

Después de seis meses la nueva empresa parecía flaquear. Carlos notó que mucha gente renunciaba y un día intentaron comprar sus acciones de vuelta. "Ya cuando te quieren comprar tus acciones es que algo anda bien mal, pues". Ante la incertidumbre decidió abandonar el barco: Vender sus acciones y buscar otro trabajo. Atribuye el fracaso de la empresa al mal momento para implementar esa tecnología. Actualmente existen sistemas parecidos en diferentes partes del mundo, pero hoy la velocidad de Internet celular es mucho más alta y más personas tienen acceso. Es un momento más propicio para que esas tecnologías se popularicen.

De hecho, muchos de los taxis de San Francisco poseen un sistema similar, que surgió gracias al ejemplo de "Blinker", según Carlos. Además, los taxis tienen ya terminal de tarjeta de crédito. Ya que ahora la gente usa más su tarjeta de crédito "pues por que es esa pinche confianza en el *Big Brother*". El fundador vendió la empresa y se mudó a San Francisco: "O sea, se vino para acá y se metió con otra compañía... Y todavía sigue en esa compañía, de hecho, de fundador pasó a ser pinchi larva. [...] Ya nomás es un trabajador, pues. Pero, ¿por qué? Porque se le fue todo el dinero en esa idea".

El siguiente movimiento de Carlos fue mudarse a Phoenix, Arizona, para trabajar en otro *startup*. En esta ocasión era una red social para adultos mayores de nombre "Senior heaven". Las cosas tampoco parecían muy promisorias, sobre todo cuando por las mismas fechas surgieron muchísimos sitios parecidos. Después de seis meses de trabajar ahí, también abandonó la empresa.

Volvió a Mexicali y entró a trabajar en "Conveyor group" del otro lado de la frontera, en Imperial. En esta empresa de *marketing* se dedicó a graficar el rendimiento de sistemas con GNU Plot para generar reportes en LaTeX. "Yo lo que hacía era como un *fifty-fifty*. Es la primera vez que trabajo como mexicano en el gringo, en el sentido de que era desarrollador de software pero también era administrador de sistemas. Y es el único lugar en todo el sur de California que hospeda todos sus servidores y tiene su *data center* en la misma oficina". Sobre esto me aclaró que en Estados Unidos, a diferencia de en México, existen roles laborales muy

delimitados, en cuanto a informática se refiere. En México eres "poquito *sysadmin*, poquito desarrollador de base de datos y desarrollador de software, ¿sabes cómo?"

Durante este periodo también se inscribió en al Imperial Valley College (IVC) y tomó algunas clases de ciencias debido a su interés por estudiar física. Es en este momento cuando sale de la casa de sus papás y se independiza en Mexicali. También, gracias a estudiar en IVC tuvo la oportunidad de participar en Google Summer of Code (GSoC), un programa de tutoría para estudiantes de programación. Éstos presentan un proyecto de aporte al software libre y se les asigna un mentor que supervisa y guía el proyecto.

Carlos entró como representante de la Perl Foundation, una organización conformada "por la comunidad de Perl, los usuarios, los desarrolladores, la gente que trabaja con Perl. [...] Es como para hacer buenas costumbres, pues. O sea, a la hora de escribir Perl o para dar especificaciones, pues. [...] Una buena costumbre es como hacer algo bien. O sea, algo... algo que está especificado, pues. Que se ha demostrado muchas veces que te ahorra mucho tiempo es una buena costumbre." Su proyecto consistía en un módulo para actualizar automáticamente el script Dancer para Perl.[46]

Software y opiniones. A lo largo de las entrevistas algo fue muy evidente para mí: Carlos tiene opiniones muy fuertes y específicas acerca de las tecnologías que usa, comenzando por los lenguajes de programación. Por supuesto, he conocido a programadores que también manejan muchos lenguajes, pero la mayoría no tiene una postura tan personal y tan fuerte hacia ellos. En ocasiones pueden comentar sobre la dificultad o facilidad que representa escribir código en cierto lenguaje, o los beneficios que esto les trae laboralmente. ¿Qué piden más los clientes? ¿Cuál tiene más posibilidades en el mercado? Incluso puede ser una elección más básica: ¿Cuál lenguaje me enseñaron en la escuela? Así que pregunté cómo había obtenido todas esas opiniones:

Pues yo digo que todo eso se aprende. Como todo... Como hay un libro que se

[46] La descripción del proyecto, el cronograma y otros datos pueden encontrarse en una de las páginas oficiales de GSoC: `http://www.google-melange.com/gsoc/proposal/review/google/gsoc2011/gnusosa/1`

> llama "Aprende a programar en 10 años", pues. Y es la realidad, pues, o sea, un programador no es un güey que tiene un título, pues, o sea. Un programador es alguien que lo hace y que empieza a aprender y que puede tomar cualquier lenguaje de programación basado en la experiencia anterior, pues. Y yo digo que aprendí a programar C por que todas las bases de UNIX son en C.

En un momento se refirió a los "lenguajes de programación horribles" como Fortran y Pascal: "Empecé a programar en Perl, y luego empecé a programar en C++, que es más orientado a objetos y luego programé en Java. Y fue cuando me quedé: Bueno, esos tres lenguajes modernos no pueden contra eso que toda la gente sabe." Es decir, esos tres lenguajes buenos y actuales no han podido erradicar esos "errores del pasado." "El único que, que considero de esos antiguos son C y Lisp. Son los únicos que siguen siendo así gigantes."

Advierte: "Yo nunca te voy a criticar algo que yo no conozco." Pero sin duda tiene mucho de qué hablar: "Ruby y Python. Bueno, con Python si he trabajado bastante, pero lo poquito que he trabajado de... O, ponle, a nivel medio de Ruby [...] ha sido increíble. Entonces no tengo nada por qué criticarlo. Cuando llegué a trabajar con .NET[47] era una pinchi basura. Y pasaron los años y mejoró bien cabrón." Esta admisión es bastante significativa proviniendo de alguien tan de acuerdo con el software libre y *open source*.

Carlos no sólo requiere de herramientas para hablar el lenguaje de la máquina, también requiere un sistema operativo para que su máquina funcione. Ya lo había escuchado hablar de Debian en las reuniones de mxlOpenSource, pero durante una de las entrevistas le pregunté por qué lo prefería.

> Realmente mi *fetish*, como le llaman, es Debian. Debian yo lo he usado y es con el que he aprendido y todo lo que tú quieras, pues. De Linux ha sido con Debian. Pero a la vez, no quiero no darle crédito a Gentoo, porque con Gentoo yo aprendí cómo es el *core* de Linux, o sea. Cómo se maneja Linux, cómo compilar

[47].NET es un lenguaje de programación de Microsoft que viene inspirado en tecnologías como Visual Basic y Visual C++.

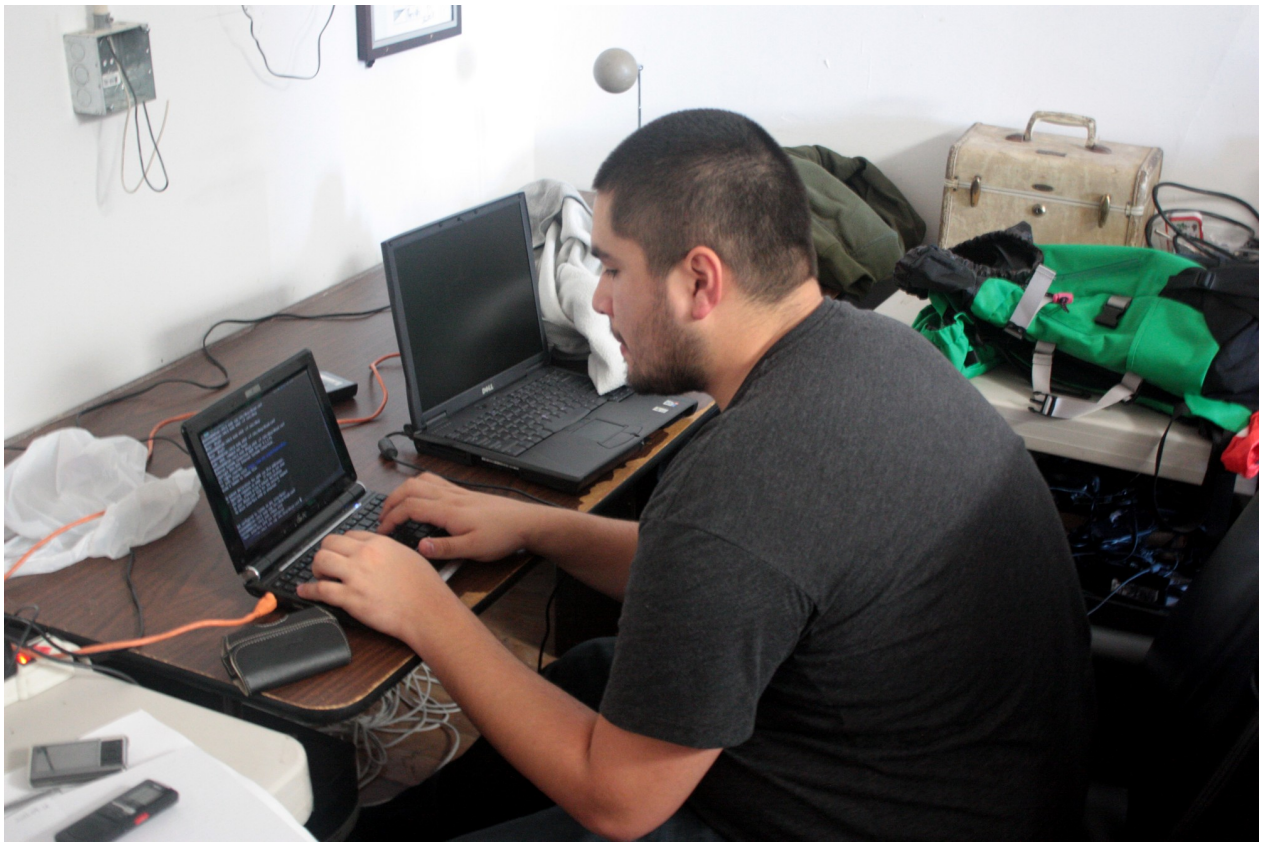


Figura 19. **:** Durante la primera entrevista en Noisebridge, dentro del salón Turing, Carlos intentó instalar Debian a una laptop obsoleta (al fondo). Estuvo intentándolo desde aproximadamente las 4 pm hasta las 8, cuando yo me fui. El objetivo era dejarla funcionando para que otras personas pudieran utilizarla. Debido a que la laptop no iniciaba mediante la red, Carlos se auxilió con su *netbook* para encontrar el problema.

> Linux, cómo hacer tu distribución Linux, cómo ir haciendo tu catedral, como dice Raymond, con Gentoo. Que Gentoo vilmente es nomás, pum, ahí están las aplicaciones, güey, chíngale.

Según lo que Carlos me cuenta, a diferencia de Debian, con el cual se puede trabajar inmediatamente después de instalarse, en Gentoo hay que instalar y configurar todo "a mano". Ir instalando cada una de las partes del sistema operativo sin mucha ayuda. Esto complica el trabajo a un usuario novato, pero ayuda a un experto, ya que se involucra directamente con todas y cada una de las partes de lo componen.

> Es que todo depende, toda distribución tiene su nicho, tiene su mejor entorno para

> lo que lo ocupas, pues, la aplicación. Entonces lo mismo funciona para lenguajes de programación. Perl es bien bueno para ciertas cosas, pero es bien malo para otras cosas. Python es bien bueno para tales cosas, y es bien malo para otras cosas. Ruby... Y es lo mismo. Como yo, yo ya estoy en ese punto donde realmente ya no es: "Debian, Debian todos los días". Pues, o sea, como ya realmente Debian llegó a un punto donde se convirtió en una religión, pues, así como Maradona. [...] Debian es la distribución más segura, nunca te lo voy a negar [...]. No significa que Debian sea la más rápida, no significa que sea el sistema operativo de GNU/Linux más pequeño, pues.

Además, en Linux hay muchas posibilidades para elegir una interfaz adecuada a las necesidades del usuario. Aunque Carlos casi no utiliza el *mouse*, posee una interfaz gráfica en su computadora. Las primeras veces que lo vi trabajar noté que no la conocía. Típicamente una computadora de escritorio con Linux tendrá instalado un ambiente gráfico sencillo y amigables. Los usuarios más avanzados podrán usar otros que requieren mayor conocimiento por parte del usuario. Carlos usa un sistema llamado xmonad, un sistema para usuarios que no quieren perder tiempo ajustando el tamaño de sus ventanas.

Le pregunté sobre las posibilidades comerciales que veía en el software libre. Esto debido a su plática en mxlOpenSource titulada "Free and Open Source software que vende", a la cual desafortunadamente no pude asistir. Me respondió que tenían muchas posibilidades comerciales. La respuesta le pareció obvia y ejemplificó con el caso de los servidores de Internet. Para él es indispensable el uso de software libre, ya que se requiere de una gran comunidad que esté constantemente revisando y corrigiendo los errores. Pronunció un apasionado discurso en donde me limité a escuchar.

> Hay un chorro de asociaciones, hay países que se han atrasado, como México, gracias a Microsoft. [...] Pues, es que mira, hay mucha gente que invirtió un chorro de dinero en certificaciones de Windows, de Microsoft, pero la realidad de Microsoft y de Windows es que todo su software es cincuenta y cincuenta,

> pues. Estás cincuenta por ciento de tu tiempo trabajando, y el otro cincuenta por ciento arreglando cosas, formateando el Windows, comprando otra máquina, o sea. Y son errores que la gente piensa que es del hardware, pues. El consumidor normal piensa: "Ok, Windows está bien, yo estoy mal". O: "Me estoy pasando de lanza", O: "Cuando estoy viendo porno todo se me llenó de virus". Que si se les llena de virus, pero la realidad es que no. Realmente es porque el pinchi Windows nomás tiene una caducidad de seis meses, luego se satura. Y porque es un kernel monolítico, pues, es un kernel que maneja todo, es una pendejada.

Durante esta entrevista estábamos en el salón Turing de Noisebridge. Dentro del salón había algunas personas trabajando, incluyendo un Argentino que daba sus primeros pasos con Linux. Se interesó por el tema y nos preguntó si podía acercarse a escuchar.

> Desde el lado corporativo, Microsoft, o sea, por el hecho de... Y volviendo a lo mismo, a las generaciones anteriores. Tú no puedes vender software libre o *open source* porque era una pinchi bola de hippies, pues, según la gente que tienen esta mentalidad de: "Yo voy a ser profesionista, y mi familia y no hay más en la vida". Ok. Entonces, toda esta generación de gente que hizo tratos con Microsoft de diez años, de quince años, de lo que quieras, ahora se están tragando sus lenguas, pues. Porque a la hora de darse cuenta que el pedo son con los servidores y UNIX, se dieron cuenta, pues, que ahora no pueden deshacer ese contrato. Y tienes dinero gastado, tienes tiempo gastado, tienes empleados gastados. Tienes dinero que pudiste haber invertido en contratar gente. Y eso es lo que Microsoft, o las grandes compañías no tienen... Bueno, Sun si lo tenía y las más grandes compañías de software libre y *open source*.

Concluyó su apasionado discurso con una reflexión acerca del software libre. Para él, el software no es sólo una herramienta, es también un producto de dinámicas y trabajo basado en principios filosóficos. No lo ve como un objeto material totalmente neutro, tiene una fuerte carga simbólica.

> El *open source* y el software libre hacen a la gente, ¿cómo se llama? Que crezca más, pues. Vas aprendiendo un chorro, te vas especializando y vas aprendiendo más cosas, pues. ¿Por qué? Porque estás compartiendo el conocimiento, porque estás compartiendo recetas, lo que quieras. El típico choro del [Richard] Stallman, que es la realidad. Entonces México está en un punto en donde están tan idiotizados con el hecho, pues de que es bonito y brilla, que lo único que vende es el .NET y Java. ¿Por qué? Porque hay más oficinas que desarrollo en México.

Con esto último Carlos se refiere a que existe más trabajo administrativo en México que creación y desarrollo de nuevas tecnologías. Evidentemente es crítico ante la situación tecnológica que vive México, incluso también en el asunto del *hacking*. En un momento de la entrevista comenté a Carlos que entrevisté a algunos *hackers* de sombrero negro y sus comentarios hacia ellos fueron muy negativos. Casi me advirtió que no perdiera mi tiempo entrevistándolos. A pesar de esto, justificó el hecho de que las intrusiones o intentos de "tronar" sistemas pueden contribuir al mejoramiento de la seguridad y estabilidad de un sistema.

> Te voy a ser honesto yo. La gente que se manejaba como *black-hat* en México, es cincuenta por ciento si y el otro cincuenta por ciento es choro. En México no ha habido *crackers* famosos, en México no ha habido esa... ¿Cómo ponértelo? No ha habido esa... Esa escena donde realmente haya, ¿cómo se llama? Retos. En México las pinchis empresas están de la mierda, pues, en cuestiones de la seguridad, la base de datos, o sea, todo. ¿Por qué crees que Banamex siempre está contratando gente? O sea, por eso mismo, pues. Porque tienes a gente que sí está capacitada, pero... Es a lo que voy, a lo que te digo, en el trabajado de desarrollador... Un trabajo Mexicano en cuanto a informática o computación y eso, es de cotorreo pues, es de cincuenta por ciento esto, treinta por ciento lo otro. No hay algo específico. Entonces, tú para ser *black-hat* en México es como: "Ay, güey, por favor. O sea, ¿qué vas a tronar?". El Politécnico y la UNAM, ellos te dicen: "Haz lo que quieras. O sea, al final un chango lo va a arreglar porque

> tenemos un chingo de changos". [...] No te estoy diciendo que no vaya a haber servidores o sistemas bien increíbles pues, pero el cincuenta por ciento es eso, es un pinchi sistema chicanada.

Carlos se explica toda esta tecnología "mal hecha" o mediocre como producto de la mentalidad mexicana. A pesar de ello, confía en que en México hay más capacidad para construir que para destruir, y citó algunos ejemplos en los cuales se basa para pensar esto.

> O sea, tenemos esta pinchi mentalidad los mexicanos pues, de: "Si truena después de mi trabajo, pues ya ni pedo, ya estoy caguameando". Es una cultura bonita, pero está bien jodida, pues. Te digo, si viajas y conoces gente y tripeas diversos estilos de vida te das cuenta de que las cosas son totalmente diferentes, pues, a la pinchi chicanada que nos venden a los mexicanos, pues. Entonces ese es el problema, pues, que yo le veo a la gente en México. No te voy a mentir, en México hay unos *white hackers* increíbles, de lo mejor. O sea, está Gunnar, está la gente de Suse, Mono salió de México, Miguel de Icaza salió de México, los que empezaron la distribución de la CERN[48] , ¿sí sabes que la CERN tiene una distribución de Linux? Bueno, la empezaron en México y eran físicos de la UNAM. Es a lo que me refiero, pues. Pero creo que somos mejor para construir que para destruir, pues. No tenemos esa mentalidad, pues, de... Y no hay esa escena de: "Ah, me voy a chingar esto", pues. Si la hay, pero casi siempre la hay en morrillos, en morrillos que nomás, porque ya se metieron a Telnor, ¡uh, ya! "Me pude meter al sistema de calificaciones de mi escuela". Realmente: "Hey, güey, mejor vete al puto gimnasio, ve y consíguete una morra, ¿yo qué sé?" O sea, no tiene mérito.

Por último, casi para concluir nuestra última entrevista, Carlos me comentó la actitud que él espera de sus iguales: La misma actitud crítica que el tiene para con todos los demás. La

[48] Gunnar Wolf es un desarrollador mexicano que trabaja en Debian. Suse es una distribución Linux. Mono es un lenguaje de programación libre compatible con .NET de Microsoft, fue iniciado por Miguel de Icaza, un mexicano que también desarrolló el sistema el ambiente gráfico Gnome para Linux. CERN son las iniciales de "Organisation Européenne pour la Recherche Nucléaire",

confrontación de ideas como dinámica para generar conocimiento y actividad productiva.

> Es que casi siempre cuando digo estas cosas todos me ven así como que: "Ah, pinchi mamón y lo que quieras". Pero no es mamón, pues es que si no le exiges, pues, ¿cómo quieres que se de la excelencia? [...] Tonces, es eso, pues, es esa crítica a la escena que hace falta, pero en el *black hat*. Porque los *white hats* aquí en México, o sea, ¿cómo ponerlo? Son críticas constructivas, pero porque se esperan, pues.

Cuando terminé mi primera entrevista con él, Carlos se quedó reparando la laptop a la cual quería instalar Debian. Al volver a Mexicali, los asistentes a mxlOpenSource lamentan haber "perdido" a un elemento que aportaba tanto conocimiento al grupo. Pero aún a distancia, Carlos se mantiene al tanto de las actividades del grupo.

Durante el tiempo que llevo de conocerlo, Carlos se ha dedicado a la programación. Al llegar a San Francisco trabajaba para una empresa, escribiendo código en Perl. Muchas veces durante mi visita mencionó que se decepcionó cansado de ese trabajo rutinario y corporativo. Me comentó que su trabajo era un "empleo Godínez".

Él había utilizado el término varias veces, así que le pregunté a qué se refería exactamente. Su respuesta fue: "Un chorro de películas y series mexicanas han tenido como que apellido del Godínez y el bato así todo de saco y bien recto, administrador, nomás vende y nomás piensa en dinero". Un empleo para gente de traje, monótono y abu-

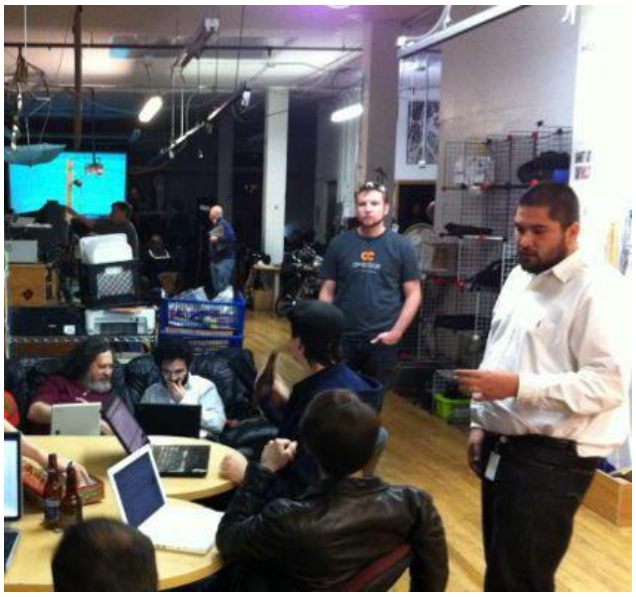


Figura 20. **:** Carlos en Noisebridge (derecha). Esto fue durante una visita de Richard Stallman en 2012. Se encuentra en el sillón de atrás, conversando con alguien mientras utiliza su netbook Yeelong Lemote.

rrido, gente de ventas. Carlos pronto cambió

de trabajo a una empresa llamada Own Point of Sale, donde se encontró más cómodo. No permaneció mucho ahí, y migró de trabajo a una empresa llamada Function X. Según Carlos, ambas son más parecidas a un *startup* que aun "empleo Godínez".

Desde que lo conozco está involucrado con la comunidad. Es visitante frecuente de Noisebridge, donde conoció a Richard Stallman cuando dio una plática durante "Represent youself", un evento abierto de la comunidad.

Determinar las postura política de los hackers *frente a la tecnología a través de sus discursos*

Desde las perspectivas teóricas sobre la tecnología se encuentra una corriente que también es hereditaria de la sociología de la ciencia. Ésta es conocida como la teoría crítica de la tecnología. Se propone pensar la tecnología como producto impregnado de ideología y creado socialmente, en oposición a la perspectiva determinista. Ésta última asume que la tecnología se encuentra en una mejora continua y que su desarrollo es lineal. (Feenberg, 2002) Uno de los principales exponentes de esta teoría es Andrew Feenberg.

El autor parte del supuesto de que "una buena sociedad debe aumentar las libertades personales de sus miembros habilitándolos a participar efectivamente en un creciente rango de actividades públicas" (Feenberg, 2002, pág. 3). Ante la perspectiva de una sociedad totalmente dominada por la tecnología, o el "regreso" a un periodo menos dependiente, Feenberg propone alternativas. Afirma que es un peligro ver a la tecnología simplemente como una "herramienta neutral" y debe tomarse en cuenta como producto del ser humano, y a la vez como instrumento de poder.

Critica incluso la visión marxista. Marx, según Feenberg, cayó en la trampa al considerar a la tecnología como "neutral" (Feenberg, 2002). Para él, la tecnología era una herramienta utilizada por el capitalismo para la explotación del proletariado (Marx y Engels, 1966). La misma tecnología utilizada con otros fines servía para dar a cada quién de acuerdo a sus necesidades. Para Feenberg es necesario darse cuenta de que dentro de la misma tecnología hay contenido político. El diseño de la tecnología no es puramente instrumental, también es una cuestión de poder.

Lizama Mendoza (2005) al estudiar los *hackers* utilizó una de las categorías desarrolladas por Feenberg para analizar las prácticas del los *hackers*. Según Lizama, la postura política de los usuarios es una actitud que muestran los usuarios ante el poder ejercido a través de artefactos tecnológicos. A continuación se listan y explican brevemente las características de cada uno de ambas actitudes:

Usuario despolitizado tecnológicamente	Usuario politizado tecnológicamente
Diseño unilateral de la tecnología: La tecnología es diseñada por las grandes empresas tecnológicas y los usuarios se limitan a utilizarla sin cuestionarla.	Cultura *do-it-yourself*: El usuario, por su propio interés, aprende sobre la tecnología y tiene la capacidad de modificarla.
Marketing como criterio de selección: Los usuarios seleccionan la tecnología en base a la publicidad y no en base a su competencia tecnológica.	Comunidad autogestionada: Los descubrimientos generados por el proceso de aprendizaje son compartidos con otros usuarios mediante una dinámica que no está condicionada por un control corporativo o centralizado.
Actualización tecnológica a partir de criterios oligopólicos: Las necesidades de actualización o migración son dictadas por la necesidad económica de la empresa en cuestión, no por las necesidades de los usuarios.	Producción de información-conocimiento: Se objetiva este aprendizaje mediante memorias, artículos o código fuente.
Usuarios tecnológicamente cautivos: La misma estructura de la tecnología provoca que el cambio de proveedor, o una migración hacia las plataformas de otra empresa sea complicado.	Politización tecnológica: Los usuarios tienen criterio para evaluar y cuestionar la tecnología.
	Reinvención creativa: Crea dinámicas tendientes a innovar el diseño y uso de la tecnología.

Lizama Mendoza propone que el *hacker* se encuentra entre el segundo tipo de usuarios y "mediante la producción de software libre y la defensa de una internet donde impere el

Figura 21. **:** Mapa de *hackerspaces* activos en noviembre de 2011.

libre acceso y distribución de la información, han generado un modelo de uso social de la tecnología [...] que impulsa directamente el desarrollo de la cultura de la información" (Lizama Mendoza, 2005, pag. 194). Esto sirve como punto de partida para analizar la actitud de los *hackers* presentados en esta tesis.

Miguel Eduardo Venegas. Conocí a Miguel Eduardo cuando entré a la página de hackerspaces.org. Un *hackerspace*, de acuerdo a este sitio, es un lugar físico, operado mediante una comunidad, donde la gente puede encontrarse y trabajar en sus propios proyectos. Escuché de estos lugares gracias a pláticas en convenciones *hacker* como DEFCON o H.O.P.E. (Hackers On Planet Earth) las cuales descargué de sus respectivas páginas oficiales.

En el sitio web se encuentra un listado de todos los *hackerspaces* activos en el mundo. También hay un mapa en el cual podemos localizarlos geográficamente. Las zonas de Estados Unidos y Europa contienen una cantidad enorme de apuntadores, al grado de que las líneas de división política son apenas visibles de tantos que hay. En el resto del mundo la historia es diferente. En África el mapa está casi en blanco, lo mismo en Rusia, donde se encuentran dos apuntadores. En México hay seis *hackerspaces* en el sitio.

Uno de ellos estaba localizado en Ensenada, Baja California, bajo el nombre de Norte-Lab. No se ofrecía mucha información, así que entré a la página web. En la página tampoco se daban muchos datos, la localización exacta coincidía con el Centro Estatal de las Artes, y se anunciaba un curso de electrónica para artistas. Aparecía también nombre del webmaster: Miguel Eduardo Venegas Monroy.

Al revisar su currículum me enteré de que es diseñador gráfico. En el sitio se ofrecía un vínculo para descargar su tesis de licenciatura de nombre: "La deficiencia de diseño gráfico de Tuquito" que intentaba criticar la interfaz gráfica de una distribución Linux desarrollada principalmente en Argentina, para sugerir cambios. Esto me llevó a sospechar que Miguel Venegas es un *hacker* de sombrero blanco.

Lo añadí a diferentes redes sociales y justo cuando estaba por enviarle un correo electrónico me lanzó a una conversación por el chat de Google. Le comenté mi interés de asistir al taller. Por cuestiones de horario y distancia no asistiría a todas las sesiones, pero las primeras dos coincidieron con el periodo vacacional de semana santa en 2011, así que podría asistir.

Me lo encontré por primera vez en persona el 18 de abril, en la primera sesión del taller. La cita fue en un almacén del CEARTE (Centro Estatal de las Artes Ensenada) utilizado para materiales y cuestiones de escenarios y organización de eventos. Le pregunté a Miguel si ése era el *hackerspace* y me respondió que no era un lugar físico si no que estaba "en la red". Eso me decepcionó un poco, ya que significaba que no había reuniones de *hackers* en el CEARTE.

La primera sesión corrió a cargo de Cynthia, una ingeniera en electrónica y se contó con la presencia de interesados de Tijuana, Mexicali y Ensenada. Cuando terminó la primera sesión, donde se presentó el curso y el temario, Eduardo explicó a las personas que se quedaron porqué eran importantes las tecnologías libres y la licencia Creative Commons.

Mencionó brevemente que él, junto con otros amigos, desarrollaron un sistema de colores similar a Pantone, de licencia comercial, pero en versión libre. Pero sus intentos no prosperaron. Su discurso era parecido al de otros *hackers* como Richard Stallman por lo que

Figura 22. **:** Miguel Eduardo explicando los beneficios de las licencias Creative Commons.

pensé que podría colaborar con esta investigación. Después de seguir un poco más sus actividades a través de la red, ya que volví a Mexicali, pensé representaba bien del fenómeno que pretendo estudiar.

Regresé a Ensenada en junio. Asistí a otras dos sesiones del taller. Miguel presentó en video algunas piezas que mezclaban la electrónica con el arte a manera de ejemplo. Los asistentes al curso debían entregar como proyecto final una pieza de arte con electrónica. Le propuse colaborar con este proyecto y afortunadamente accedió. Pude ver que se encontraba feliz de que alguien se interesara en su trabajo, gran parte de las labores que realiza son de difusión así que quizá vio en esta investigación otra oportunidad para que su voz fuese escuchada.

Sostuve tres entrevistas con Miguel durante 2011 y 2012. Dos de ellas fueron en Ensenada, y una a través de videoconferencia. Asistí a dos sesiones del Taller de electrónica para

Figura 23. **:** Asistentes a la primera clase del taller de electrónica para artistas. Entre ellos se encontraba Ramón Amezcua (a la izquierda), integrante del colectivo Nortec de música electrónica.

artistas, donde habló de componentes electrónicos como los capacitores y de instalaciones artísticas utilizando electrónica. Me mantuve al tanto de sus actividades, talleres, proyectos y colaboraciones para conocerlo mejor y entender sus motivaciones.

Biografía. Miguel Eduardo nació en Ensenada hace 29 años. Desde muy pequeño se interesó por la tecnología ya que la tenía a la mano gracias a su padre:

> Mi papá tiene un negocio de computadoras que ahorita tiene veintiocho años, casi ya. [...] Entonces yo creo que mis juguetes eran las computadoras, me acuerdo cuando desarmaba las Commodores [64] o las máquinas XT[49]. Antes... yo me

[49] De la Commodore 64 ya se habló antes en este trabajo. Las computadoras XT eran modelos compatibles con IBM populares durante los años ochenta. Todavía son recordadas por muchos de los aficionados que crecieron en esa época como las primeras computadoras personales que utilizaron y en las cuales aprendieron computación.

> acuerdo que las desarmaba y armaba y a veces ni las armaba, nomás las desarmaba por la inquietud de saber qué había, ¿no? Siempre saber qué había, había, había, y yo creo que en base a eso me empecé a incursionar. [...] O sea, las computadoras siempre han sido como tipo mi novia. Bueno, siempre ha sido de que desde la infancia, o sea, hasta que tengo razón, hasta la época. Siempre las he usado como medio, ¿no?
>
> Me puse a programar, me gustaba mucho programar de morro. En PW me ponía a... Pues a programar, supuestamente yo a programar, ¿no? [...] Solamente eran programitas que salieran para que salieran frasecitas, monitos caminando. Era algo bien sencillo, o sea, para mí. Entonces creo que yo ahí me empecé a involucrar bien cabrón con lo que es la tecnología.
>
> Y ya de repente a los diez años, me regalaron mi primer máquina, tenia mi primer máquina, eh... y era una 386 me acuerdo. Y esa máquina me acuerdo que ni duró creo que el mes prendida por qué la desarmé toda, la desarmé toda y yo supuestament... toda, toda, toda, toda la desarmé, y me acuerdo muy bien porque pues me interesaba más lo hardware, saber qué tenia más ahí.

Esta búsqueda que inició en la infancia fue, por la mayor parte, autodidacta. Los únicos cursos de computación que Miguel tomó fueron en la preparatoria, en la materia de informática: "Pero era bien aburrida, nunca entraba". Sentía que el conocimiento de clase era demasiado básico en comparación de lo que había aprendido por su cuenta: "Mejor le decía a la profesora, ¿sabe qué? Mejor hágame un extraordinario". Y así aprobaba la materia.

Su relación con la informática y la tecnología digital se incrementó cuando tenía alrededor de 17 años: "Creo que todo empezó porque tenia una máquina muy lenta y yo lo que quería era, más que todo, hacer *overclock* a la computadora". Esta es una técnica utilizada por aficionados para incrementar la velocidad de un componente electrónico. Éstos son diseñados por el fabricante para trabajar a cierta cantidad de ciclos por segundo (hertz). Sin embargo, mediante modificaciones no oficiales que con frecuencia involucran soldadura,

puede lograrse incrementar la velocidad de componente. Esto acelera el funcionamiento, pero también arriesga el equipo y reduce su tiempo de vida. Miguel resolvió su problema mediante el *overclocking*:

> Lo que yo quería era que mi máquina estuviera más rápida, pero era una pinchi máquina bien lenta. Entonces la hice poquito rápida... Yo lo veo más como si fuera tipo un experimento, ¿no? Siempre era de que siempre veía todo como experimentos, todo, todo, todo.

En las clases de laboratorio donde en ocasiones entraba, Miguel a veces hacía travesuras con el equipo de cómputo:

> Siempre les robaba memorias a las computadoras, me acuerdo. Como era el único morro que sabía cómo era una computadora por dentro, pues sabía cómo meter la mano y sacar memorias. Y hasta me acuerdo que ni las vendía, a fin de cuentas lo que hacía con las memorias las tiraba al excusado. Por hacer travesuras, ¿no? Sólo por hacer travesuras. Por enfadar, ¿no? Y me acuerdo que mucho... Al tiempo, ya era mucho más de hacer virus, más que todo troyanos.

Estos virus que desarrolló a los 17 años consistían principalmente en archivos que desactivaban la CMOS[50] de una computadora, animaciones flash que sin autorización del usuario ejecutaban algún código malicioso, código que enloquecía a impresoras de matriz de puntos.

> Lo que hice con un compa era una diversión bien curada, porque lo que hacíamos era que la impresora se volviera loca. Entonces la impresora enloquecía, era que le mandaba datos por el TCP/IP, impresoras conectadas por red. Tonces lo que hacíamos era controlar el pinche motorcito, o sea, se agarraba así de... [Hace un ruido imitando un pequeño motor acelerado]. Eran puras matrices de puntos. Era, pues, interesante ¿no?

[50] Complementary metal-oxide-semiconductor. Estos componentes almacenan información básica sobre el hardware de la computadora y son vitales para el proceso de inicio.

Miguel se refiere a todas estas acciones actualmente como "puro pinchi *hacking*, así bastardo tonto". También ha cambiado un poco su manera de pensar desde entonces: "Creo que ahorita estoy en una visión de que estoy buscando otras cosas, más ya lo hardware, hardware."

No todas sus acciones fueron solitarias, en esta época se reunía con amigos y conocidos para compartir diferentes actividades relacionadas con la tecnología. Durante un tiempo se comunicaba mediante IRC, un sistema de conversación por Internet, con personas que seguían tutoriales sobre *hacking*. Intentaron replicar los pasos de un archivo que explicaba cómo llamar gratis a través de teléfonos públicos:

> Ese tutorial se hizo muy famoso en latinoamérica por que, en efecto, podías usarlo con los teléfonos de Telnor, güey, los podías *hackear* y podías hacer llamadas telefónicas, uf, indefinidas. [...] Lo que hacía es que tu tarjeta si era de 100 pesos, con ese esquemita hacías una tarjeta que tenia crédito indefinido.

Otra de sus actividades eran LAN Parties los sábados en la colonia Moderna con sus amigos: "Éramos una bolota. Y sí, nos juntábamos más, pero para *hackear*". En estas reuniones reuniones cada quien lleva su propia computadora y todos se conectan en red. Al principio se asignaba un número de puerto a cada asistente. Los puertos son números que utilizan las computadoras para conectarse a programas específico de otra computadora. Aunque son vitales para el funcionamiento de las redes, también presentan un peligro, ya que son puertas abiertas a un sistema. El objetivo de su juego era intentar dejar fuera de servicio (o "tumbar") la computadora de los otros:

> A fin de cuentas solamente era marcar un territorio con nuestros compas, o sea, quién podía tumbar la computadora de la persona. [...] Ahí empezó todo, y pues ahí ya ¡puf! Ya se hizo grande la bola y empecé a involucrarme más ya no tanto al *hacking*, me empecé a involucrar más en otras cosas. [...] Me empecé a integrar más colectivos como ELUG, a los 17 años, de Linux, y empecé mucho a dar mucho lo que es la difusión con gente y empecé a juntarme con gente a hacer pues como festivalillos, del Flisol.

Este ELUG (Ensenada Linux User Group) es el mismo mencionado por Carlos. Durante el periodo de asistencia de Miguel organizaban eventos para difundir el software libre, incluyendo los llamados "Install fests". Estos son eventos en donde los usuarios pueden llevar sus computadoras y se les instala software libre de acuerdo a sus necesidades. Uno de estos eventos es el FLISoL, que significa "Festival Latinoamericano de Instalación de Software Libre" y se realiza una vez al año en diferentes ciudades del continente.

Sus participaciones con ELUG se prolongaron durante varios años, mucho más allá de la preparatoria. Los integrantes del grupo ofrecían pláticas a estudiantes y maestros de la UABC para sugerir alternativas libres a software comercial y educativo. Al explicar el porqué hacían esto, Miguel dice: "Por que nos latía, porque, pues nos identificábamos, o sea... A fin de cuentas era pura gente muy pesada que estaba ahí que eran puros físicos, matemáticos gente de del primer mundo".

Me pareció curiosa esta denominación de "gente de primer mundo", así que pregunté a qué se refería: "Pues, es que... es gente estudiosa, que... no gente normal, gente que tiene mucho conocimiento para... Es más, ni contándome yo... Yo no... Yo estaba afuera, si me comparaba con ellos nunca los iba a alcanzar. Hasta ahorita todavía no. Pero son gente... doctorados, investigadores, que tienen un índice académico bien pesado". De nueva cuenta me llamó la atención que dijera que no son "gente normal", así que le pedí que me describiera a la "gente normal" y respondió:

> Gente que no busca alternativas, gente que no va mas allá, más que todo, no va mas allá. [...] Se puede dar en este caso el software, ¿no? Pues se queda solamente con Microsoft, como Windows, y gente que es así, gente que se quiera, eh... Gente que no le importa, pues, que solamente dice: "Ah, pues yo nomás con que que saque el trabajo y chido".

Uno de los intereses de ELUG era la difusión del software libre y el incremento de su base de usuarios:

> Nos juntábamos y decíamos: "Oye, ¿cómo vamos a explicarle esto a la gente, a los

> alumnos, o a la gente normal esto?" Porque muchas veces la gente, pues no te lo digería fácilmente, ¿no? Entonces también por eso hacíamos eventitos.

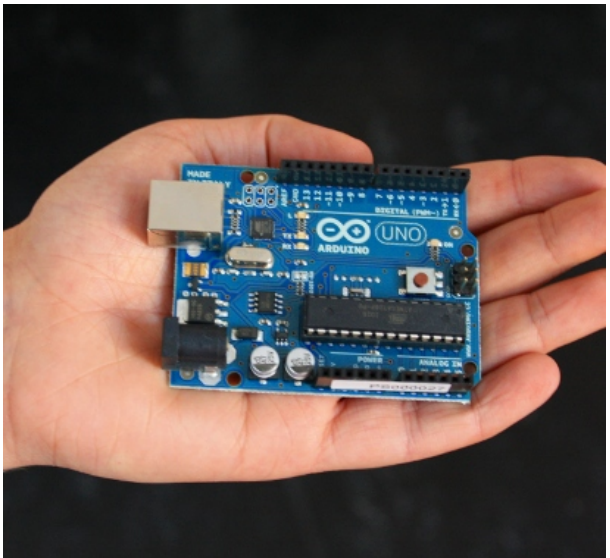
Casi al salir de la prepa Miguel se enfrentó a la decisión de elegir una carrera universitaria. Eligió ingeniería en electrónica en la UABC debido a su interés por la tecnología y el hardware. Sin embargo, después de cursar un año, abandonó la carrera: "Me saturó mucho las matemáticas". Acostumbrado a una experiencia mucho más directa y empírica con la tecnología, tuvo un conflicto con los esquemas universitarios: "Dije: 'Ay, a la monda, ¡ahí se ven!' ¿Qué chingados? Yo quiero algo práctico".

Esto lo llevó a Tijuana a estudiar diseño gráfico en la Univer, de la cual no tiene una buena opinión: "Pinche escuela bien sarra". Decepcionado, viajó a Guadalajara a estudiar lo mismo en la Universidad de Guadalajara. Con la carrera casi terminada, regresó a Ensenada a terminarla en el CETYS, pero se la revalidaron hasta segundo semestre: "Entonces empecé otra vez, casi de nuevo, aquí".

Fue precisamente durante su formación como diseñador donde Miguel experimentó mezclando la electrónica y el diseño. El interés surgió cuando veía proyectos similares en Internet: "Siempre fue como que, wow, yo lo quiero, yo lo quiero aprender, yo quiero aprender."

Este interés lo llevó a conocer lenguajes de programación como Pure Data, el cual sirve para crear audio y video mediante programación. También conoció plataformas electrónicas como Arduino, un microcontrolador libre que sirve para facilitar los proyectos multidisciplinarios.

"Desde el tercer semestre hasta el décimo semestre yo me pagué toda mi universidad, ¿y de dónde salio el dinero? Con puros festivales". Miguel presentaba piezas artísticas y de diseño en festivales de arte y música electrónica, muchos de ellos creados con Pure Data. "Te puedo sacar todos los *flyers*. Son



un putamadral. Son más de doscientos".

Estos festivales a veces se organizaban alrededor de algún músico invitado, o en oportunidades más grandes: "Mira, te puedo decir nombres, eh... Satoshi Tomiie, Doctor Martin, Hippie, Infected Mushroom, Sasha, eh... Sandra Collins, eh... futa madre, güey... Armin van Buuren..." Le pedí a Miguel que me describiera alguna de estas piezas:

> La webcam eran los ojos de la araña. Entonces, si hacía esto [gira la cabeza], la araña miraba, hacia cierto movimiento entonces era mucho como tipo si fuera, eh, parecido a animatronics. Entonces... Otra fue aquí una pieza de un insecto, cuando se inauguró el CEARTE hace cuatro años, que se inauguró con una arañita. Una arañita que seguía a la gente, en una proyección en una pared.

Mientras estudiaba diseño, deseaba involucrarse en proyectos de software libre y colaborar. Debido a que no sabía programar, pensó en ayudar con lo que sabía: diseño gráfico. Su principal preocupación era mejorar la apariencia visual del software libre: "Por que todas las interfases de casi todo el software libre están pa' la madre". Buscó alguna distribución de Linux en la cual pudiese colaborar mejorando o modificando la interfaz gráfica. "Sentía como que no comunicaban nada, no tenían una identidad. [...] Esa fue como que mi pregunta primera, así como, pum, y dije ¡a la madre! Esa es mi pregunta, esa es mi problemática, ahora hay que buscar un target, ¿no?"

Para encontrar algún proyecto en el que pudiera participar, preguntó en los foros de Ubuntu-es[51], en donde recomendaron la distribución de Linux llamada Tuquito, desarrollada en Tucumán, Argentina. El proyecto de remodelación de la interfase de Tuquito aportó al

[51] Este foro en línea sirve para que la comunidad de Ubuntu en español pueda plantear y resolver las dudas de otros usuarios. También para compartir y comentar noticias sobre Ubuntu y el software libre.

software libre, pero también la presentó como tesis de licenciatura. Fue un reto para Miguel, quien nunca antes elaboró una tesis, pero también para su asesora:

> Ella no tenia conocimiento sobre interfases gráficas y fue también como que... Al principio, pues nos dijeron: "¿Qué temas van a escoger?" Yo también toqué el tema y claro que mi tema era el más puñetero, o sea, el tema mas difícil, complejo y... y ella no tenía conocimiento. [...] Claro que lo que me comentaba mucho mi maestra, me decía: "¿Sabes qué? Es que uno de los retos tuyos". Yo soy muy malo para la redacción, muy malo, porque como pienso tan rápido... Las ideas van muy rápido, a una velocidad que a veces me trabo.

Cuando intentaron promocionarlo, pensaron en la creación de carteles y volantes para repartir. Pero había un problema: todo fue diseñado con herramientas libres como Gimp e Inkscape. Para versiones imprimibles de carteles y volantes se requería un complemento privado: Pantone.

Pantone es un sistema de color para impresión. Este sistema divide las imágenes en cuatro componentes: uno amarillo, otro magenta, otro cían y otro negro. Cada combinación de colores en Pantone tiene un nombre y las combinaciones se encuentran patentadas. El problema de utilizar este sistema en el software libre es que se le tiene que añadir un componente de software cerrado.

> Entonces cuando vayas a imprimir estas usando algo privativo. Es como un ejemplo con Ubuntu cuando le instalas el flash. O cuando, un ejemplo bien pelada, cuando tienes el Ubuntu, o cuando, un ejemplo, te crees de súper que eres de software libre y que manejas software libre y de repente le mandas a alguien un archivo en doc.[52] ¿Si? [...] Pues es como que una contradicción. A mi me pasó eso. Yo estuve en un grupo con unos compas, y de repente le digo al profe: "Oye profe, le voy a mandar la ficha académica de un curso". "Ah, simón". Y se la mando en

[52]Se refiere a los archivos *.doc, que es el formato utilizado principalmente por Microsoft Word.

> Word. Y casi todos me querían quemar: "No mames, ¿cómo se te ocurre hacer eso?" Y yo, ¡en la madre! Es que la neta les mandé dos versiones, pero si es cierto, se me chispotió [ríe]. Entonces el plus está en eso, pues, ¿cómo Ubuntu va a tener una herramienta a que nomás te forza, ¿a qué? A que solamente es RGB. Claro que te acepta, pero un *plug-in* privado. Un plug-in que te va a leer un Pantone de Adobe. ¿Por qué no mejor hacer un Pantone libre?

Con esta idea varios integrantes de la comunidad de Tuquito trabajaron en nuevas definiciones de color, usando el mismo sistema. La idea fue crear toda una nueva paleta, un poco más oscura y en base a eso desarrollar un nuevo complemento libre. Un amigo de Miguel calculó hasta el color 300 y lo publicó en la página Gimp-es, un foro en español de discusión sobre software libre para fotografía. Es muy visitada, así que inmediatamente llamó la atención. Miguel trabajó con una hoja de cálculo hasta llegar a los mil colores:

> Pues mi compa subió hasta el 300 y de volada me escribió un correo y mi compa bien asustado me dice: "¿Sabes qué, güey?" Ese güey vive en Argentina. "A la verga, ¿qué hacemos, güey?" Dije: "No mames, güey, pues ya tengo el hexadecimal". Me dice "No mames, güey, no lo sub... no, no, no, ni lo subas a ningún lado, o sea, quédate con ese archivo". O sea, el proyecto está completo, yo lo tengo todo completo.

Jorge recibió un correo electrónico por parte de Pantone. Le anunciaron que iniciarían un proceso legal en su contra por haber violado su propiedad intelectual: "Creo que todavía tiene hasta la fecha la demanda. Él no puede pisar Estados Unidos, no puede, no puede. Chingó pues, o sea, lo liberó, pues. ¿Me entiendes? Lo liberó, lo liberó, se liberó pues". Ante la amenaza de una demanda, de acción legal y su misma juventud, volvieron sobre sus pasos:

> Pero éramos morros y pues nos asustamos y como que: "¡Ay, a la madre!" [...] Borró los Pantone y puso nomás el plug-in, güey. ¿pues qué le quedaba? [...] Pero pues a fin de cuentas no era libre, pues. Nos emputamos y dijimos: "A la verga,

ve cómo es el control, güey". A lo mejor podría ser que la demanda solamente fue un susto, pero pues...

Mientras hablaba de eso, Miguel recordó que todavía tenía el archivo donde habían hecho el Pantone libre: "Ahorita que me acuerdo ahí lo tengo y me dan ganas de liber... Y lo puedo liberar, güey. Lo puedo liberar, me vale verga, yo no tengo visa, a Estados Unidos".

Le pregunté por qué y ante mi asombro me contó que un día tomó unas tijeras y cortó su visa por la mitad. Su última visita a Estados Unidos fue al Museum of Modern Art (MoMA) de Nueva York. Cuando pregunté qué pasó después de que rompió su visa, su respuesta fue: "Pues ya no puedo ir. Pero no hay pedo, sencillo, ya no hay pedo y mucha gente me dice 'Miguel, ¿por qué no vas?' Estoy en contra, güey, como que no me laten muchas cosas de Estados Unidos..."

Pero quería que fuera mas específico sobre el tema de Pantone. Le pregunté si, en retrospectiva, no pensaba que quizá la empresa tuviera razón en establecer la demanda y me pregunta:

-¿Quién es el dueño de RGB?

-No sé- le respondí.

-Yo quisiera que el Pantone así estuviera: No sé.

Mi pregunta incluso lo indignó un poco y lo hizo exaltarse. Me preguntó si el aire tiene dueño, o el agua, y que lo mismo se aplicaba al color: "O sea, los indígenas producen el color. Ya lo están pirateando. Tienen que pagarle regalías al Pantone. Tonces cuando tu utilizas colores naturales de pigmentos, ¿estas pirateando?". Concluyó que muchos diseñadores no saben de este tema que le parece sumamente importante.

A esto se unieron varios casos más de activismo, también "tumbaban páginas" en protesta por inconformidades sociales y políticas. Sobre este tema Miguel se guardó muchas cosas. En parte debido a su puesto y reputación actual, prefirió no hablar detalladamente sobre actos de *hacktivismo*.

Pero además de esas experiencia, sus colaboraciones en Tuquito se enfocaron en torno al entorno gráfico de Linux. La comunidad Tuquito se divide en diferentes equipos, cada uno colabora en un área diferente del desarrollo del sistema operativo:

> Eramos *teams* todos, había el *team* de diseño, el *team* de programación y cada *team* tenía un coordinador y el coordinador era el que le decía. Yo llegué a un momento en donde me tuve que desafanar porque el coordinador de diseño era un déspota. O sea, no, no... no dejaba que los otros opinaran, pues. Entonces tú hacías tu diseño y te lo cambiaba totalmente. Pero horrible así es como que... ¿Sabes qué? Ahí se ven. Entonces yo hice mi tesis y pup [hace el gesto de aventar un documento], ¿sí me entiendes? [...] Claro que a él no se lo entregué, se lo entregué al mero mero que es Mauro, que es el creador de Tuquito.

A pesar de que Miguel se separó del equipo de Tuquito, con la tesis la tesis se tituló como diseñador gráfico.[53] Pero ese no era el único propósito, piensa aplicar el conocimiento desarrollado en su tesis posteriormente:

> Claro que eso me puedo ir con otro y puedo aplicar esos fundamentos, ¿cuáles fundamentos? Todos los que cité. Más que todo mi tesis estoy aplicando como un estándar a fin de cuentas, que es lo que falta, lo siguiente, es que mi tesis es un estándar a que tú lo puedes aplicar en mi tesis en cualquier otro interfaz gráfica. ¿Por qué? Porque vienen muchas citas de mucha gente que esta bien pesada en interfases gráficas. Entonces yo empecé a agarrar tips, empecé a recolectar tips, y esos tips los aplique en esa interfase, ¿no?

Después de graduarse, intentó entrar a ingeniería en electrónica de nuevo. Cursó un semestre y la abandonó de nuevo. Su tiempo era consumido principalmente por su trabajo de consultoría en arquitectura:

[53] La tesis completa se puede leer en: `http://www.miguelmonroy.com.mx/temp/tesis.pdf`

> Empecé a ganar mucho dinero y pos dije, ¿sabes qué? Aquí hay lana y ahorita ocupo lana, pues no ocupo perder el tiempo tanto, ¿no? Y empecé a darle. Duré bastante yendo a electrónica, me encantó bastante. Conocí a mucha gente y creo que hasta la fecha, yo creo que me voy a volver a meter otra vez, porque tenía un buen promedio yo creo que fue el mas alto del salón, de promedio, de electrónica.

Su siguiente trabajo fue en el CICESE. Ahí hizo tests de programación de Oracle y diseñó interfases gráficas donde aplicó los conocimientos construidos en su tesis. También aprendió a escribir hojas de estilo para páginas web. Cuando lo entrevisté ya tenía cuatro años trabajando en el CEARTE en diferentes puestos:

> A mi nadie me recomendó este puesto. Nadie me recomendó entrar aquí. Nadie. Nadie. [...] O sea, no le dije a mi papá: "A ver recomiéndame y eso". No. Yo estoy aquí porque yo pregunté y yo empecé aquí en este trabajo desde abajo, como recorridos guiados. Primero como limpieza, luego como recorridos guiados y empecé a subir tac, tac, tac, tac, y pues ahorita soy encargado de un área.

En la actualidad es encargado de las áreas de educación a distancia y del área multimedia. En CEARTE, Miguel ha impartido y organizado diferentes talleres sobre tecnologías libres como Arduino y Pure Data.

Entre los proyectos que Miguel ha desarrollado fuera de su trabajo se encuentra el colectivo NorteLab. Cuando ELUG se desintegró por falta de interés de los miembros, Miguel deseaba crear otro espacio de colaboración en equipo:

> Pues si, la gente no se comprometía, pues que no, no, no, no se comprometía, pues, no, no iban a las juntas. O sea, eramos siempre los mismos, los mismos, y dijimos "¿Sabes qué? Muerte a ELUG". Y pues cada quién en su rollo. Era mejor, vaya, no estar así como en un grupo donde ni los fundadores estaban.

La solución fue crear un colectivo nuevo, con miembros nuevos. La idea surgió cuando una persona llegó a CEARTE a pegar un cartel:

> Él un día llegó y quiso pegar un cartel sobre un lenguaje orientado a objetos que se llama Pure Data, y me dijo "Oye, dónde lo puedo pegar". Yo estaba antes en los recorridos guiados. Le dije: ¿Y esto donde va a ser?" "No pues que yo voy a dar un taller". "No pues, ¿en dónde?" "En mi casa" Y yo, "ah, cabrón". Le dije "Eh, oye yo sé programar esto". "¿A poco?" "Sí". Me dice: "Ah, pues hoy en la noche tengo una tocada en el Jazz Café. ¿Quieres ir? Pon tus visuales, a ver si es cierto". Llegué e hice al bato se quedó así como que "Guau".

Desde ese momento Adrián y Miguel colaboraron juntos, aunque a veces a distancia debido a que Adrián se fue a estudiar su maestría a University of California, San Diego (UCSD). Crearon NorteLab inspirados en las reuniones de Dorkrobot, que es un concepto similar a los *hackerspaces*. Consiste en tener reuniones informales para discutir temas de creatividad electrónica para crear comunidades de creadores.

Adrián y Miguel se propusieron como objetivo de NorteLab el confrontar ideas y crear un espacio donde los miembros fueran "efímeros": "Todo salió por que queríamos un espacio donde todo mundo se confrontara, donde todo mundo chocara, donde todo mundo platicara, ¿no? Entonces lo curioso del proyecto era un laboratorio NorteLab de que es cualquiera puede llegar. Cynthia se pueden ir y puede regresar cuando quiera, es un espacio solamente, ¿no?".

Yo le pregunté por qué decidió listar a NorteLab en la página de hackerspaces.org si no se trata de uno de estos espacios: "No lo he corregido, porque el hackerspace está en mi casa. [...] En mi estudio, mi taller. Yo tengo un estudio de atrás de mi casa y va gente, pues. [...] La ubicación siempre está cambiando de lugar. ¿Por qué lo cambié? Porque la ultima vez el taller de electrónica fue ahí, en CEARTE, tonces siempre está cambiando de ubicación el *hackerspace*".[54]

NorteLab participó en diferentes festivales de música electrónica y ofreció diferentes talleres. Entre ellos el de "Electrónica básica para artistas", al cual asistí en algunas sesiones.

[54] Esto es común en los inicios de un *hackerspace*. En la página de hackerspaces.org existe también una sección que lista los *hackerspaces* que se encuentran aún en proyecto y muchos de ellos no tienen una ubicación fija.

Figura 25. **:** Una sesión del taller de electrónica para artistas el 7 de junio de 2011 en la sala multimedia de CEARTE. Cynthia muestra el funcionamiento de un capacitor a los estudiantes mientras Miguel observa.

La maestra Cynthia, quien inició el curso, es parte del colectivo y se hizo cargo de la parte electrónica del curso. La parte artística fue impartida por Miguel y ambos asistían a las clases del otro para aprender: "Todos a fin de cuentas como es un colectivo, todos podemos ser parte de él y muy interesante, yo digo que esa es la misión".

Para el enfoque a tomar en el Taller de electrónica básica se presentaron varias propuestas en un principio. Todos tenían claro algunos principios que querían que guiaran el diseño del programa:

> No queríamos que fueran solamente operadores. [...] Que solamente siguieran instrucciones. Que solamente siguieran tutoriales. ¿Por qué? Por la gente cuando sigue un tutorial se enfrenta ante una problemática, ¿cuál? Que qué llegara a pasar si se descompone el aparato. No sabe de dónde viene la bronca. Entonces,

> al deducir problemas sabes dónde está el pedo, por eso lo hicimos tan teórico, pues.

Uno de los proyectos en los que estaba más entusiasmado la última vez que lo vi estaba relacionado con la electrónica. Miguel invitó a Arcángel Constantini, un artista multidisciplinario, a dar un taller de *circuit bending* en CEARTE. Después de una de las sesiones Miguel dejó a Arcángel en un tianguis local llamado Los Globos:

> Ya lo fui a recoger, ya me fui al trabajo y lo dejé allá una hora. Dos horas... Dos horas y media. O sea, el bato regresó y me dice: "Güey, ¡lo que me encontré!" Y yo: "¿Qué?" "Me encontré un juguete que se llama Mindflex. El Mindflex es un juguete que te sale... En eBay te sale en 76 dólares. En Amazon te sale en 99 dólares. Ahí lo encontró en 30 pesos, güey. Y entonces, esa madre es... Cuando me lo enseñó me dice: "Güacha, güey, me lo encontré". Yo le dije: "¿Qué es, güey?". Dice: "Me lo encontré, güey, es un analizador de ondas cerebrales".

Ese juguete incluye una diadema que posee sensores que captan información cerebral. En base a la información recopilada se pueden mover objetos a distancia. Para mí, comprender esta tecnología ha sido un poco complicado, y aún no la comprendo del todo, pero este tipo de juguetes ya tienen bastante tiempo en el mercado. El objetivo de comprarlo, para Arcángel, fue la posibilidad de modificarlo y utilizarlo en su trabajo.

Miguel volvió a Los Globos buscando algún juguete como el que encontró Arcángel. No había unos idénticos, pero encontró algunas diademas de otro juguete parecido llamado "Star Wars Science - Force Trainer." La idea era *hackearlo* para conectarlo a Arduino, y éste a su vez comunicarlo con Pure Data y generar información visual con la

mente: "Teniendo datos a Pure Data puedo hacer una instalación interactiva donde los morros les conecte unos cascos y generen gráficos audiovisuales en una instalación". También intentará colaborar de esta forma con Arcángel en el festival Dorkrobot, en el D.F.:

> Por que él decía: "pues hay que presentarnos. Dame quebrada de presentarme". "¿Sabes qué? Es que tú no generas audio. Tonces, lo que vamos a hacer, vamos a hacer un *battlegate* de dos". Él tiene uno de estos y yo uno de estos [se refiere a la diadema del juguete]. Tonces él va a crear la música con uno de estos y yo voy a crear el video con esto. Tonces va a estar curada porque, pues, los dos artistas, pues, tratando de... interpretar cosas con... con el cerebro, ¿no?

Aunque también Miguel ve este recurso como un "nuevo medio", o un nuevo diálogo con las computadoras. Una posible manera de interactuar diferente, controlar la máquina de maneras novedosas y quizá de reemplazar el teclado y el ratón. ¿Por qué lo hizo?

> Más que todo la experimentación y experimentar y también yo quisiera esto compartirlo, ¿no? Me interesa mucho a mí la onda de compartir y pues trato, ¿no? De compartir. Que a fin de cuentas en el taller de Arcángel pues eso fue, ¿no? Yo fui como su asistente y le estuve ayudando a la gente compartiendo. [...] Tonces, más que todo es la búsqueda de nuevos medios.

También le llegó una invitación para impartir un taller en el Instituto de Matemáticas Aplicadas y Sistemas de la UNAM y se la preparaba durante nuestras entrevistas. Otro

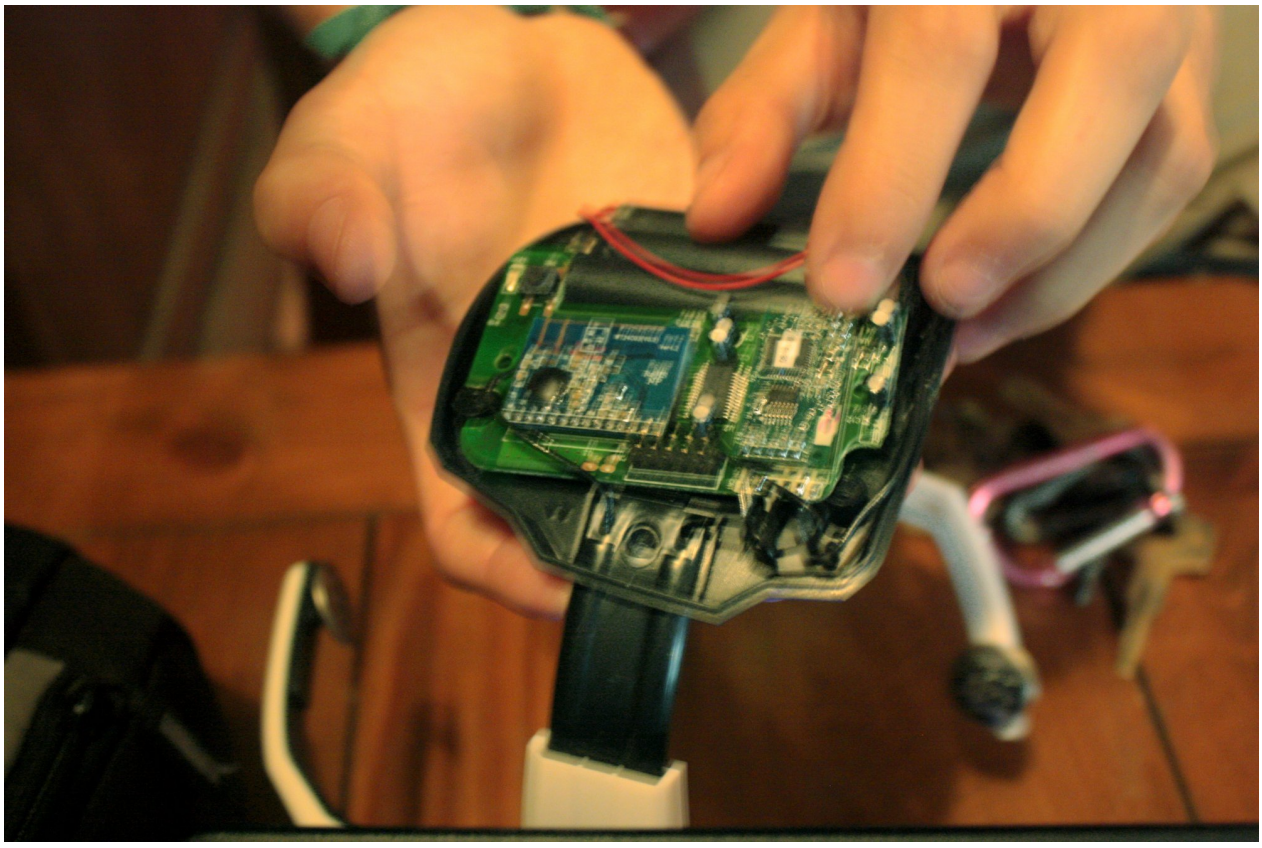


Figura 27. **:** Miguel mostrándome la diadema del juguete "Force Trainer," específicamente el componente Neurosky.

proyecto en el que se encuentra involucrado es en la creación de un libro libre sobre Arduino. Éste es un proyecto iniciado por David Cuartielles, uno de los creadores de Arduino.

Miguel pensó que como diseñador podría ayudar en la maquetación del libro, mientras que otros equipos se encargaban de la redacción, la traducción, los diagramas y otras cosas. Al momento de la entrevista, Miguel no encontraba todavía un programa libre adecuado para esta tarea. Uno que pudiese resaltar con colores el código de programación automáticamente. Intentaron con procesadores de texto sencillos como Microsoft Word o LibreOffice:

> Pero decíamos, es que esa madre no es para eso, pues. Entonces gracias a eso yo descubrí que el Adobe Ilustrador, el Corel no te sirven para hacer esquemas de libros de programación. Entonces gracias a eso agradezco a la comunidad que pues empezó a surgir mas y mas preguntas... Cada vez salen más y más preguntas.

Sin embargo, el proyecto se encuentra actualmente casi detenido. En la lista de correos no ha habido mucha discusión y el avance ha sido muy lento: "Como que... creo que la misma gente misma de ese proyecto ha estado parada, creo. ¿Por qué? Por sus quehaceres, ¿no? Y es además un proyecto donde cada quien pone un granito entonces igual, es una comunidad".

Una de las cosas que Miguel se cuestiona del trabajo de su vida es el tener intereses muy diversos y no especializarse en una sola cosa para convertirse experto en ella:

> Es lo que me dice mucha gente, güey, debes de centrarte en algo, por que yo soy una persona que, te vuelvo a decir, soy una persona que me cuestiono un putero, de muchas cosas, pero de muchísimas cosas. Entonces cada vez que me estoy cuestionando soy más duro, soy más duro, más y más y más y más yo no puedo ser una persona que... me puedo especializar en una cosa. No puedo, güey, no mames, o sea, no puedo...

En uno de los momentos de la entrevista, Miguel reflexionó extensamente sobre las motivaciones detrás de sus actividades. que ha tenido a lo largo de su vida para hacer lo que ha hecho e involucrarse en los diferentes proyectos que me ha platicado:

> Me he comportado como un niño. Soy como un niño explorador, güey, como que me gusta explorar como que eso no me quiero, ajá, totalmente cerrar tajantemente: "Sólo soy diseñador web y ya." Como que digo: "A la verga, eso no me interesa" y digo "eso no quiero", como que una forma de exploración. ¿no? Tonces yo es como a la forma de exploración hace que uno se está autoconociendo esta conociendo y de alguna forma, es un ejemplo muy sencillo: El Pantone. ¿A qué me sirvió haber subido un Pantone a la red? Qué, a la chingada, hay una extradición de una persona que no puede pisar Estados Unidos. Eso lo sabemos muy pelada, pero, ¿qué conocimiento me dejó eso? Hay solamente un conocimiento, ¿qué, güey? Que eso tengo que difundirlo ante los diseñadores que deben de saber, güey, que saliendo de diseño gráfico su culo va a pertenecer a Pantone. [...]

> O sea, si eres extremadamente de software libre, como Richard Stallman, que dice que Ubuntu es privado y que él hizo su propio sistema operativo que no es privado y que no carga paquetes privados y si te pones extremadamente izquierdista y a la verga, acá, así güey... todos los diseñadores... Muy pelada, no imprimirían carteles. Así de sencillo. [...] Pero pues las escuelas de alguna forma te educan a depender, ¿no? Es como igual, también mi tesis, a lo que iba. ¿Por qué chingados en las escuelas te hacen depender de software que son puramente caros que un estudiante en la carrera no tiene derecho a pagarlos, ¿por qué? Por que no tiene tanto dinero para pagar mil dólares o quinientos dólares o cien dólares si con apenas anda pepenándola. ¿Por qué no chingados enseñar en una escuela o porqué chingados no hay un sistema de educación donde la educación es libre y laica, sencillamente? Es como, igual, como si un poeta abriera su pensamiento a la libre difusión. Es como hablar de un poema y hacerlo libre y órale cabrones, modifíquenlo. Es como una mutación interesantísima. La gente todavía está como que a la madre, güey, es como que cuando estas viendo una fotografía como que la subes con miedo, ¿no? Yo creo que por eso mi tesis la hice libre. Por que a mí me interesaría que la gente.... Yo pensé que es un sueño viajado diciendo "¡Ay, que ojalá que mi tesis fuera como un estándar!" Yo quisiera que esa madre otra persona la agarrara y la modificara, solamente que reconociéndome, ¿pues de quién fue la idea? Pues mía. Yo ya de alguna forma quisiera que eso fuera un estándar, yo digo que si fue una idea y si no pega no hay pedo pero el autoconocimiento lo tuve. Tonces eso nadie te lo va a dar.

También reconoce cierto choque de intereses al querer la liberación de la información, pero trabajar para el gobierno que tiene esquemas más rígidos y de control.

> No sé cuanto quede ahí, porque yo voy a estar buscando más y más y más y no tanto por que a mi la neta, güey, aquí entre nos se me hace bien sarra el gobierno, güey, bien sarra. Por que te tratan bien sarra, les interesa puros números. Si me

> gusta mucho lo de *research lab*, me gusta mucho crear cosas, laboratorios, dejar algo, con fines creativos, me interesa mucho. Pero no me empezó a gustar eso del gobierno de las cosas burocráticas, de la gente que tienes que hacer. En el trabajo se ha manejado mucho que si tienes buenos conectes tienes un mejor empleo, güey. Eso que digo: [...] Esa madre no me interesa. [...] La neta, güey, si fuera persona lame huevos [...] me subieran a un pinche puestazo. [...] No tengo ese interés. A mi me interesa más el dejar algo, la comunidad...

Hasta aquí quisiera dejar la historia que Miguel me contó. Entre las diferentes visitas y convivencias que tuve con él, hay una anécdota que me parece sumamente significativa y que se detallará a continuación.

Visita a BajHack. Cuando le hice la segunda entrevista en Ensenada, le comenté que unos días antes había iniciado otro *hackerspace* en la ciudad. Lo encontré listado en la página de hackerspaces.org y tenía como nombre BajHack.[55] Conseguí el domicilio del lugar, contacté a uno de los encargados del espacio, Antonio Maldonado, y junto con Miguel visité el espacio.

En su página principal decían que todos los miércoles se reúnen a las 6:30 pm, así que llegamos ese día aproximadamente a esa hora. Nos encontramos con Antonio, uno de los fundadores, quien nos dio acceso y posteriormente llegó otro miembro del *hackerspace*. Nos presentamos. Miguel habló brevemente de sus principales proyectos y yo de la investigación que me encontraba realizando.[56]

Los dos son ingenieros en electrónica y trabajan en una fábrica de lentes llamada Augen. El *hackerspace* apenas llevaba un mes en funcionamiento y el aspecto era más bien parco. Rentaron un departamento pequeño, dentro del lugar había solamente una mesa con cuatro sillas, un sillón y dos muebles con repisas. En uno de ellos había algunas cajas con componentes electrónicos pequeños. También había una pequeña cocineta pero no aparentaba estar en servicio. En la barra se encontraba un cautín.

[55] La página oficial de este *hackerspace* es: `http://bajhack.com/`

[56] Nuestra visita fue mencionada en uno de los posts del blog de BajHack: http://bajhack.com/index.php/archivos/50

Figura 28. **:** Miguel mostrándonos su trabajo visual en BajHack. Fotografía tomada por Antonio Maldonado con su celular y publicada originalmente en el blog de BajHack.

Nos platicaron que ambos tenían la idea de crear un espacio para la colaboración en proyectos de electrónica y cuando conocieron la idea de los *hackerspaces* les hizo mucho sentido. Iniciaron el lugar cobrando 400 pesos al mes a cada uno de los miembros. En ese momento se encontraban trabajando en un proyecto para un amigo suyo. La idea era crear una pantalla grande para una báscula de camiones, ya que querían tomarle foto al camión junto con el peso, pero la pantalla de la báscula era muy pequeña. Lo que estaban haciendo era modificar una pantalla de autobús estadounidense para que cumpliera esa función. Con este proyecto esperaban obtener fondos para comprar más material. Muchos *hackerspaces* no harían este tipo de proyectos, ya que por la mayor parte se trata de espacios no lucrativos.

Miguel sacó su computadora y mostró algo de su trabajo visual en YouTube. También explicó cómo lo programaba en Pure Data. Esa fue la primera oportunidad que tuve de ver este sistema en funcionamiento después de escuchar tanto sobre él. A mi me sorprendió el interés mostrado por los ingenieros del *hackerspace*. Hacían preguntas y se encontraban sumamente interesados en el sistema y su funcionamiento.

Posteriormente Miguel explicó otro proyecto que le gustaría desarrollar, enfocado principalmente a panaderías, pero aplicable a otro tipo de negocios. Lo describió como "un sistema para avisar cuando hay pan caliente" en la panadería. A ninguno de los tres nos quedó muy claro a qué se refería, así que preguntamos más: "¿Cómo vas a medir la temperatura del pan? ¿Lo medirás en el horno o ya cuando se encuentra a la venta?". Miguel nos dijo que eso no era necesario, simplemente había un interruptor para el panadero.

Aún no nos quedaba claro. Si el panadero debía activar un interruptor cuando el pan estuviera caliente, ¿cuál es la utilidad del sistema? Después de muchas explicaciones, entendimos que se trataba de un sistema para que el panadero pueda anunciar, a través de Twitter, mensajes de celular, o algún otro medio, cuando el pan está recién hecho. De esta manera los clientes podrían acudir a la panadería en ese momento. Miguel prosiguió: "Lo he ofrecido pero hasta ahorita todo mundo me ha bateado".

Antonio dijo: "A ver, explícamelo otra vez por que yo estoy a punto de batearte". La manera tan repentina de decirlo me dio risa, pero noté que a Miguel no le agradó en absoluto. "Te voy a poner un ejemplo", prosiguió Antonio, "cuando salieron los celulares yo me suscribí a las noticias por mensajito. Y cuando me llegó el primero, dije 'qué bien'. Pero luego, a cada ratito me estaban llegando mensajitos de tonterías, y como al tercer día lo desactivé porque me enfadó. Entonces si tu haces un sistema que todos los días les envíe un mensajito a tus clientes, podrías terminar por enfadarlos y te saldría contraproducente".

Aclaró el motivo de su confrontación: "No te estoy diciendo esto por mala onda ni nada, ni para romperte la ilusión de tu proyecto. Te lo digo porque si lo quieres vender, tienes que convencer al cliente de que le va a dejar dinero, de lo contrario te van a batear". Yo añadí que quizá ese sistema no beneficiaría tanto a una panadería pequeña. Tienen unos clientes muy locales, quienes ya saben a qué hora hay pan caliente, por lo que no se requiere un sistema tan complicado.

Antonio continuó: "¿Sabes qué podrías hacer? Imagínate poner un letrero afuera que tuviera fotos de cada pan. Ponle una concha, una dona, un corico y eso. Y cuando hubiera

donas calientes se prendiera el foco y la gente que fuera pasando viera que hay donas calientes en ese momento". Nos quedamos pensando un poco en esa idea, y al menos a mí me pareció muy buena y añadí: "Si yo viera eso, llegaría a comprar pan".

Miguel seguía sin estar convencido. La conversación giró en torno al tema poco tiempo más, pero pronto lo abandonamos. Nos despedimos después de un rato y mientras íbamos en el carro, Miguel me dijo: "¿No sientes como que estos batos están bien clavados en sacar lana? Además, ese proyecto del que les platico ya existe, ya funciona en otras partes".

Creo que lo que vi ahí es, principalmente, un choque de éticas profesionales. Los creadores de BajHack son ingenieros, trabajan en una fábrica. Lo que se busca en esos trabajos es muy pragmático: Sacar adelante la producción, optimizar los recursos, incrementar ganancias. En cambio, Miguel trabaja en el Centro Estatal de las Artes, utiliza tecnología con fines educativos y artísticos. Además, Miguel busca ofrecer talleres gratuitos con el respaldo de una institución y BajHack es un proyecto independiente que paga luz, renta y materiales. La propuesta de Miguel iba en contra de los valores de Antonio, y la propuesta de Antonio iba en contra de los valores de Miguel.

Por otra parte, esa forma de confrontación me parece más propia de las ingenierías. Me refiero al decir bruscamente: "Esto no sirve, cámbialo" sin endulzar las frases ni preocuparse por el aprecio que la otra persona tiene por su propio proyecto. Incluso yo he tenido problemas al verbalizar observaciones en ese tono en ámbitos alejados de la ingeniería.

En lo personal, también me pareció interesante la dinámica que se hizo en el espacio. Miguel llegó con una idea, y a través de los comentarios de los otros se mejoraba y modificaba. Esto lo he visto suceder en la mayor parte de las reuniones a las que he asistido. Alguien llega con una idea, todos piensan en ella y dan su punto de vista. Generalmente la persona sale del lugar con una idea enriquecida. Si toma o deja las sugerencias ya es cuestión de él.

Cuando lo entrevisté por segunda vez, Miguel volvió a recordar el incidente sin que yo lo mencionara. Se sintió afrontado personalmente por Antonio. Le comenté: "Pienso que te estaba dando su punto de vista sobre tu trabajo y te estaba dando sugerencias". Miguel me

Figura 29. **:** Segunda visita de Miguel al *hackerspace* BajHack. A la izquierda, con camiseta de rayas, se encuentra Antonio Candela. La fotografía fue proporcionada por Antonio Maldonado.

dice: "Pues si, pero no tenía por qué hacerlo tan agresivo. ¿Y por qué enfrente de todos?" Miguel asistió a una reunión posterior en el *hackerspace* y los ingenieros insinuaron que ya era hora de que pagara la cuota mensual. A Miguel no le gustó la insinuación, ya que todavía no se sentía parte del lugar. A partir de ese incidente hubo una ruptura y ambos bandos cortaron comunicación. Miguel se encontró de nuevo a Antonio en el CEARTE, llevando a su hijo. Se saludaron pero no se detuvieron para platicar, las diferencias de intereses fueron demasiado grandes como para colaborar juntos.

Analizar éticamente los usos sociales de la tecnología

Como ya se mencionó, la ética ha sido un problema que se encuentra en gran parte de las investigaciones sobre los *hackers*. Desde los primeros trabajos académicos sobre el impacto de prácticas como la piratería y la intrusión ilegal de sistemas, ha sido notable que las prácticas *hacker* tienen un alto componente ético (Denning, 1996; G. R. Meyer, 1989; Mishkin, 1983). La mayor parte de estos trabajos se enfocan en el aspecto legal de estas prácticas.

El trabajo que avivó y reconfiguró esta discusión fue el libro de Pekka Himanen "La ética del hacker" (2001), donde se piensa la ética del *hacker* como una ética de trabajo: "Si se les considera a este nivel, los *hackers* informáticos constituyen un ejemplo excelente de una ética del trabajo más general, a la que podemos dar el nombre de la *ética hacker del trabajo*, afianzada día a día en la sociedad red, en la cual el papel de los profesionales de la información se halla en expansión" (Himanen, 2001, pág. 29).

Las bases que utiliza Himanen afirmarlo se afianzan en largas tradiciones sociológicas que se remontan a Marx y Weber. Marx desarrolló la idea de que dentro del capitalismo la labor se disocia de la persona, convirtiéndose únicamente en un medio para satisfacer una necesidad externa (Marx y Engels, 1966; Ritzer, 2005) de manera que el trabajador se encuentra alienado e impotente por no disponer del producto de su trabajo. En "La ética protestante" (2008), Weber explica cómo la ética de trabajo protestante ayudó a la expansión del capitalismo en países como Estados Unidos. El trabajo es para el protestante el medio para obtener la salvación: un buen trabajo en esta tierra significa una recompensa eterna. (Himanen, 2001)

A pesar de los estudios anteriores sobre el tema, la mayor parte de ellos se quedan en la abstracción. Himanen, por ejemplo, únicamente recurre a los grandes referentes: Linus Torvalds, Eric S. Raymond y Richard Stallman. Su análisis parte de publicaciones y del discurso oficial de los *hackers*. Esto quizá se deba a que el trabajo de Himanen es filosófico. El intento más antropológico para estudiar la ética *hacker* es el que actualmente realiza Gabriella Coleman, quien se enfoca principalmente a los desarrolladores de software libre y

sobre todo la comunidad Debian.(G. Coleman, 2004, 2005; E. Coleman, 2005; E. G. Coleman y Golub, 2008)

Aún así, hay mucho trabajo pendiente. Existen varios recursos para comprender la ética a través del discurso de las personas (Cheney y cols., 2010). Tres recursos particularmente pueden ayudarnos a encontrar la ética en las prácticas tecnológicas de los *hackers*.

- Compartamentalización: En ocasiones los individuos tienen divisiones éticas entre sus diferentes actividades o experiencias. Esto permite comportamiento que evita tener una sola ética para todas las actividades de la vida de un individuo.
- Esencialización: Sucede cuando se utiliza un individuo, lugar o principio como ejemplo para representar todo lo malo, o todo lo bueno.
- Abstracción: Es cuando los individuos reducen la ética a una serie de principios o teorías, abstrayéndola de las prácticas.

Andy Isaacson. El viaje que hice a San Francisco para entrevistar a Carlos tuvo también otro objetivo: Visitar el *hackerspace* Noisebridge y encontrar otro informante. En una visita a la reunión 2600 de San Diego me comentaron que en San Francisco había muchos *hackerspaces*. Cuando Carlos me dijo que se mudaría para allá, le comenté el dato. Esto despertó su interés y decidió visitar Noisebridge, que si bien no es el único *hackerspace* en el área de la bahía, es el único localizado en San Francisco. Me comentó que tenían cursos y que había muchísima gente interesante con quien platicar.

Antes de ir decidí averiguar más sobre el lugar a través de su sitio web.[57] En él aparecía listado Danny O'Brien como secretario, así que lo contacté para avisar que iría a investigar. También se encontraba la localización geográfica del lugar y tomé nota para encontrarlo fácilmente.

Cuando llegué a San Francisco me encaminé por la Mission street buscando el lugar. Es una calle con muchas fruterías mexicanas, restaurantes de comida sudamericana, mexicana, china, italiana y japonesa, incluso había también una tienda de mariguana medicinal. En

[57] `http://www.noisebridge.net`

los alrededores había también muchas *smoke shops*, aproximadamente dos por cuadra. Finalmente llegué a la reja frontal que se encuentra precisamente al lado de una frutería cuyas cajas invaden la acera de nombre "Mi ranchito". Lo identifiqué por una lona con el logotipo impreso arriba de la reja. Justo en ese momento una mujer la abría y le pregunté si era ahí. Comenté que buscaba a Danny. Me confirmó que estaba en lo correcto y dijo que no conocía a Danny, sin embargo me dejó pasar.

Figura 30. **:** Reja frontal de acceso a Noisebridge. Para entrar se necesita una llave o tocar el timbre para que alguien permita el acceso desde dentro.

Noisebridge está localizado en un tercer piso que se accede mediante una escalera de aspecto descuidado. Al abrir la puerta del lugar quedé verdaderamente sorprendido: El lugar me pareció enorme. Lo primero que vi fue un gran librero repleto de libros técnicos, repisas con dispositivos electrónicos y cables, carteles y dibujos en las paredes e incluso una pequeña cocina. Algo que llamó poderosamente mi atención fue una pantalla LED colgando en lo alto de una de las paredes. En ella se representaba una simulación del juego de la vida inventado por Roger Conway. A un lado se encontraba un cartel con el lema del *hackerspace*: *be excellent with each other, dudes*. (Sean excelentes unos con otros).

Una vez dentro, me encontré bastante desorientado. En el lugar había unas diez o quince personas trabajando en diferentes proyectos.[58] La mujer que me dejó entrar me

[58] Con el tiempo me he dado cuenta de que esa es muy poca asistencia, normalmente en un día común puede haber sesenta o más personas.

Figura 31. **:** Parte de la biblioteca de Noisebridge, una pantalla que mostraba una simulación del juego de la vida de Roger Conway y un cartel que muestra el lema del *hackerspace*.

ignoró se sentó a trabajar en una laptop, abandonándome a mi suerte. Me acerqué a unas personas en la cocina y les expliqué que estaba haciendo investigación y que estaba buscando a Danny O'Brien. Me comentaron que no estaba pero que uno de ellos, Rob, me podría dar un tour por el espacio. "*He's great to talk to*", me comentó su acompañante.

Me llevó a la entrada e inmediatamente me sentí como en un paseo turístico donde el guía tiene memorizado todo el discurso que da a los visitantes. Me mostró un pequeño mapa y las reglas del espacio. Me explicó que Noisebridge lab es una organización no lucrativa 501(c)(3), lo cual los exenta de pagar impuestos. Sus principios se basan en lo que ellos llaman la "do-cracy" que consiste en que la persona interesada en hacer algo es la que tiene el poder en ese momento (sobre esto se hablará más adelante). El lugar está abierto para proyectos tanto internos (de los mismos miembros) como externos y cada quien puede llegar

Figura 32. **:** Vista general de Noisebridge, hacia el ala oeste.

Figura 33. **:** El "hack rack" dentro del área de costura.

a trabajar en sus proyectos ahí. Simplemente deben seguir el lema: “Sean excelentes unos con otros, y eso nos ha funcionado muy bien”, añadió Rob.

Noisebridge cuenta con dos salones de clases, un taller de carpintería y metales, una cocina en donde cualquiera puede preparar sus alimentos y también se dan clases de cocina. Hay una zona de costura con algunas máquinas de coser industriales y otras domésticas. Existe una zona denominada *hack rack*, que es un espacio con ropa donada que cualquiera puede tomar para modificarla. Hay un criadero de hongos comestibles, un cuarto oscuro de fotografía, algunas cajas con botellas de cerveza a medio fermentar, un área de trabajo para electrónica y repisas con componentes electrónicos que cualquiera puede usar.

El lugar se sostiene mediante diversas donaciones. Aunque Noisebridge está abierto al público, se puede solicitar membresía mediante una donación mensual. Esto garantiza algunos privilegios extra, por ejemplo un lugar personal para almacenar cosas y llaves para

Figura 34. **:** El área de la cocina se denomina "tastebridge".

abrir la reja principal. Se venden algunos artículos pequeños como camisetas para obtener fondos. Había también una especie de "pasaportes" ficticios. En ellos se coloca un sello con el logotipo de cada *hackerspace* que visita el portador, como una forma de fomentar la visita a diferentes lugares como este.[59]

El tema de los *hackerspaces* es enorme y necesita de, por lo menos, otra tesis. Actualmente existen más de mil de estos lugares funcionando en todo el mundo. Académicamente es un tema que se empieza a tratar. Andrew Schrock intenta pensar en estos lugares como espacios de colaboración de *hackers* y *makers*.[60] Me pareció importante dar una descripción general del lugar y su manera de trabajar debido a que mi siguiente informante, Andy Isaacson, fue uno de los fundadores del lugar.

[59]La explicación de este pasaporte, así como el archivo PDF imprimible con el formato pueden encontrarse en: `https://www.noisebridge.net/wiki/Passport`

[60]En blog de Andrew Richard Schrock pueden encontrarse las últimas actualizaciones sobre su trabajo. el URL es: `http://andrewrschrock.wordpress.com/`

Cuando Rob terminó el tour, le comenté que buscaba alguien a quién entrevistar antes de regresar a mi ciudad. Sugirió enviar un correo electrónico a la lista de discusión de Noisebridge explicando la situación y preguntando si alguien tenía tiempo para una entrevista. En la página de internet del lugar advierten que la red inalámbrica de Noisebridge no era de fiar. Imaginé que posiblemente habría algunos *hackers* monitoreando el tráfico. Le pregunté a Rob al respecto, pero me tranquilizó y dijo que revisara mi correo electrónico sin preocuparme.

Con el tiempo comprendí la importancia de la lista de anuncios de Noisebridge. Es similar a un tablón de anuncios: Cualquiera de los suscriptores puede leer los mensajes y colaborar en las discusiones. Los temas son muy variados, van desde anuncios de clases, nuevos eventos y cambios físicos en el lugar, hasta quejas sobre algunas de las condiciones como la limpieza. También se hacen sugerencias y propuestas, incluso algunas personas sueltan ofertas de trabajo por si a alguien le interesa, aunque hay una lista exclusiva para ello.

Al poco tiempo de enviar un mensaje recibí una respuesta por parte de Andy Isaacson, quien me comentaba que tendría tiempo al día siguiente. Me envió también su dirección de Twitter y la página de su proyecto Noisetor, del cual me comentó un poco Rob. Me citó en un café llamado Blue Bottle en la plaza Mint a las once de la mañana del sábado. Me dijo que lo reconocería por su cabello azul.

Llegué al lugar y se encontraba desayunando un plato de avena. Platicamos un poco y le expliqué los objetivos y enfoque de mi investigación. Se sorprendió de que Rob conociera su proyecto, ya que él no conoce a Rob y nunca ha platicado con él al respecto. Es uno de los primeros casos, dice él, de "comunicación asimétrica" que ha tenido: Aquellos casos donde los demás saben información sobre él, pero que él no tiene nada de información sobre la otra persona.

Después de tomarme un chocolate caliente, sugirió que fuéramos a otro lugar para entrevistarlo, donde hubiera menos ruido y menos gente. Me contó que su exnovia había hecho investigación y se volvía loca al transcribir algunas entrevistas con ruido ambiental. Dijo que tenía un lugar planeado, pero que si no funcionaba, tenía el plan de contingencia.

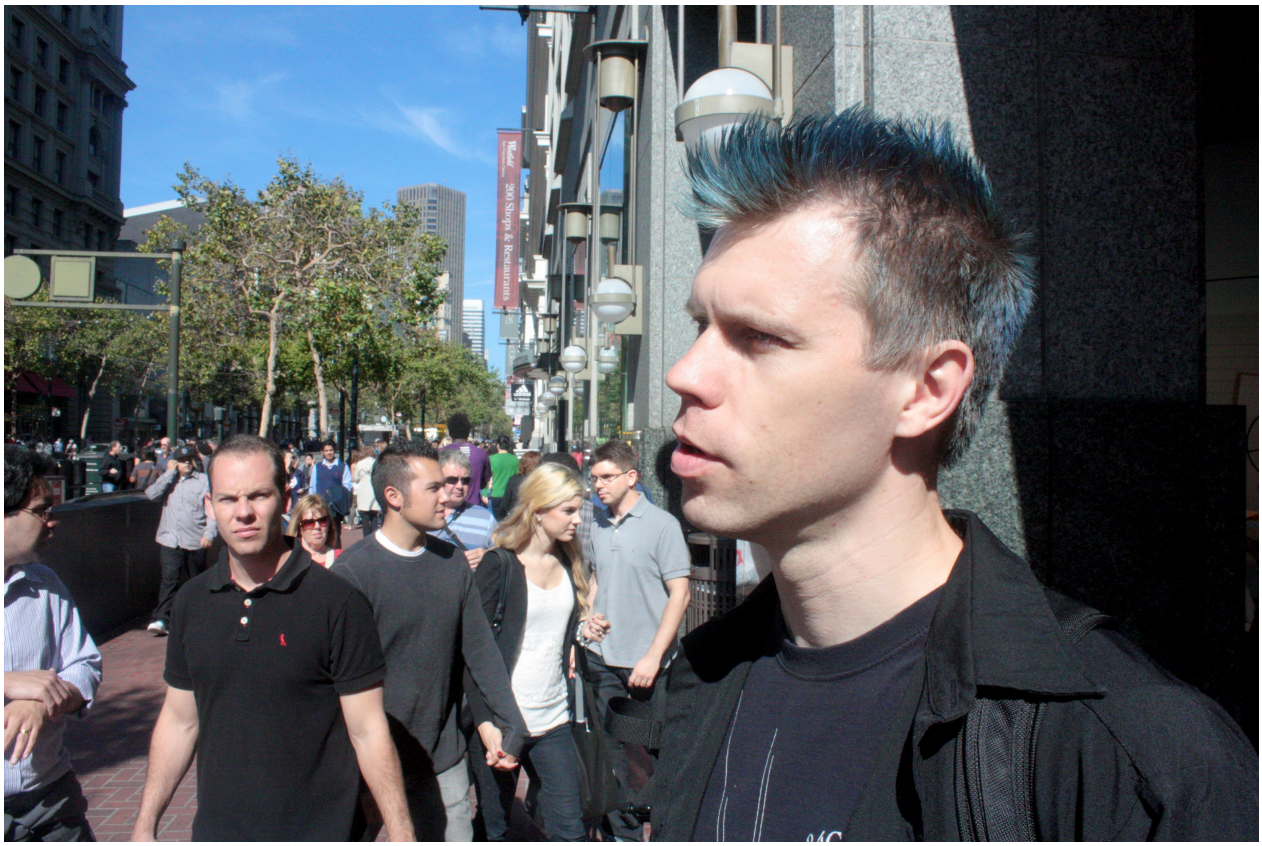


Figura 35. **:** Andy Isaacson.

Me llevó a un hotel a dos cuadras de la plaza Mint. No encontrábamos la entrada y me dijo: "Así construyen los hoteles elegantes para que a la gente que va pasando casualmente no se le ocurra entrar". Subimos al segundo piso del hotel, donde había una terraza al aire libre. Nos sentamos en una de las mesas del lugar y comenzamos una entrevista de casi tres horas.

Biografía. Andy Isaacson nació en 1977 y creció en Sebeka, Minnesota. Es un pueblito alejado de las grandes ciudades que en el censo de 2010 tenía 711 habitantes. Sus padres lo criaron en la religión Apostólico-Luterana de Finlandia, una secta que posee una comunidad relativamente grande en los Estados Unidos. Su padre trabajaba como maestro de educación primaria para niños con problemas de aprendizaje. Su primer contacto con la tecnología se dio precisamente en la escuela donde trabajaba su padre y en la que él estudiaba, donde había una computadora Apple II.

En casa, Andy encontró un manual de programación de BASIC para microcomputado-

ras que era de su padre: "Probablemente leí ese libro una docena, o veinte, o treinta veces cuando tenía como siete u ocho años por que era tan interesante para mí".[61] Sin embargo, no se interesó especialmente en la computadora de la escuela. A veces utilizaba algunos juegos como Oregon Trail, y programó algunas pequeñas bromas y juegos en BASIC. Me ejemplificó un pequeño programa de dos líneas que podría haber hecho a esa edad:

```
10 PRINT "Hello!"
20 GOTO 10
```

Este pequeño programa en BASIC imprimiría "Hello!" en el monitor hasta que el usuario interrumpiera la ejecución del programa.[62] "Pensé que eso era genial, pero no hizo clic para mi". Andy siente que su historia no es tan impresionante debido a que tiene conocidos que cuentan anécdotas muy diferentes:

> Tengo amigos que tienen historias sobre sus inicios en la programación. Llegaron al mismo punto al que yo llegué, pero ellos siguieron adelante, y siguieron... Y dos semanas después habían escrito un programa propio, y estaban ejecutándolo, y tenía cien lineas de extensión y... Hice algo de eso y me di una idea, pero nunca pude enfocarme los suficiente como para construir algo interesante a esa edad. [...] Nat Friedman[63] escribe en su blog sobre cómo utilizaba su primera Apple II, escribía código durante horas y horas y horas. Escribiendo código y ejecutándolo en el monitor y así. Hice un poco de eso. Recuerdo haber cambiado los colores de la pantalla de la Apple II, pero nunca escribí nada verdaderamente emocionante. Comencé un juego... Zelda, de hecho. Trabajé en GWBASIC en una PC compatible con IBM en octavo grado o algo así. Trabajé en ello durante un tiempo, obtuve algunos resultados, hacía algunas cosas pero nunca fue jugable.

---

[61] La entrevista con Andy se llevó a cabo en inglés. Todas las citas que aparecen son traducciones mías.

[62] Este programa es idéntico al primero que intentó Linus Torvalds cuando aprendía BASIC. En su autobiografía cuenta que es el primer ejemplo que viene en los manuales de ese tipo de computadoras caseras (Torvalds y Diamond, 2001, pág. 8).

[63] Nathaniel Friedman es un programador que fundó la empresa Ximian junto con Miguel de Icaza. Esta empresa desarrollaba soluciones de software libre de escritorio en la plataforma Gnome.

Figura 36. **:** La computadora Apple IIe.

Andy continuó jugando en la computadora y con el tiempo se interesó más en ella. Otro de los maestros de la escuela tenía una Macintosh SE y en ocasiones la llevaba a la escuela. Debido al peso de esa computadora (más de diez kilos). Andy se ofrecía a ayudarle con tal de usarla un poco. Pero todavía no aprendía a programar. Sus primeras clases se dieron hasta que se encontraba en *middle school* (educación secundaria), donde también programó archivos por lotes de MS-DOS.

Ya en la preparatoria Andy sabía más sobre computadoras que el resto de los estudiantes. Junto con sus amigos tomaron una clase optativa de programación y otra de mecanografía. Durante este periodo entró a concursos de programación donde participaban aproximadamente 50,000 estudiantes de todo el país. Superó la competencia regional, luego la estatal y llegó hasta el nacional en su primer intento. Para ir, los estudiantes necesitaban recaudar fon-

dos ya que la escuela no poseía los recursos para enviarlos a todos los participantes. Vendió chocolates y otras cosas para financiar el viaje.

La competencia se basaba en problemas de programación que debían resolverse en cierta cantidad de tiempo. Andy admite: "La competencia no era muy buena. La gente contra la que competía no era muy hábil. [...] Supongo que a la gente que podría ser buena no le importaba o ni siquiera sabían que la competencia se llevaba a cabo. O sea, es la preparatoria, ¿qué tan buena puede ser?". La programación ya llamaba poderosamente su atención y "era como me definía a mi mismo, como lo que hacía".

Uno de sus amigos era más hábil con la tecnología que él y también era aficionado a los videojuegos. Jugaba muchos juegos piratas, "crackeados" o copiados ilegalmente. Andy supo que ese no era el camino que tomaría:

> La justificación original era que mis padres se iban a enojar mucho conmigo si se enteraban que estaba haciendo algo así. Pero creo que... No sé si lo formulé así a una edad tan temprana, pero creo que tenía la noción de que las estructuras que construimos como sociedad no siempre son justas, no siempre son correctas, pero tienen una razón. La gente las crea por diferentes razones. Y trabajar dentro de las estructuras causa mayor bien que destruirlas.

Aún así, admite haber jugado con algún software pirata, especialmente el juego "Karateka" que tenía una protección contra copia: "Por supuesto alguien obtuvo una copia 'crackeada' de alguien de otra escuela, hizo una copia para su amigo e hizo una copia para otro amigo". Después de clase había un periodo en el laboratorio de computación donde los estudiantes trabajaban en lo que quisieran, siempre y cuando se comportaran. Durante un tiempo se llenaba de estudiantes que jugaban Karateka hasta que cansaban al laboratorista y ordenaba a todos trabajar en procesadores de texto.

Sebeka no tenía un proveedor de Internet local cuando Andy era niño. Se conectaba a través de un servicio de larga distancia llamado Delphi. La compañía telefónica de su ciudad decidió capacitar a dos personas para instalar la infraestructura necesaria para ofrecer servicio

de Internet. Un tío de Andy trabajaba para la compañía e incluso lo dejó pasar al lugar donde se encontraban el equipo electrónico. Gracias a su tío Andy se enteró de que pronto se ofrecería el servicio de Internet.

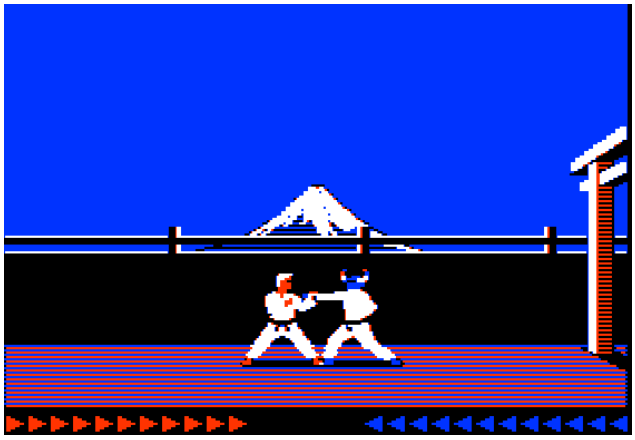


Figura 37. **:** El videojuego Karateka para la computadora Apple II.

Sabiendo esto y con las vacaciones decembrinas encima, Andy se propuso obtener el servicio lo más pronto posible. Se dirigió a las oficinas de la compañía telefónica a las 3 pm del último día antes de que salieran de vacaciones:

> Me aproximé al escritorio y dije: "¡Hola! Soy fulanito de tal". Es un pueblo pequeño. "Mi papá trabaja en la escuela, conocen a mi padre, conocen a mi tío, trabaja para ustedes. Sé que pronto ofrecerán el servicio de Internet, sé que todavía no lo tienen listo, pero, ¿podrían darme el teléfono, un nombre de usuario y una contraseña? Probablemente lo tengan listo después de año nuevo, el dos de enero o algo así, ¿podrían configurármelo?" Y las dos mujeres en el escritorio pensaron que... No querían reírse en mi cara, pero pensaron que esto era graciosísimo: "Aquí está este niño que acaba de entrar exigiendo un servicio que ni siquiera tenemos". Y dijeron: "Bueno, creo que podemos hacer los trámites necesarios". [...] Escribieron todo lo que necesitaba y ya iba de salida cuando me dijeron: "Espera, antes de irte, ¿podemos tomarte una foto?" Porque el pueblo tenía un periódico semanal, se publicaba cada semana y si enviabas cualquier cosa que fuera remotamente noticiosa la publicaban. Así que tomaron mi foto. Estaba de pie sonriendo como idiota porque tengo 17 años, ¡y voy a usar Internet esta noche! ¡Puedo sentirlo!

Su fotografía apareció en el periódico como la primera promoción para el servicio de Internet.

Andy llegó a su casa a configurar el servicio y había sacado un libro de la biblioteca del condado para auxiliarse. Pero había un gran problema: Olvidaron darle la dirección del DNS.[64] Decepcionado y molesto, Andy leyó y releyó el libro para encontrar una pista que lo ayudara. En una fotografía aparecía la configuración de Internet del autor.

> Y había una fotografía de su Mac con el mismo software que yo, de hecho, en Edinburgh, con la dirección del DNS de su universidad. Pensé: "Esto no va a funcionar, ¿o sí?" Lo escribí en mi programa y cargó... Intenté con un nombre y funcionó. Me puse como [...], ¡eureka! ¡El *hack* funcionó! Y estaba muy emocionado por esto y fui a decirle a mi papá. El sabía algo de tecnología. Conocía las computadoras, sabía lo que estaba haciendo, más o menos, y realmente se enojó conmigo. [...] Por robar servicio: "¡No puedes usar ese servicio! ¡Es de ellos!"

Intentó calmar a su padre explicándole que no se encontraba "robando" los servicios de otra universidad, simplemente hacía peticiones como cualquiera en Internet. Aún así no quedó del todo satisfecho y pidió a su hijo que dejara de usar ese servicio en cuanto consiguiera el número de DNS de su proveedor. Andy reflexionó un poco acerca de este evento:

> A mis padres siempre les preocupaba que me convirtiera en un bueno para nada. En parte por ello no elegí el camino *black hat*. En parte porque mis padres insistieron en ello, pero también porque hicieron un buen trabajo al enseñarme, o gritarme lo suficiente para asegurarse de que: "Sigue las reglas, es más fácil de ese modo, la gente no te molesta y puedes hacer lo que quieres".

En su casa Andy nunca tuvo televisión. La lectura siempre fue un pasatiempo para él y sus hermanos. Su introducción al *hacking* llegó a través de libros que encontró en la biblioteca local. Para él, era claro que la palabra *hacker* significaba el Tech Model Railroad Club del MIT

[64] DNS son las siglas de Domain Name Server. Es un dispositivo que traduce los nombres de dominio de Internet (por ejemplo www.google.com) a direcciones de Internet (por ejemplo 74.125.159.105). Sin este servicio, los usuarios deben conocer las direcciones de Internet de todos los sitios a los que quieren entrar y la mayor parte de los vínculos dejan de funcionar. Aunque técnicamente se encontraría conectado a Internet, la funcionalidad estaría severamente limitada.

y "otras cosas buenas", pero también era un término utilizado por los medios para hablar de los "crackers". Para Andy se trataba de una formulación como: "Aquí está la palabra, significa esto, tú la estás usando mal".

> Lo tenía muy claro... El *hacker* es el tipo que pone a trabajar a la computadora. Y las cosas que las computadoras hacen son buenas o malas. [...] Tienes tu acceso hábil. [Hace una separación con la mano]. Hábil, inhábil. [Hace una división imaginaria con la mano en la sección "Hábil"] Y tu acceso bueno: ¿Eres una buena persona o una mala persona? Y la idea de que puedes burlar sistemas sin que sea necesariamente malo, aunque la gente que no sabe lo que pasa puede sentir que burlarte del sistema es inherentemente malo. El problema con el DNS es un buen ejemplo de ello. Evité un problema usando un sistema de forma que... Usando un pedazo de información en un sistema de manera inesperada, pero sabía... Había evaluado, o al menos pensado en el impacto de lo que estaba haciendo y evité seguir haciéndolo.

Los padres de Andy eran muy estrictos en muchos sentidos, pero también sentían que no salía lo suficiente de casa y que pasaba la mayor parte del tiempo en la computadora. Instituyeron horarios para restringir el tiempo que él y sus hermanos pasaban frente a la máquina. Durante ciertos periodos debían mantener la computadora apagada y dedicarse a alguna otra cosa. Esto incluía la lectura, incluso leer un libro sobre computadoras, pero no usar una.

Encontraron una manera de sacarle la vuelta a esa regla. Durante un remate de equipo viejo de una empresa de la localidad, había algunos dispositivos electrónicos de diez años de antigüedad. Entre ellos se encontraban algunas terminales seriales. Estos dispositivos no son computadoras aunque lo parecen, y pueden comunicarse una con otra a través de un módem. Andy ofreció cinco dólares por las terminales y como nadie más las quería se la llevó a casa. En ocasiones, durante los periodos cuando no usaba computadoras, Andy y su hermano conectaban las dos terminales para escribir a distancia.

La afición por la lectura llevó a Andy a entrar a concursos de lectura aparte de las

competencias de programación. Se asignaba una meta mensual de minutos de lectura y él leía a tres o cuatro veces la velocidad de sus compañeros, además se fijaba metas de tiempo mucho más altas.

> Una sesión de una hora representaba muchas más palabras para mi que para ellos. Pero no podían... Contar palabras es difícil. Así que [los maestros de la escuela] no se molestaban en contar palabras y sólo contaban minutos. [...] Durante un mes [mis compañeros] tendrían como meta de lectura... 300 minutos o algo así, no mucho... Si. 600 minutos, veinte minutos de lectura por noche, porque estaban ocupados viendo televisión o lo que sea. Mis metas de lectura eran 3000 minutos o 4000 minutos o 6000 minutos al mes. Mis compañeros de clase no me creían que realmente estaba haciéndolo.

También participó en la carrera campo traviesa y en concursos de conocimiento general, pero en los deportes no fue tan bueno como en la programación. Así que cuando se acercaba el final de la preparatoria y Andy se enfrentó a la elección de una carrera, era obvio que sería de programación e informática. Tomaba cursos optativos y entraba a concursos de programación. Los adultos que lo veían utilizando la computadora le decían: "Wow, ¡vas a tener una buena carrera programando computadoras!"

Al considerar universidades pensó en solicitar admisión en tres de ellas, pero terminó solicitando solo en una por que sabía que sería admitido. "Había una mejor escuela en la cual debí haber intentado entrar... Comencé a llenar el formato pero no lo envié por que estaba intimidado por el costo, la distancia con respecto a todos mis familiares y el calibre que esperaba de la competencia. Así que nunca envié el formato, nunca me molesté". La escuela era Carnegie Mellon, la cual es sumamente reconocida por sus áreas tecnológicas y también por su departamento de ciencias computacionales. En retrospectiva se encuentra satisfecho con el rumbo que tomaron las cosas:

> Ahora conozco mucha gente que estudió en Carnegie Mellon y... Tengo la impresión de que tendría una experiencia mucho menos positiva. No estoy seguro,

> nunca he estado en una clase de Carnegie Mellon para ver, pero la experiencia que tuve en Michigan Tech fue verdaderamente positiva para mi.

El laboratorio de Michigan Tech estaba basado en UNIX y el programa de ciencias computacionales era muy riguroso. Uno de sus proyectos, por ejemplo, era escribir un analizador léxico de C en diez semanas, crear una unidad aritmética-lógica utilizando un simulador de compuertas lógicas. "Era mucho trabajo pero era increíblemente satisfactorio".

Otra de los motivos por los cuales se inclinó por esta universidad fue la religión. Cerca de ahí había una iglesia apostólico-luterana y en ese momento de su vida todavía deseaba ir, aunque cada vez se distanciaba más. Pertenecer a una religión minoritaria marcó a Andy como un *outsider*, alguien que nunca se integró al resto de la gente.

> No miraba televisión, no iba a los bailes de la escuela, no tomaba alcohol... También me mantuve fuera de los grupos sociales de la escuela porque... No sé por qué. Creo que era demasiado inteligente y no quería fingir ser tonto. Tenía amigos, nunca fui acosado mucho pero nunca fui incluido al grado de sentirme cómodo. Nunca aprendí como salir con muchachas en la iglesia, mis papás no me animaban a hacerlo. Fue muy tonto de su parte. Si querían que me quedara en la iglesia me debieron dar chicas. ¡Hubiera funcionado! ¡Oh, vaya que sí! [...] Pero todas estas experiencias me dieron mucho conocimiento sobre qué es ser un *outsider*, pero no me dieron las experiencias traumáticas de muchas de las personas que lo son. [...] Nada terrible me sucedió, pero tampoco fue una experiencia enriquecedora. Fui capaz de decir: "El sistema está jodido. Hagamos otra cosa".

En la universidad terminó distanciándose de la religión completamente. Pero el sentimiento de incomodidad ante los demás y la poca motivación para integrarse continuaron. Andy no sentía la necesidad de formar parte de ningún grupo y por la mayor parte continuaba sus actividades individualmente.

A todos los estudiantes de la universidad se les proporciona una cuenta gratuita de UNIX. Como en el caso del proveedor de Internet de Sebeka, Andy se acercó a la universidad

para solicitarla antes de comenzar clases. Pero esta vez la estrategia no funcionó. En cuanto entró a la universidad pasaba la mayor parte del tiempo dentro del laboratorio de computación, explorando el sistema. Además de las 40 horas a la semana que invertía en cosas de la escuela, también pasaba más tiempo, de 40 a 80 horas a la semana, trabajando en sus propios proyectos personales.

Entre estos proyectos se encontraba un proyecto de *ray-tracing*[65] distribuido que coordinaba a través de un script de bash. En un momento se encontró con una dificultad que no podía superar, así que preguntó a alguno de los estudiantes de grado avanzado, el cual quedó impresionado por el trabajo y llamó a sus amigos para que también lo vieran. Sobre la motivación para programar el script, Andy explica:

> Porque hay un sistema aquí y estoy tratando de aprender cómo usarlo y ponerlo a hacer lo que yo quiero, y lo que yo quería era que las computadoras trabajaran juntas, y todavía no tengo las habilidades para usar los sistemas que otra gente ha creado para resolver esto, así que voy por mi propio camino, usando los bloques de construcción que tengo disponibles. [...] Si en ese momento alguien me hubiera preguntado: "¿Eres un *hacker*?" Le hubiera sonreído y preguntado de vuelta: "Bueno, ¿qué quieres decir con eso?" En el sentido que utilizan los medios, no lo soy. No soy un intruso en los bancos, no tengo ningún... No hay ningún motivo para que yo haga eso, por que el posible costo de hacerlo en términos de cárcel o de imposibilidad de conseguir empleo es mucho mayor que los posibles beneficios. [...] Era obvio para mí que era mucho más fácil seguir las reglas, mientras no odie demasiado seguirlas.

Tomó la mayor cantidad posible de clases sobre ciencia computacional inmediatamente. Buscó integrarse a la investigación sobre el área, un departamento de ciencias computacionales completamente teórico y preocupado únicamente por las matemáticas y teoría de grafos.

[65] Ray-tracing es una técnica computacional para generar imágenes tridimensionales. Estos procesos normalmente requieren grandes cantidades de poder de procesamiento, por lo que muchas veces se hacen a través de sistemas distribuidos.

Como deseaba algo más aplicado, fue al departamento de ingeniería eléctrica donde tenían una investigación en curso sobre sistemas de archivos, pero los profesores se encontraban de año sabático. Sólo uno de ellos permanecía en la universidad: "Había un profesor que todavía hacía soporte para sistema de archivos, hablamos y pensé: '¡Es un idiota! Usted no tiene idea de lo que está hablando.' Por supuesto que no le dije esto en su cara, pero le dije: 'Muchas gracias por explicarme su trabajo. Ya me voy'".

Andy prefirió colaborar en el departamento de física, como ayudante de investigación sobre sistemas distribuidos en UNIX. También trabajaba medio tiempo como programador de C mejorando un sistema de monitoreo para una empresa. El trabajo era muy complicado para un estudiante de los primeros semestres, pero logró sacarlo adelante. En retrospectiva siente que no lo resolvió óptimamente. Posteriormente aprendería mejores técnicas para escribir código fácilmente actualizable, óptimo y modificable.

En una de sus clases descubrió un error en uno de los programas utilizados por el profesor para que los estudiantes enviaran su tarea. Lo logró utilizando una técnica llamada "ingeniería inversa", que consiste en tomar un código máquina y transformarlo en código fuente legible por seres humanos. No utilizó el error para su beneficio, solamente informó al profesor, el cual aún así se alteró. Cuestionó los motivos y la autoridad de Andy para revisar el código. Al final, a pesar de todo, le agradeció y corrigió el error.

También exploró la comunidad *hacker* de la localidad. Sabía de la existencia de DEF-CON y otras convenciones. Asistió a algunas reuniones 2600, pero no regresó con experiencias muy positivas. Andy denominó a la mayoría de los asistentes como *griefers*, un término utilizado en los videojuegos en línea para designar a jugadores que se dedican a entorpecer el juego para los demás.

> Creo que fui a una reunión 2600, conocía la historia de [Kevin] Mitnick y sabía que había muchas oportunidades para... Y había leído un par de revistas 2600 para este punto. Sabía que había muchas oportunidades para ser capturado por la policía. Y era demasiado fácil involucrarse periféricamente y terminar más

> involucrado por accidente. Yo no quería nada de eso. Y de nuevo pensé: si sigo las reglas puedo obtener un buen trabajo y hacer lo que quiera por mi cuenta. No tengo que seguir estas reglas foráneas. Siempre ha sido un acto de malabarismo para mi, siempre he... Ha sido una larga serie de situaciones donde soy el *outsider*.

En Michigan Tech también obtuvo su primera experiencia en lo que llama "do-cracy", o el poder en manos de los que hacen. Los laboratorios RSA[66] tenían un concurso en 1998 donde retaban a todos al público en general a descifrar un mensaje cifrado. El proceso es largo y necesita mucho poder computacional para resolverse. El proceso se acelera si hay muchas máquinas trabajando en el mismo problema. En el momento en se resuelve, se considera que esa tecnología criptográfica no es lo suficientemente segura y se migra a otra con mayor poder.

El equipo que logra dar con la solución correcta se hace acreedor a un premio en efectivo. Dentro del laboratorio de cómputo alguien propuso la idea de ordenar a todas las máquinas a trabajar en el problema. Esto no debería suceder cuando alguien utilizara la computadora, de lo contrario los usuarios notarían una severa disminución del rendimiento del sistema.

> Tuvimos una reunión al respecto y alguien sugirió que implementáramos [un programa que ejecutara el código cuando nadie estuviera en la máquina] y mucha gente rechazó la idea diciendo: "Es mucho trabajo. Tienes que escribir el código, y alguien tiene que ejecutarlo y darle soporte. Revisar que nadie esté usando la máquina va a alentar el sistema". Lo cual era cierto, si lo haces mal.

Andy decidió encargarse del problema y escribió un programa que revisara si algún usuario estaba en la máquina con muy poco impacto para el desempeño del sistema. Después de que el laboratorista revisó el programa, se implementó y no hubo quejas por parte de los usuarios. Como resultado, Michigan Tech contribuyó significativamente a este reto: "Esa fue mi primera experiencia con 'solo hazlo.' Escribe el código. Una vez que el código exista podemos discutir

[66] Definir la RSA.

al respecto. O ejecutarlo. No puedes ejecutar código que no existe. Puedes discutir todo el tiempo que quieras, pero el código existente es un argumento muy fuerte".

Una experiencia hizo reconsiderar a Andy su estancia en la universidad. Tomó una materia con un profesor sobre el cual se encontraba optimista. Como proyecto final debían construir un sistema en parejas. Andy fue posponiendo este proyecto debido a que se encontraba más enfocado a su empleo. Se había acostumbrado a un salario real y no a uno de estudiante. Además, generalmente no tenía mucha motivación para las tareas de la escuela, porque "ya entendía" los temas y sentía que le quitaban mucho tiempo. El semestre siguió su curso, y su compañero y él pronto comprendieron que no tendrían tiempo para terminarlo:

> El profesor decidió que era su culpa por no haber monitoreado nuestro progreso, así que nos dio una nota aprobatoria falsa, como una B con una especie de: "No vamos a hablar de esto, ¿verdad?" Eso me complicó el seguir trabajando más con ese profesor. Y no me agradaban las ideas que tenía sobre el área en la que estábamos trabajando. Teníamos ideas diferentes sobre cómo debían funcionar las cosas.

Esta última experiencia convenció a Andy de que no obtenía lo que deseaba de la universidad y buscó empleo en otra parte. Contactó a Larry McVoy, creador de la empresa BitMover que desarrollaba el programa BitKeeper. Mediante este programa, los desarrolladores del núcleo de Linux controlaban y manejaban los cambios en el código. Andy se interesó en el proyecto y decidió trabajar en él. Ya sabía cómo funcionaba el sistema gracias a sus aportes al núcleo de Linux.

Continuó estudiando su último semestre en la escuela mientras también trabajaba remotamente con Larry, pero después de la experiencia en la asignatura donde lo aprobaron injustamente, decidió salir de la universidad. Su plan era mudarse a San Francisco para trabajar tiempo completo en BitMover. Andy ya había pensado en mudarse al área de la bahía debido a la gran cantidad de empresas tecnológicas y la comunidad que existe en el área.

Ya había visitado San Francisco un par de veces. El área de la bahía tiene un

> gran caché en el campo tecnológico. Todo mundo sabe que Silicon Valley es el lugar donde las cosas suceden. Incluso en los noventa apareció un artículo en Minneapolis que decía: "Vamos a hacer la Pradera del Silicon".[67] Llegué aquí y fue obvio que esto no era transportable. No va a ser lo mismo. [...] El nivel de conectividad y la infraestructura y profundidad que existe aquí no es reproducible en ninguna otra parte. Así que, posiblemente para el tiempo cuando me mudé a Minneapolis estaba pensando que me gustaría estar en California en algún momento. Estar en Silicon Valley en algún momento. Era muy... Incluso en aquella época era muy consciente de las ciudades. Pensaba que las ciudades eran geniales. Crecí en una granja en las afueras de un pueblo con 600 habitantes, a 15 millas de un pueblo de 3000 personas, a 50 millas de una ciudad de 18,000 personas. Si queríamos hacer compras de verdad debíamos manejar tres horas hacia un pueblo que tenía 60,000 personas y algunas tiendas grandes.

Larry ofreció pagar todos los gastos de mudanza de Andy para que trabajara de tiempo completo para él. Sin embargo, antes de que esto sucediera, Larry reconsideró y comentó a Andy que quizá no era una buena idea después de todo. Pero como ya había pagado por el boleto de avión y la mudanza, le ofreció un viaje a San Francisco con la estancia pagada para se se entrevistara con otras empresas. Durante este breve periodo de desempleo, tuvo tiempo de establecer contactos en la localidad, aunque no fue fácil. Andy asistió a su primera convención *hacker*, CodeCon, la cual describe como una experiencia "sumamente formativa".

> Nunca me conecté mucho a la comunidad. Es como un proceso iterativo. Ya conocía algunas personas por Internet, y ahora vivía en la misma ciudad que ellos así que podía ir a reuniones o lo que fuera. Conocí un par de personas en CodeCon. No me integré inmediatamente a las cosas de la comunidad. Tenía algunos amigos que también se mudaron para acá. [...] Fue un proceso orgánico muy lento. Me

[67] Cuando Andy me comentó esto no pude dejar de pensar en los proyectos, actualmente detenidos, de construir "Silicon Border" en la frontera entre California y Baja California.

> sentí verdaderamente aislado por un par de años. Lentamente conocer gente y establecer conexiones me llevó mucho tiempo. No es algo que encuentre fácil de hacer. Es satisfactorio a la larga, pero desgastante a corto plazo. Soy mucho mejor en ello hoy que hace cinco años.

Andy conoció a Jake Appelbaum a través de Internet y luego personalmente cuando regresó de Iraq en 2005. Se hicieron amigos y viajaron a una convención *hacker* en Europa, el Chaos Communication Congress número 22 (22c3) organizado por el grupo Chaos Computer Club. "Fue mi primera conferencia *hacker* europea, mi primera exposición a la escena *hacker* europea y fue súper emocionante y energizante".

Junto con sus amigos pensaron en iniciar algo parecido en San Francisco y la primera idea que tuvieron fue crear un *hackerspace*, una idea que conocieron en Europa. Reunieron gente, Mitch Altman, Jake y otros también se unieron a la causa. Se reunían una vez por semana para discutir lo que se podría necesitar, de dónde obtendrían los fondos así como los trámites legales para establecerse formalmente.

> Si quieres hacer fácilmente una organización no lucrativa, debes seguir el mismo molde que todo mundo utiliza. Pero ese molde dice cosas sobre cómo debes administrar tu corporación, y eran cosas que nosotros no queríamos. Nosotros queríamos que fuera un esfuerzo colaborativo, queríamos estar basados en el consenso en vez de un sistema por votos. No queríamos que hubiera oficiales que tuvieran más poder que otra gente. [...] Así que tratamos de escribir reglas que implementaran características que nosotros queríamos pero que a la vez se apegaran a los requerimientos del estado. Pero las leyes del estado están escritas suponiendo... Son como: "Tu corporación va a tener un presidente". Y nosotros nos quedamos como: "No, no queremos un presidente".

Estas discusiones se prolongaron durante 18 meses. Noisebridge, el nombre con el que bautizaron ese lugar que todavía no existía, se dio a conocer en el área. Más personas se interesaron

Figura 38. **:** Una de las primeras fotografías en el primer local de Noisebridge. A la izquierda se encuentra Mitch Altman, al centro Jake Appelbaum y a la izquierda de Jake se encuentra Andy. La fotografía se encuentra en la cuenta de Flickr de Jake bajo una licencia Creative Commons. No pude identificar a las otras dos personas.

en el proyecto y la noticia llegó a oídos de Enki, un *hacker* austriaco que se encontraba de visita en San Francisco. En una ocasión asistió a una de las reuniones.

> Su inglés es fantástico, su acento es muy... Es como el de Schwarzenegger. No suena como él en absoluto, pero es el mismo nivel de acento. Perfectamente comprensible pero muy reconocible. Y tuvimos una junta y estábamos repitiendo las mismas cosas de siempre una y otra vez, y él nos observa y luego dice... Se pone de pie y nos dice: "¿Puedo decir algunas cosas?" Y nos dio un discurso: "¡Simplemente háganlo! Vayan y hagan que suceda. Todo esto quedará atrás cuando tengan un espacio". Y nos quedamos: "¡Eso es genial!" Era como... Nos animó mucho escuchar eso.

> Ok, tenemos todos estos problemas pero no estamos totalmente mal, vamos bien.

La abogada que los asesoraba metió los papeles para formar la organización no lucrativa casi como vienen de molde, con la idea de cambiarlos posteriormente. Así que a los pocos días buscaron un espacio disponible para renta. Lo encontraron en la calle Mission, cerca del centro de San Francisco. La primera ubicación de Noisebridge era muy pequeña, pronto se mudaron.

> Los primeros meses del primer espacio de Noisebridge fueron un tiempo muy interesante por que construimos una comunidad bastante grande que estaba interesada en el lugar. Y repentinamente ya existía, así que la gente tuvo que aprender a lidiar con su existencia. ¿Para qué es el espacio? ¿Qué podemos hacer aquí? ¿Para qué no es el espacio? ¿Qué tipo de cosas no podemos hacer aquí? Se sentía sobrecogedor cuando iniciamos. Cuando rentamos el lugar nos quedamos: "Guau, ¡tenemos espacio! Tenemos algo". Pero rápidamente fue obvio que no sería suficiente. Se llenó de cosas en menos de un mes.

Consiguieron mobiliario donde les fue posible. Por ejemplo, una empresa remató algunos estantes para libros y alguien rentó un *pick-up* para recogerlos. Organizaron el hardware y aceptaron donaciones. Andy comenta que la gran lección que aprendió en esos primeros meses fue el permitir que las cosas sucedan sin mucha intervención permite que surjan buenas ideas.

Como ejemplo, Andy platica sobre una noche en donde se encontraban metiendo sillas donadas al *hackerspace*. Mientras las cargaban, se encontraron un horno de microondas en la acera. Había sido abandonado por algún restaurante y no servía. Lo metieron y permaneció días en la pequeña cocineta del lugar, descompuesto. Durante la fiesta de inauguración se presentaron muchas personas, ya que el evento fue anunciado en algunos blogs.

> Y dos tipos de apariencia sospechosa a quienes nunca había visto y que no he vuelto a ver, me dijeron: "El microondas no sirve". Les respondí: "Sí, lo sé. Arréglenlo". Y me dicen: "Danos un desarmador". Así que estaban tomando cerveza

> y trabajando en electrónicos de alto voltaje. [...] Le quitaron la tapa al microondas y estaban tomando cerveza mientras lo inspeccionaban y Shannon me dice: "¿Quienes son esos tipos?" Y le respondí: "No lo sé, sólo pidieron un desarmador. Quieren electrocutarse y no los voy a detener". Diez minutos después lo habían echado a andar, lo reensamblaron y cocinaron lo que querían cocinar. Se fueron y no los he vuelto a ver.

Recientemente en Noisebridge surgió una discusión con respecto a la apertura y el acceso. ¿Son demasiado abiertos? ¿Deberían restringir el acceso al lugar? La discusión se inició debido a incidentes que incluyen pequeños robos y confrontaciones. El cuatro de febrero de 2012 por la madrugada, un individuo entró y amenazó a los presentes. Alegando que tenía una pistola debajo de la sudadera exigió a todos que se retiraran o les dispararía. Varios de los presentes lograron someterlo y expulsarlo. Lo detuvieron hasta que llegó la policía y presentaron cargos en su contra.

De manera similar, por las mismas fechas, el *hackerspace* se pobló con simpatizantes del movimiento Occupy San Francisco. Este movimiento protesta en contra de la desigualdad económica y social. Una de las formas de protesta es acampar frente a edificios de importantes instituciones económicas como la reserva federal. En este contexto surgió el movimiento Hackupy, donde *hackerspaces* desarrollan tecnología e infraestructura para ayudar al movimiento Occupy.[68] Como Noisebridge es participante del movimiento, muchos de los protestantes se convirtieron en asistentes regulares a las instalaciones del lugar.

Esto causó algunos problemas. Las quejas más comunes de otros miembros de Noisebridge es que los protestantes dormían en el lugar y no querían involucrarse en el *hacking*. A pesar de que Andy afirma que no hay reglas en el *hackerspace*, o casi no las hay, en la práctica existen bastantes principios informales. Uno de ellos es el de no dormir en el lugar. En el contrato de renta está prohibida la residencia. Pero aún así, algunas personas de Noisebridge piensan que la verdadera razón es que el *hackerspace* es un lugar donde se va a hacer cosas.

[68] En la página web http://hackerspaces.org/wiki/Hackupy se enlistan los objetivos del movimiento y la lista de *hackerspaces* que participan. Entre ellos, Noisebridge.

Figura 39. **:** Cartel visible en Noisebridge, donde se muestra el lema del lugar.

Figura 40. **:** En Noisebridge existen carteles como este, donde se afirma que dormir en el lugar es algo no-excelente.

La inactividad es muy mal vista.

En la lista de discusión se barajaron posibles alternativas para solucionar este tipo de problemas. Algunas de ellas involucran complejos sistemas de acceso para restringir la entrada a las personas indeseables. El 8 de febrero de 2012, en un correo en la lista de discusión uno de los miembros escribió un mensaje con una propuesta:

> Definición del problema: En lo que parece una serie creciente de conflictos, muchos de los miembros que pagamos y cooperamos a la comunidad nos estamos preocupando sobre nuestra propia seguridad. Además de esta preocupación fundamental, existe la preocupación más importante de que la cultura de Noisebridge

y la comunidad se están afectando.

Requisitos para resolver el problema:

Los miembros de noisebridge deben sentirse seguros en noisebridge.

Debemos preservar la cultura de noisebridge con un enfoque hacia la apertura universal y la aceptación de la gente en su misión de compartir el conocimiento y la experiencia.

Propuesta de solución:

Cerrar noisebridge a las 11 pm hora del pacífico cada día hábil y cerrar a la media noche los viernes y sábados.

Quiero decir universalmente cerrados. Sin exclusiones ni excepciones. Como la última llamada de un bar. Siempre puedes irte a casa pero no te puedes quedar aquí.

Resultados:

No más locos vagabundos acosando gente por la noche. No más gente dormida, sin casa y con cara de tonto, a quienes les gusta arruinar algo bueno para todos. No más borrachos que llegan a molestar gente. No más riesgo de tener un accidente de camino al metro / autobús / bicicleta / ala delta... etc. porque no hay luz de sol. No habrá impacto en la naturaleza abierta del espacio, cuando el espacio esté disponible todos serán bienvenidos.

¡TADA!

Misión cumplida.

Este mensaje, titulado "una modesta propuesta", desencadenó una extensa discusión y decenas de mensajes relacionados con el tema. Todos discutían la idea de tolerar que gente duerma dentro de Noisebridge y si se prohibiría el acceso a algunas personas.[69] A algunos tomaron a

[69]Cabe destacar que tanto el título como el contenido de este correo electrónico hacen eco en un ensayo de

bien la idea de cerrar en horarios definidos, y propusieron complejos sistemas de acceso que con frecuencia involucraban electrónica y matemáticas. Por ejemplo, una de las propuestas fue instalar una pantalla y un teclado numérico en la reja frontal. Un software presentaría una operación matemática compleja en la pantalla. Para ingresar, el interesado debería teclear la respuesta correcta.

Otras propuestas involucraban cámaras de seguridad, diferentes tipos de candados, llaves en conjunto de contraseñas numéricas, procedimientos de verificación visual, tarjetas de identidad y una larga lista. Las propuestas se han discutieron aún más, pero al momento de entrevistar a Andy por segunda vez, todavía no se llegaba al consenso. A pesar de las ideas restrictivas, Andy piensa que la discusión es acerca de un problema más grande: "No creo estar de acuerdo con algunas de las soluciones que la gente propone, pero la existencia del problema no es una sorpresa". ¿Se está formando una comunidad diferente dentro del *hackerspace*, que no es la misma que lo mantiene? ¿Cómo distinguirla?

> Es fácil crear una distinción superficial, o visual, como una especie de primera impresión entre la gente que... Entre comillas, "la gente que queremos" y "la gente que no queremos". [...] Si dos personas están en la discusión, tienden a coincidir sobre a quién quieren y a quién no quieren, cuando en realidad no están de acuerdo pero todavía no se han dado cuenta. Así que Alice y Bob[70] decidieron que quieren expulsar a alguien, y es obvio a quién quieren expulsar. [...] Pero resulta que el grupo que Alice considera malas personas coincide solo parcialmente con el grupo que Bob considera malas personas, pero no coinciden perfectamente. Y hay alguien que Bob quiere expulsar que Alice no quiere expulsar y viceversa.

Andy admite que, a pesar de favorecer la apertura y la carencia de reglas, muchas veces es

---

Jonathan Swift del mismo título. En este ensayo, el autor propone sarcásticamente una solución para la pobreza de Inglaterra y para evitar que crezcan niños en situaciones de calle. La solución es utilizar esos niños como alimento para el resto de la población (Swift, 1980).

[70] Estos nombres no representan verdaderos miembros de Noisebridge. Se utilizan frecuentemente en ejemplos de física, computación, o criptografía para ilustrar dos personas cualquiera. Son más fáciles de recordar que "persona A" y "persona B".

inevitable que exista gente grosera o desconsiderada que no aporta nada a la comunidad. No le gustan las ideas que se presentaron para cerrar el lugar o restringir el acceso pero lo único que puede decir es: "Yo no haría eso. No creo que deberíamos hacer eso". Lo mejor es mantener el mismo modelo y buscar que la gente de fuera se integre e interese en hacer cosas y contribuir al lugar.

A pesar de los problemas, Andy considera que Noisebridge crece y que tiene muy buena cobertura en los medios, incluyendo la British Broadcasting Company (BBC). Casi todas las personas relacionadas con tecnología de San Francisco han visitado Noisebridge o al menos saben que existe.

Mi experiencia al visitar fue sumamente positiva, a pesar de las rencillas cotidianas de muchos de los asistentes. Por supuesto que existen algunos personajes extravagantes. Conocí, por ejemplo, a una mujer que quería lanzarse como candidata a alcalde de San Francisco. Por su manera de ser, hablar y vestir, pensé por un momento que fue parte del movimiento hippie. Me enteré luego de que la disuadieron de sus intenciones, ya que buscaba convertir a Noisebridge en su casa de campaña.

También encontré un personaje peculiar llamado Mike. Los diferentes días que lo vi, vestía exactamente la misma ropa. Ya tendría casi sesenta años y recorría el *hackerspace* observando a los demás, buscando una oportunidad para compartir su trabajo. Como me mostré interesado, me platicó durante media hora sobre un dispositivo que construyó. Servía para aumentar la adherencia de pegamentos a superficies difíciles, y fue elaborado con materiales caseros.

Lo que llamó mi atención fueron sus expectativas sobre Noisebridge: según él, era tan solo un "espacio técnico". La gente usa y reinventa la tecnología, pero deseaba que se elevara hasta convertirse en un "espacio de ingeniería". Es decir, un lugar donde se desarrolle nueva tecnología.

"Pero aún sería mejor si se convirtiera en un espacio de investigación", continuó. En éstos, me explicó, no sólo se desarrollan tecnologías, también se genera nuevo conocimiento.

"¿Y qué hay en esos espacios?", pregunté yo. Mike sonrió complacido de que lo hubiese preguntado: "No tienes idea", respondió, "hay cosas maravillosas que ni siquiera imaginas que pueden existir".

Noisetor. Actualmente Andy coordina el proyecto Noisetor, fundado para proteger el anonimato de los usuarios de Internet. Se trata de un subproyecto de Tor, un proyecto global. Tor se apoya en tecnología de seguridad desarrollada por la marina de Estados Unidos. Sirve para ocultar la comunicación a través de redes informáticas, de manera que los usuarios que navegan por Internet no pueden rastrearse. Andy lo describe así:

> El proyecto Tor es una organización no lucrativa establecida en Estados Unidos que desarrolla el software y ayuda a administrar la red y que funcione sin problemas. La red Tor es usada por cientos de miles de personas para acceder a Internet a través de un canal anónimo y sin filtros. Así que si te encuentras en un país restrictivo, que filtra el Internet, usar Tor te podría dar acceso a sitios que el gobierno no quiere que veas.

Para que el sistema funcione, se requieren máquinas conectadas a Internet que puedan canalizar el tráfico. Estas máquinas se convierten en nodos a través de los cuales los usuarios pueden conectarse. Cuando un usuario dentro de Tor solicita una página web, la petición se va pasando aleatoriamente entre los diferentes nodos, y de forma cifrada. De esta forma, es imposible rastrear el origen y destino de las peticiones. Existen también los llamados "nodos de salida". Estos se encargan de llevar a cabo las peticiones, buscar la información en Internet y enviarla anónimamente al usuario que la pidió. El sitio web destino no sabe en realidad quién hizo la petición, cree que fue el nodo de salida.

Un usuario de China puede conectarse a la red Tor y acceder a páginas bloqueadas por el gobierno gracias a un nodo de salida ubicado en otro país. Uno ubicado en México, por ejemplo, obtendría la página y la enviaría de manera anónima al usuario. El sitio web del proyecto ejemplifica casos de uso: Ha servido a periodistas que buscan enviar artículos que

Figura 41. **:** Andy después de la primera entrevista, esperando el ascensor para bajar del hotel. La camiseta que trae puesta es la de 243C, el congreso *hacker* del cual me platicaba.

Figura 42. **:** Imagen de la página principal del proyecto Tor, con el vínculo de descarga. El logotipo es una cebolla por que el sistema añade múltiples capas a los paquetes de información para que no se rastreen.

denuncian al país en donde se encuentran. Lo utilizan trabajadores del gobierno que se encuentran en el extranjero y necesitan una manera segura para conectarse con las instituciones de su país.

Varias personas dentro de Noisebridge intentaron crear sus propios nodos y con el tiempo optaron por colaborar con el nuevo proyecto. No todos tenían las habilidades técnicas o el tiempo para mantener el sistema funcionando. Había otros *hackers* hábiles, pero sin el dinero requerido para establecer y mantener el nodo.

Los nodos de Tor son costosos ya que requieren una gran cantidad de ancho de banda para aportar significativamente. Andy pensó que si Noisebridge administrara un nodo, las personas que no colaboraran con trabajo podrían donar dinero a la causa. Existen otros nodos en Europa que operan de la misma forma, pero muchos norteamericanos se rehúsan a donar a ellos debido al tipo de cambio. Preferirían donar a un proyecto estadounidense. Otro incentivo es que Noisebridge es una organización no lucrativa, y por tanto las donaciones son

deducibles de impuestos en Estados Unidos. Así surgió el proyecto de Noisetor, pensando en que el interesado pudiera colaborar en medida de lo posible, con dinero o trabajo.

En la página principal del proyecto, `http://tor.noisebridge.net`, hay una brevísima explicación para los visitantes:

> ¿Cómo funciona?
>
> 1. Dona dinero a Noisebridge con los botones y vínculos de abajo.
>
> 2. Noisebridge configura nodos de Tor.
>
> 3. ¡Juntos hacemos de Internet un lugar más libre!

La propuesta fue revisada por algunos abogados de la Electronic Frontier Foundation para buscar posibles problemas legales. En agosto de 2011 se obtuvo el consenso por parte de Noisebridge para recibir donaciones y se lanzó una convocatoria en Internet. Mantener el nodo de Noisetor actualmente cuesta entre setecientos y ochocientos dólares al mes. La inversión inicial fue de más de mil quinientos dólares. Las donaciones llegaron desde varias partes del mundo y el nodo estuvo listo el 13 de septiembre del mismo año. Justo tres días antes de que yo conociera a Andy en el café Blue Bottle.

> Cada vez que hay un proyecto o un evento en Noisebridge... Si es más grande que un evento realmente pequeño, hay un miembro o un par de miembros que lo patrocinan. Se presentan y dicen: "Estoy trabajando en este proyecto y soy un miembro de Noisebridge. Si hay problemas yo seré responsable o me encargaré de resolverlos, o me aseguraré de que funcione con Noisebridge". Yo, Shannon, MCT, aestetix y otras personas fuimos los miembros de Noisebridge que nos presentamos para ello.

Andy se encargó principalmente de la parte técnica, junto con algunas personas que colaboraron en menor medida, como Shannon Lee. Configuró los servidores y estableció la página de donaciones.

> Ha funcionado muy bien. Hemos obtenido donaciones suficientes para mantener funcionando el nodo. Lo tenemos activo desde hace seis meses y tenemos donaciones que serán suficientes para entre nueve y diez meses. Así que tenemos un "colchón" en el banco que nos permitirá funcionar durante algunos meses más. [...] La mayor parte de las donaciones proviene de ingenieros en Silicon Valley. Gente interesada en proteger la seguridad y privacidad y que tiene cierta cantidad de recursos disponibles para gastar en ello. Algunos ingenieros de Google donaron cantidades significativas y un par de personas más que conozco del área de la Bahía. También hemos recibido donaciones de otras personas con domicilios estadounidenses a quienes no conozco. También algunas donaciones menores en cantidad de dólares, pero razonables en cantidad de donadores, provenientes de Europa: Alemania, España, Holanda. Alguien de Argentina, creo, basándome en su correo electrónico. Es un espectro bastante amplio.

Le pregunté si tenía alguna forma de saber si Noisetor estaba funcionando, o si aportaba algo a los usuarios finales. Andy no me dio información específica. Para empezar, los usuarios que buscan el anonimato no divulgarán a través de cual nodo de salida se conectan. Además, normalmente no prestan atención a ese tipo de información, sólo buscan la conectividad. A pesar de ello, afirma tener evidencia anecdótica de varios usuarios que le han escrito comentando que utilizan mucho ese nodo de salida.

La última vez que lo entrevisté, Andy me comentó que Noisetor proveía el 10 % del ancho de banda de los nodos de salida de Tor. Al revisar las estadísticas de la red descubrí que esta cifra cambia frecuentemente. En un periodo de algunas horas, el nodo principal (noiseexit01a) puede encontrarse entre los primeros diez, o incluso treinta lugares abajo. El volumen de tráfico es variable, así que esta clasificación depende también de la capacidad de otros nodos.

Un evento relacionado con Noisetor requirió la intervención de Andy. El 21 de febrero de 2012 dos agentes del FBI llegaron a Noisebridge interesados en el nodo de salida. El

Application Server Details	
Cache Last Updated (Local Server Time):	2012-02-28 18:20:42 MET
Last Update Cycle Processing Time (Seconds):	796
Current Cache Expire Time (Seconds):	300
Number of Routers In Cache:	2921
Number of Descriptors In Cache:	4179
Approximate Page Generation Time (Seconds):	0.1467

Router Name	Bandwidth (KB/s)	Uptime	Hostname	ORPort	DirPort	Bad Exit
gorz	21737	4 d	gorz.torservers.net [109.163.233.205]	443	80	
wau2	20979	4 d	wau2.torservers.net [109.163.233.201]	443	80	
rainbowwarrior	20963	6 d	rainbowwarrior.torservers.net [77.247.181.164]	443	80	
Kromyon1WH2A	17442	14 d	tor-relay-1wh2.kromyon.net [209.236.66.136]	443	None	
wau	17200	4 d	lh18189.limehost.ro [109.163.233.200]	443	80	
lumumba	15920	6 d	lumumba.torservers.net [77.247.181.163]	443	80	
politkovskaja	15756	6 d	politkovskaja.torservers.net [77.247.181.165]	443	80	
obelix	15081	56 d	obelix.mdnx.net [135.196.0.10]	9090	9030	
politkovskaja2	14789	6 d	politkovskaja.torservers.net [77.247.181.165]	110	143	
Celestra	14270	110 d	75.126.182.109-static.reverse.softlayer.com [75.126.182.109]	443	9030	
noiseexit01a	13767	17 d	exit-01a.noisetor.net [173.254.216.66]	443	80	
TorLand5	13597	18 d	torland5-this.is.a.tor.exit.server.torland.me [146.185.23.181]	443	80	
hazare2	13392	3 d	96.44.163.76 [96.44.163.76]	443	80	
Unnamed	13148	16 d	exit2.ipredator.se [93.182.132.103]	9002	9031	
BigBoy	12979	14 d	torproxy10.teamcymru.net [38.229.79.2]	443	8080	
TorLand1	12729	18 d	torland1-this.is.a.tor.exit.server.torland.me [146.185.23.179]	443	80	
manning	12727	3 d	manning.torservers.net [173.254.192.36]	443	80	
hazare	12512	3 d	hazare.torservers.net [96.44.163.77]	443	80	
bolobolo	12238	3 d	bolobolo.torservers.net [173.254.192.35]	443	80	
TorLand2	11984	18 d	torland2-this.is.a.tor.exit.server.torland.me [146.185.23.180]	443	80	
noiseexit01b	11713	17 d	exit-01b.noisetor.net [173.254.216.67]	443	80	
chomsky	11577	6 d	chomsky.torservers.net [77.247.181.162]	443	80	
saeed	10994	3 d	saeed.torservers.net [96.44.163.75]	443	80	
noiseexit01d	10792	17 d	exit-01d.noisetor.net [173.254.216.69]	443	80	

Figura 43. **:** Esta tabla muestra los nodos de salida de Tor que más aportan ancho de banda. Entre más arriba en la lista, el aporte es mayor. En el momento en que capturé esta imagen, el nodo de Noisetor (noiseexit01a) se encuentra en onceavo lugar mundial. Durante otros momentos apareció más abajo o más arriba. La lista cambia muy seguido y de manera repentina. La versión actual está en `http://torstatus.blutmagie.de/`

argumento era que existía una "ciberamenaza" proveniente del *hackerspace*. Esto se reveló en un mensaje dentro de la lista de discusión el 26 de febrero:

> Los agentes [...] y [...][71] nos visitaron e hicieron algunas preguntas. Las respondí completamente y les expliqué la cultura de noisebridge. Les dije que no guardamos bitácoras, que nuestras redes son abiertas, que tenemos terminales públicas, que dejamos entrar a todos, que tenemos un wiki[72] público, que no tenemos jerarquía y que nos reunimos los martes a las 8 PM. Les dije que funcionamos en base al consenso y que las notas de las reuniones se almacenan en el wiki. Les dije que no tenemos bitácora de quién ha entrado o salido del espacio. Les dije que no se requiere membresía para usar el espacio. Les dije que aunque tenemos una población culturalmente formada por académicos y profesionales de la seguridad, acogemos cualquier forma de creatividad. Les dije que teníamos nuestros propios problemas con gente desagradable. Fueron muy amigables y nada amenazadores.

Alguien más comentó que Wikileaks había filtrado nuevos documentos en días anteriores y que quizá la investigación giraba en torno a ello. Andy respondió que le parecía extraño que visitaran el lugar, ya que no es ahí donde se encuentra el nodo de salida físicamente. Respondió lo siguiente al mensaje anterior:

> Un agente del FBI me pidió información sobre el nodo de salida de Tor a través del correo electrónico, sospecho que la visita tiene que ver con eso. Aparentemente alguien ingresó a Twitter a través de Tor y el FBI intenta rastrearlo. Llegar a 2169[73] es una manera... extraña... de lograr ese propósito, considerando que Tor es una red global específicamente diseñada para oscurecer la respuesta a ese tipo de preguntas.

---

[71] Aunque en el mensaje aparecieron los nombres de los agentes, los omití intencionalmente de la tesis.

[72] Un wiki es una página colaborativa que cualquier usuario puede editar. Es una dinámica similar a la de "wikipedia" pero que se aplica para una gran variedad de sitios web.

[73] La dirección de Noisebridge es "2169 Mission St, San Francisco" por lo que muchas veces simplemente se le refiere tan solo como 2169.

Danny O'Brien sugirió que clarificaran la información de contacto en el sitio de Noisetor. Si sucediera un evento similar, los agentes contactarían a la persona adecuada para obtener más información. Andy sugirió crear una página o listado de información de contacto que ahorre tiempo al FBI o cualquier otra institución con preguntas.

Alguien se quejó de que la persona que publicó el primer mensaje hubiese conversado con el FBI y los dejara pasar al espacio. Es obvia la desconfianza que existe en Noisebridge hacia las agencias de seguridad pública, como puede quedar claro por una de las respuestas: "*uh, DON'T TELL THE COPS SHIT. EVER. Fucking duh*". La persona que inició el tema respondió: "La conversación que tenemos en estos momentos es mucho más reveladora que cualquier cosa que les dije". Todos los mensajes de la lista de discusión se encuentran archivados y accesibles públicamente, cualquiera puede verlos. Varios más coincidieron en este punto, ya que suponen que el FBI monitorea frecuentemente la lista de discusión.

La solución propuesta por Andy fue aclarar los datos de contacto en la página de Noisetor y en la información del nodo de salida. Creó una página en el wiki de Noisebridge con instrucciones para que los miembros supieran cómo actuar en situaciones similares. Tiene como título: "¿Qué hacer si el FBI/Servicio Secreto/etc llega a 2169 para investigar el tráfico de Internet?".[74] Las sugerencias se dividen en cuatro apartados:

1. Se cortés: En esta sección se sugiere cómo tratar a los agentes. "El agente quizá es una persona decente haciendo un trabajo duro. No seas maleducado con ellos. Sin embargo, mantén en mente que ningún agente de la ley es tu amigo, no importa qué tan agradable parezca, y mentirle a un agente federal es un crimen serio, incluso si no intentaste engañarlo, e incluso cuando la mentira no es relevante para su investigación. Es probable que ellos sepan más acerca de la situación de lo que te revelan".

2. Contacta a los técnicos: En esta sección se publica el número telefónico de Andy Isaacson y Shannon Lee, los encargados de la parte técnica.

3. Verifica que la investigación se relacione con Noisetor: Aquí se publican las direcciones

[74] En inglés: "What to do if the FBI/Secret Service/etc comes to 2169 investigating Internet traffic" y se encuentra en `https://www.noisebridge.net/wiki/Noisebridge_Tor/FBI`

IP de los sistemas de Noisebridge. Se sugiere que los miembros pregunten a los agentes la dirección IP que están buscando y la comparen con las existentes en el lugar. Si no coinciden, no es un problema de Noisetor y tendrían que contactar a la persona responsable del otro sistema.

4. Comunícate con la comunidad: Se espera que los miembros pregunten el nombre y teléfono de los agentes y los publiquen en la lista de la discusión, comentando la visita.

Noisetor se encuentra todavía en sus etapas iniciales y aparentemente queda mucho por resolver. Este proyecto, como todos los que me comentaron los informantes, son trabajos en proceso. Sólo con el tiempo podremos ver hacia dónde llevan estas estrategias. Algo común en los proyectos en los cuales ha participado Andy, Miguel y Carlos es la búsqueda de mejores dinámicas de trabajo y, en general, convertir al mundo en un mejor lugar. Mejor, por supuesto, en base a sus valores y creencia. La forma en que ellos creen que puede lograrse se discutirá con profundidad en las conclusiones.

Un mundo mejor

¿Qué sucedió con el recetario de mi mamá? En cuanto le comenté a ella que los programadores de mxlOpenSource deseaban convertir su recetario en software libre, se entusiasmó y aprobó la idea. Ellos pusieron manos a la obra, buscando en Internet otros programas similares, para encontrar alguno que se ajustara exactamente a sus necesidades. Pronto encontraron otro software que contenía casi todas las características y era gratis, mas no libre. La sugerencia de los programadores fue que mi mamá revisara el software e hiciera anotaciones sobre lo que siente que falta. También lo que le gustaría que fuera diferente. Todas esas notas servirán de guía para desarrollar un nuevo recetario que se ajuste precisamente a lo que ella necesita.

No sólo ella recibió ayuda. Durante mi asistencia al grupo yo llevaba mis propias dudas e inquietudes y muchas veces recibía respuestas. En el transcurso de redacción de este documento, aprendí LaTeX, un sistema de preparación de documentos bastante complejo. En mis visitas al campo, en reuniones, en foros, listas de correo y salas de charla, me di cuenta de que cada vez más recurría a "la comunidad" en busca de ayuda. Este concepto fue difícil de comprender para mí. Tardé muchos meses en comprender a qué se refieren los *hackers* cuando hablan de comunidad. A pesar de leer libros al respecto, a pesar de escucharlo muchas veces antes, no lo comprendí hasta que formé parte de ella.

Durante entrevistas y encuentros los *hackers* se apasionaron y lanzaron largos discursos sobre cómo deberían funcionar las cosas. Desde el software hasta el gobierno. Muchas veces estos discursos se dieron al final de las entrevistas, cuando les agradecía por prestarme algo de su tiempo. En una ocasión, me encontraba ya en la madrugada con Miguel Venegas dentro de su automóvil, en el estacionamiento de un centro comercial. Yo caía del sueño, fue una semana verdaderamente agotadora y le agradecí la entrevista. Él me exhortó a mantener la grabadora encendida, y prosiguió la plática acerca de su trabajo y lo que le molestaba de él. No fue sino hasta el día siguiente, ya sin cansancio, cuando volví a escuchar la grabación y caí en cuenta de la información tan importante que me había dado.

Fue casi hasta el final de esta investigación cuando entendí lo que los *hackers* me estaban diciendo. A través de las entrevistas y la observación participante se me fue revelando un mundo. Un mundo que todavía no existe de manera física, pero que se encuentra en la mente de los *hackers* entrevistados y los guía en su trabajo diario. Su trabajo, sus actividades, la fuerza que los motiva a emprender muchos de sus proyectos es la esperanza de un mundo mejor. Intentaré, en base a lo que me dijeron y lo que observé, describir ese mundo.

Para llegar a estas conclusiones, como ya mencioné, me apoyé en el análisis del discurso. Las tres categorías en las cuales basé el análisis fueron: Ética, práctica social y politización tecnológica. Éstas categorías se tradujeron en observables, es decir, fragmentos de discurso o de observación que ayudan a comprender a estos *hackers* desde la dimensión teórica planteada. Después de transcribir las entrevistas, busqué sistemáticamente en todo el texto menciones o referencias que respondieran a mis objetivos y arrojaran luz sobre las categorías que propuse. Marqué todas estas menciones y las agrupé con respecto a la categoría, para encontrar similitudes y diferencias. Este proceso de análisis de datos, sistemático y racional, me ayudó a describir las características que los tres *hackers* entrevistados quieren para el mundo. Decidí dividirlo en varias secciones, que presento a continuación.

Un mundo lleno de objetos interesantes

Cuando eran niños, los tres *hackers* se desconcertaron ante la presencia de un objeto extraño: Una computadora personal. Aprendieron a usar la máquina por diversión, se atrevieron abrirla, cuestionarla, programarla. Miguel desarmaba sus computadoras para ver dentro, como muchos niños desarman sus carritos de control remoto. Las primeras experiencias de Carlos iniciaron con los emuladores y videojuegos de consola. Andy programó algunas bromas y juegos para la Apple IIe de su papá. Iniciaron, en pocas palabras, para jugar y curiosear. Con los años esta aproximación se convertiría en un interés mucho más serio, hasta la fecha continúan con una relación similar con estos artefactos. Buscan explorarlos y conocerlos por dentro y por fuera.

Andy comentó que cuando era niño se encontró un tutorial de BASIC de su papá:

"Era un tutorial engargolado de programación en BASIC para minicomputadoras... para microcomputadoras. Y probablemente leí ese libro una docena, o veinte o treinta veces cuando tenía siete u ocho años porque era muy interesante para mi". Algo similar sucedió con la Commodore 64 para Carlos y Miguel. Los artefactos tecnológicos eran interesantes por sí mismos.

Durante la universidad, Andy intentaba hacer *raytracing* de manera distribuida en las computadoras del laboratorio. No era tarea para una asignatura, era un proyecto iniciado por sus inquietudes personales. Como no tenía la capacidad de utilizar las herramientas profesionales disponibles para ello, creó las suyas: "De hecho escribí mi propio *script* de comunicaciones, donde todos los nodos tenían directorios NFS montados, y yo quería dos nodos trabajando en el mismo problema". Cuando trabajó para Cray, haciendo *raytracing* en sistemas distribuidos. Le pregunté si el trabajo que había iniciado en la universidad fue realizado pensando en trabajar para Cray. Me respondió que no, que fue una coincidencia.

> Si, todo se relacionó, si, todo encajó muy bien. El hecho de que trabajé en programación distribuida en IBM y más en las clases universitarias y que hice procesamiento paralelo con computadoras en red para raytracing y DES significaba que era obvio para mi ir a programar en red para Cray. Y lo hicimos. En eso trabajé cuando estuve ahí.

El trabajo que Andy realizó por curiosidad le sirvió posteriormente para obtener ese empleo. Lo mismo sucedió con Carlos, quien aprendió tecnologías por curiosidad y gracias a eso consiguió su primer empleo en un cibercafé y posteriormente en los diferentes lugares donde se ha desempeñado. De la misma forma, Miguel intenta siempre dar un giro tecnológico a sus actividades dentro del Centro Estatal de las Artes, incluyendo los talleres artísticos.

A lo que voy es que, para los tres, la pasión por los objetos es central. Como dijo Miguel: "Las computadoras siempre han sido como tipo mi novia". Si consiguieron empleos gracias a este interés, ha sido sólo una feliz coincidencia. Ellos trabajarían con tecnología con empleo o sin él.

Su relación con el objeto es tan estrecha, que buscan maneras de controlarlo mejor. Las computadoras y los dispositivos se controlan a través del cuerpo. Normalmente a través de las manos, los dedos. Estos *hackers* han desarrollado habilidades combinadas con herramientas para controlar mejor la tecnología.

Carlos memorizó los comandos de Vim, y se acostumbró al teclado Happy Hacking que posee configuraciones muy específicas que se adaptan a su gusto. Cuando impartió el taller de Vim en la reunión de mxlOpenSource, los asistentes vimos que tenía un mayor dominio sobre la máquina que el resto de nosotros, al menos en cuanto a la edición de texto, que es parte de su trabajo como programador.

Miguel modificó un juguete que lee algunas ondas cerebrales para ayudarse en su creación artística. Esto para no depender del *mouse* para generar arte visual con su computadora. Podría conectarse los sensores del juguete y modificar la salida en pantalla mediante las ondas de su cerebro. Él siente que pronto el mouse y el teclado quedarán atrás para dar paso a un control mental, o "nuevos diálogos" como él los llama: "Ya eso ya no, güey. Ni plumas, ya ahora la onda va a ser esa".

Andy configuró el ambiente gráfico de su laptop para ajustarse exactamente a sus necesidades. Se trata del manejador de ventanas fvwm, pero altamente modificado para ajustarse específicamente a sus dedos. Ha utilizado la misma configuración desde la universidad y se encuentra sumamente familiarizado con ella, aunque nadie más puede entenderla.

> Hay muchos lugares en donde, si haces algo perfectamente razonable, la ventana hará algo completamente absurdo. Yo sé cómo arreglarlo, porque he usado la misma configuración durante 20 malditos años... 18 años, pero no es una muy buena configuración. Tiene algunas características muy específicas que encuentro realmente valiosas, que conozco extremadamente bien, porque las he usado del mismo modo durante 20... 16 años o algo así. Así que sigo usándolas. [...] Son importantes para mis dedos, ni siquiera son importantes para mí, son cosas que se basan en acciones. Así que si hago cierto... Hago cierto gesto y sé... Mis dedos

saben exactamente qué sucederá.

Para Andy este tipo de interacción sumamente estrecha y corporal con los artefactos tecnológicos no es nada extraña en el mundo de los *hackers*.

> No creo que sea tan inusual. Mi configuración específica es bastante inusual, es casi totalmente única. Pero eso es muy común para las personas que usan mucho las computadoras, así que muchos *hackers* tienen un entendimiento muy profundo... o entendimiento incrustado muy dentro de ellos sobre lo que sucede, o cómo funcionarán sus interacciones con las cosas.

Encontré una característica más en la relación de los *hackers* con los objetos. Siempre buscan la herramienta adecuada para cada trabajo. Aquella que se ajuste perfectamente y esté diseñada específicamente para la tarea a realizar. Cuando encuentran una que les resuelve sus problemas de manera óptima, se dedican a estudiarla y convertirse en expertos en ella. Esto en contraste con un usuario que decide crear cartas, reportes, ensayos o documentos del trabajo mediante una sola herramienta, como Microsoft Word.

Por ejemplo, cuando entrevisté a Miguel Eduardo por segunda vez, me preguntó si conocía alguna herramienta para hacer libros. En ese momento se encontraba involucrado en la maquetación del libro *Practical Arduino*. Los libros sobre Arduino contienen código fuente, y normalmente resaltan la sintaxis del lenguaje de programación mediante colores, para facilitar la lectura. Miguel buscaba la herramienta adecuada para ese trabajo.

> Entonces creo que lo que me paró mucho fue encontrar una aplicación que pudiera hacer los *highlights*[75] específicos. Por que empecé a redactar un tutorial, te digo que por medio de Adobe Ilustrador. Pero empecé a producir, y ya iba en la pagina 20, un libro alrededor de 100 hojas, entonces yo de repente empecé a cuestionarme: "Oye, ¿qué va a pasar si a mí ahorita me atropellan y les doy el PDF a ellos?" No puedes mantener el mismo estándar, pues. [...] Suponiendo, el Adobe Ilustrador no

[75]Se refiere al resaltado mediante colores.

> lo puedes instalar en Linux, tienes que tener Wine, ¿no?[76] Y además es privativo. Entonces era ahí como que, pum, nos detenía, ¿no? [...] Pero decíamos, es que esa madre no es para eso, pues. Entonces gracias a eso yo descubrí que el Adobe Ilustrador, el Corel no te sirven para hacer esquemas de libros de programación.

Yo le recomendé algunas herramientas, pero nunca super si realmente encontró alguna perfecta. Intentó con LaTeX y DocBook. Buscaba algo que se adecuara exactamente a sus necesidades y que además fuera libre. Herramientas disponibles hay muchas, y todas podrían resolver el problema, pero Miguel intenta todas hasta encontrar la mejor.

Carlos me comentaba que cuando llegó a San Francisco le llovieron ofertas para trabajar en México, de manera remota. Consideró una de ellas, en parte por que elegían la herramienta adecuada para cada trabajo.

> Porque este güey son el tipo de empresa que me gusta, no es un *startup* y no son tan modernos, pero para cada problema o cada trabajo que consiguen, utilizan una aplicación diferente, pues. En el sentido de que si es algo de ofimática, utilizan Mono. [...] Tonces, si ocupan al web, así, usan PHP y Codeigniter.

Para contrastar, recordaré la pequeña oficina del software que diseñaría el recetario para mi mamá. Todos sus programas estaban desarrollados en Visual Basic. La misma herramienta para todos los trabajos. Andy también reveló esta actitud a través de varias anécdotas. Especialmente a través de su preferencia por el sistema operativo BSD.

> Ejecutaba BSD en la mayor parte de mis sistemas y de hecho lo expresaba públicamente. Tengo amigos de esa época que todavía se extrañan con la idea que ahora sólo uso Linux, no uso BSD para nada. Me decían: "Tenías una calcomanía de BSD en la defensa de tu carro, ¡no lo entiendo! Te expresabas enérgicamente al respecto en aquel entonces". Yo respondía: "Sí, la cosas cambian".

[76] Wine es un programa que permite instalar paquetería de Windows en sistemas opearivos Linux.

Este cambio sucedió cuando Andy descubrió que BSD ya no era la herramienta adecuada para su trabajo, y que Linux podía resolver sus problemas. Además, Linux poseía la ventaja de ser un proyecto con una comunidad mucho más grande detrás. Se discutirá sobre este cambio un poco más adelante, al hablar de la comunidad..

Estos *hackers* tienen un interés intrínseco por los artefactos tecnológicos. Favorecerán aquellos que les permitan explorar, modificar y aprender mejor sobre ellos, aquellos que no tengan límites a la hora de ser explorados. Su conocimiento y experiencia con su uso los ayuda a conseguir empleo, ganarse la vida y trabajar en sus propios fines. Además, se relacionan estrechamente con los objetos, cuando encuentran uno que les funciona bien, deciden apegarse e interactuar con él de una manera casi íntima. Con el tiempo se convierte casi en una extensión de ellos mismos y esto les permite conseguir sus propios fines a través de la tecnología.

Un mundo más libre

No basta con que los objetos sean interesantes y les proporcionen conocimiento, también deben ser libres. Los objetos libres, tanto de software como de hardware, facilitan la dinámica de trabajo de los *hackers*. Buscan la libertad para compartirlos, modificarlos, hablar de ellos, de analizar detalladamente su funcionamiento. No únicamente la libertad de utilizar las herramientas. Ellos creen que la tecnología afecta la sociedad y que los objetos que restringen la libertad de sus usuarios no crean un mundo mejor, todo lo contrario. Andy me proporcionó dos buenos ejemplos.

El primero tiene que ver con una tecnología llamada *Digital Rights Management* (DRM). Es un sistema de restricciones digitales para evitar la piratería, generalmente aplicada a libros, música y películas. Por ejemplo, cuando un usuario compra un libro, éste es firmado digitalmente por el proveedor. Para acceder a él es necesaria una clave secreta que guarda el fabricante. Cada vez que el usuario necesita acceso al libro, el sistema se conecta a Internet y verifica que el usuario esté autorizado para ver el contenido.

> No trabajé en un producto con DRM, pero trabajé para una compañía que tenía una línea de productos con DRM. Y mientras trabajé ahí, decidieron que ya no era un negocio que querían continuar. [...] No sacaron mucho dinero de él como esperaban, así que decidieron cerrarlo. De hecho apagaron los... Dejaron los servidores funcionando por seis meses o nueve meses o algo así, pero al final simplemente los apagaron y los clientes que todavía usaban ese video perdieron acceso al video que habían comprado porque la compañía apagó los servidores DRM. [...] Eso es bastante malvado.

Yo le pregunté cómo conciliaba su ética personal con sus actividades laborales. Muchas veces en los empleos se requiere el involucrarse en actividades que no coinciden del todo con nuestras creencias personales.

> He evitado los trabajos en donde me pidan trabajar en cosas a las que me opongo firmemente. Muchas veces he trabajado para compañías que hacen otras cosas, en las que no trabajo directamente, sobre las cuales no concuerdo. Y generalmente, incluso cuando no es directamente mi trabajo, me expreso diciendo: "¡Hey! ¡Debemos hacer lo correcto con respecto a esto!". A veces eso ha traído beneficios.

Su segunda anécdota tiene que ver con la restricción de derechos de acceso. Las películas en DVD y BluRay están cifradas y sólo pueden reproducirse en dispositivos de alguna región determinada. Norteamérica, por ejemplo, es la región 1, y latinoamérica la región 4. De esta forma, un disco comprado en México no puede ser reproducido en Estados Unidos y viceversa.

> Me entrevistó una compañía hace algunos años, y parecía que tenían problemas interesantes para resolver. Pero el producto en el que trabajaban fue algo que encontré realmente... realmente perturbador. Probablemente legal, pero no era una tecnología positiva para los usuarios, tampoco pensé que era buena para la [empresa] a la que se la vendían. Ni siquiera para la industria en general.

Me explicó que era un sistema DRM para discos BluRay que involucraban una comprobación que requería conexión a Internet. Era una medida de control demasiado restrictiva, desde el punto de vista de Andy. Durante la entrevista, el empleador explicó la tecnología para ver si Andy tenía la habilidad de resolver los problemas.

> Empezó a explicar lo que era, y después de diez minutos dije: "¿Sabes? Gracias por hablar conmigo, pero no es necesario continuar esta conversación por que no hay manera de que yo trabaje para ti. No puedo trabajar en ese producto y creo que tu tampoco debes hacerlo". Y se quedó como: "Ok, gracias por ser honesto". [...] Me imaginé que tenía sus detalles, pero yo pensé: "No, ese no es un sistema en el cual estoy interesado en absoluto, no puedes pagarme lo suficiente para trabajar en ello". Así que hay gente haciendo cosas realmente estúpidas en el mundo y trato de trabajar con ellas lo menos posible.

La moraleja es que Andy no trabajará en algo que limite la libertad de las personas. Decidí enfocarme en estas dos anécdotas porque son muy reveladoras, pero la libertad es una constante en el discurso de los tres *hackers*. Miguel me comentaba su frustración ante el incidente del Pantone libre y cómo las corporaciones intentan controlar la información: "Más que todo cuando empiezas... a... a ver que hay un control de información, que la gente ya no puede escribir, que la gente es espiada, la gente te sigue tus pasos entonces tú dices ¿qué pedo, güey? O sea, la información es libre".

Esta fue también una de las sus quejas contra las escuelas. Según Miguel, el hecho de que a los estudiantes se les enseñe en clase paquetería de software comercial los hace depender de corporaciones. Por ejemplo, le enseñaron diseño gráfico mediante Adobe Ilustrator, que es software comercial, pero él deseaba utilizar Inkscape, que es una alternativa de software libre:

> En las clases nos dijeron: "Vas a usar Adobe Ilustrador" "¿Por qué? Yo quiero usar el Inkscape". "No, es que no sabemos usarlo". O sea, como que en las escuelas te

> cierran, te hacen como en un cubito con paredes, y como que tu vas en el centro y que tú te quieres safar tantito: "¡Eh, eh! ¡Véngase pa' ca!" Tonces como que tú dices, "Ey, aguanta". No te dejan. [...] Entonces yo creo que los alumnos ocupan salirse de esto.

Para Miguel el uso de un software no se limita únicamente a considerar las ventajas técnicas. Es una cuestión de libertad, de eliminar la dependencia de las corporaciones y el "imperialismo" como él dice. De dejar que los individuos puedan resolver sus propios problemas y trabajar en comunidad.

Carlos Iván tiene puntos de vista similares. Se quejaba de que en su trabajo "Godínez" no le daban suficiente libertad de resolver los problemas como él quería. Se quejaba de que muchas órdenes le llegaban de arriba y se tenía que apegar a ellas, incluso cuando él consideraba que había mejores opciones. Se quejaba de las empresas que sólo buscaban sacar dinero y se olvidaban del individuo. Yo comenté que en toda empresa comercial se busca dinero y él respondió:

> Pero hay maneras de conseguirlo y hay maneras de tratar de conseguirlo y hay maneras de crear un producto. Tonces, como te has dado cuenta en los *startups* les importa más los programadores. O sea, como [...] github. En github no importa si no vas a la oficina. [...] Lo que hacen es: Si puedes ir a la oficina, tienes que ir a la oficina a hacer tiempo, pero realmente prefieren que estés creativo y que estés feliz y así sacan más producción. [...] Tonces las horas no son mediciones de productos exitosos, pues. Y más en esto de tecnología. En esto de tecnología derrepente a alguien se le puede prender el foco y hacer algo bien exagerado en unas horas.

Para Carlos, los trabajos que eliminan las restricciones y permiten al individuo tomar sus propias decisiones son mejores. Permiten que el empleado esté feliz e incluso sorprenda entregando más de lo que se le había pedido.

> Y lo bueno de eso es como... puedes hacer todavía cosas de más, y se te... se te, ¿cómo se dice? Se te recompensa, pues. Te dicen, está bien. No con el dinero, pero te... reconocen, pues, que... que te fuiste más allá de lo que debiste haber hecho y dice: Y si es útil, está bien, si no, también, o sea, qué fregón por que te da una motivación en el trabajo. Tonces, eso es tratar a alguien como un ser humano, la verdad.

Estos son sólo algunos de muchos ejemplos de cómo estos *hackers* conciben la libertad. Ellos creen en la capacidad de autodeterminación del individuo. Por supuesto, su área de conocimiento es la tecnología, por lo que la mayor parte de sus opiniones giran en torno a esta área. Pero durante diversos momentos, como ya se mencionó, hablaron también del gobierno, las escuelas y la sociedad, expresando ideas similares. Saben que la libertad, por sí misma, no mejora el mundo, pero creen que las restricciones impiden encaminarse esa dirección. Dadas dos alternativas, los *hackers* escogerán la más libre.

Un mundo comunitario

Comunidad es una palabra que apareció una y otra vez en el discurso de los *hackers*. Ellos prefieren el trabajo comunitario, organizaciones no autoritarias en donde todos tengan la posibilidad de aportar al bien común. Normalmente se refieren al mundo del software pero, como se ilustró con el ejemplo de Noisebridge, también puede operar en otros ámbitos.

Cuando Miguel Eduardo me hablaba de NorteLab, me comentaba: "El proyecto es un colectivo donde cualquier persona puede unir su granito de arena, ¿no?". Esta es la idea central detrás de los proyectos colectivos que él inicia y en los cuales participa. Tanto en la distribución Linux Tuquito, como en ELUG, como en el libro libre de Arduino, como usuario Pure Data, Miguel forma parte de varias comunidades autogestionadas que no dependen de un poder centralizado. Para Miguel, la comunidad también sirve para liberar a la gente de la dependencia. Logra que ellos tengan la capacidad de crear las cosas por sí mismos y resolver sus propios problemas tecnológicos.

> Y que no sean inútiles, a fin de cuentas. Que la gente, a fin de cuentas, no se ponga en la parte de decir: "Ay, es que no puedo porque no sé electrónica". No, aquí te podemos ayudar, somos una comunidad y todos podemos aprender. Cada quién es una eminencia en lo que sabe, ¿no? Es como que aprendemos de todos, a fin de cuentas.

En la comunidad, entonces, se comparte el conocimiento bajo la idea ya mencionada de que el conocimiento es libre. Incluso cuando yo usaba el término "grupo de programadores", Miguel me corregía diciendo: "Es una comunidad". Este punto es central para los tres *hackers*.

Este modelo es incompatible con el orden social en el que se encuentran insertos. Las instituciones sociales normalmente son jerárquicas y poseen una autoridad central. En el caso de Noisebridge, cuando la comunidad intentó registrarse como una organización no lucrativa se encontraron con que, por cuestiones legales, la organización debía tener un presidente. Ellos no querían una figura de autoridad como esa, pero tuvieron que designar todo un comité para cumplir con el papeleo. Andy me comentó: "Todos sabíamos que necesitaríamos un tesorero. Alguien que estaría a cargo del dinero, que se asegurara de que pagáramos la renta. Cosas así. Realmente queríamos evitar las dinámicas de poder por tener presidente y así".

Noisetor, el otro proyecto de Andy, también funciona gracias a una comunidad. Algunas personas sólo donan dinero y otras su tiempo. Algunas tienen mayor conocimiento, otras menos, pero si desean aportar y les interesa el proyecto, pueden encontrar algún problema que pueden resolver.

Un ejemplo revelador es cuando Andy dejó de usar BSD como sistema operativo en su laptop. En ese tiempo BSD no ofrecía mucho soporte para hardware, así que había algunos dispositivos que no funcionaban al instalar el sistema. Investigando, descubrió que no funcionaban debido a un problema técnico de BSD. Intentó resolver el problema por sí mismo, pero pronto comprobó que era demasiado complejo para sus habilidades. Descubrió también que la comunidad de BSD ya había intentando resolver el problema varias veces, sin éxito. Como no se vislumbraba solución en el futuro cercano, buscó alternativas.

> Así que el sistema Debian era mucho más interesante para mí, y también el hecho que Debian era un proyecto desarrollado por comunidad en vez de un proyecto corporativo[77] fue llamativo también. Así que terminé cambiando mi máquina, mi laptop principal a Debian en algún punto.

Cuando una comunidad no dio respuesta a sus necesidades, Andy migró a otra. Dice que se mantiene enterado de lo que sucede en la comunidad de BSD, pero concede que Linux ya se encuentra mucho más adelantado tecnológicamente:

> Puedo resolver la mayor parte de mis problemas usando Linux y... No soy un gran fan de las monoculturas, me gusta la diversidad en el software, creo que es bueno para la sociedad y para la comunidad técnica que haya mucha polinización y diferentes lugares para que surjan ideas.

Carlos Iván también es parte de varias comunidades. Para él, las comunidades de software libre son oportunidades perfectas para compartir el conocimiento, aprender de los errores y aciertos de los demás. Le pregunté por qué migró a software libre cuando era tan joven, me respondió:

> Pues básicamente por el hecho de poder compartir herramientas, o sea. No tenía mucho dinero y, número dos, aprendes mucho viendo el código de los demás. O viendo sus ejemplos o cómo hizo tal cosa. Entonces... Pues yo digo que eso fue lo, más que nada, pues del hecho que te puedan compartir algo y lo puedas accesar sin problemas.

Él participa en la comunidad de Perl, Debian y Linux en general. En realidad, no se requiere ser desarrollador de software para formar parte de la comunidad, cuando alguien simplemente utiliza software libre se considera que está dentro de la comunidad. Carlos me aclaró cuando le pregunté qué es la comunidad de Perl: "Los usuarios, los desarrolladores, la gente que trabaja

[77] Aunque todas las versiones de Linux son software libre, algunas son desarrolladas por corporaciones con fines de lucro como ya se mencionó. Este es el caso de Red Hat, por ejemplo.

con Perl". Aunque no hay una autoridad central en ellas, normalmente existe un coordinador del proyecto. Carlos ha realizado aportaciones a varias comunidades, principalmente parches que resuelven errores.

La mayor parte de los proyectos de software libre tienen una lista de correo en donde se avisa sobre los últimos errores detectados por la comunidad. Al ser públicos, cualquier miembro de la comunidad puede proponerse solucionarlos. Cuando alguien desarrolla una solución, la publica con una descripción y un archivo adjunto que contiene el código necesario, conocido como "parche". Carlos ha aportado varios de estos parches. "Ese... ese es mi tipo de contribución pero nunca he hecho así como un software sólido". Es decir, Carlos nunca ha iniciado un software desde el principio. "Siempre me he manejado dentro de los que ayudan adentro de un software sólido".

Por supuesto que todo este trabajo implica una fuerte cultura *do-it-yourself*. Primero intenta resolver los problemas tú mismo, si no puedes, recurre a la comunidad. Miguel con sus talleres sobre tecnología intenta transmitir eso a sus estudiantes: "Tonces como que más que todo estamos apoyando mucho a la filosofía de 'hazlo tú mismo'"; Carlos con sus aportes a proyectos de software libre y los *hackatones*: "Es bien DIY el pedo"; Andy con la creación de Noisebridge: "*Do-cracy it's what they call it at Noisebridge*".

Otra característica que mencionan es la capacidad de poder expresar ideas y críticas libremente dentro de la comunidad. Como afirma Miguel cuando creó NorteLab: "Todo salió por que queríamos un espacio donde todo mundo se confrontara, donde todo mundo chocara, donde todo mundo platicara, ¿no?". También Carlos coincide, cuando comenta la manera de trabajar en Noisebridge, y cómo le gusta que las ideas se puedan criticar abiertamente:

> Es por eso que Brasil es una chingonería, pues, porque en Brasil se tiran mierda entre ellos, o sea. Entre los brasileños es de: "Este pedazo de software es basura". Y van con el gobierno y les dicen y los del gobierno: "Ok, repáralo" y ahí está. Por eso, desde que entró Lulla a Brasil se cambió toda la infraestructura, pues, porque él mismo dijo: "Yo no puedo hacer nada con la computadora, pues y eso que yo no

> sé nada de computadoras". [...] Yo espero que a mí también me critiquen, yo sería feliz de que [mi jefe] me dijera algo de mi código. O sea, y eso es lo que me gustó de Noisebridge, en Noisebridge, el primer día que llegue: "Ah, ¡cómo eres güey! Pudiste haberlo hecho de esta forma". Y sí es cierto, y aprendí, o sea, aprendí de eso. No me engreí y: "Ah, te voy a partir la madre". O sea, no.

Los tres *hackers* prefieren trabajar en comunidad. Las ventajas del trabajo comunitario son obvias para ellos: Compartir el conocimiento, solución rápida a problemas que afectan a todos, desarrollo personal de los individuos que forman parte de ella. En palabras de Andy, sobre la fundación de Noisebridge:

> Si le das espacio a la gente para hacer cosas... No sólo espacio físico, también espacio mental, y una ausencia de reglas, algunas personas te sorprenderán positivamente. Harán cosas que nunca hubieras pensado y que hacen del mundo un mejor lugar.

Los tres saben que no todo mundo generará cosas increíbles al trabajar en comunidad. Siempre existen las personas más involucradas, los más activos y creativos. Pero por la simple posibilidad de que surjan y se desarrollen ideas para mejorar al mundo, por que ese potencial esté ahí, vale la pena trabajar en comunidad. De nueva cuenta, la comunidad proporciona la libertad para que los individuos más interesados aporten en medida de sus posibilidades.

¿Cómo programar el mundo mejor?

Estos *hackers* se encuentran con un problema. Desean un mundo mejor, lo tienen bien definido en su cabeza, pero la realidad en la que viven todavía no es compatible con esa utopía. Windows sigue siendo el sistema operativo más utilizado, el software comercial genera billones de dólares anualmente, la mayor parte de las estructuras sociales poseen autoridad centralizada y las computadoras cada vez más instrumentos de control, no de liberación (Feenberg, 2002). ¿Cómo lidian con esta discrepancia? ¿Cómo negocian su vida diaria? Este es el punto donde los *hackers* difieren bastante.

Miguel Eduardo acepta que es imposible, por el momento, ser diseñador gráfico y no depender de ningún software comercial:

> Si eres extremadamente de software libre, como Richard Stallman, que dice que Ubuntu es privado y que él hizo su propio sistema operativo que no es privado y que no carga paquetes privados y si te pones extremadamente izquierdista, [...] todos los diseñadores... Muy pelada, no imprimirían carteles. Así de sencillo.

Por otra parte, Carlos Iván se queja de la gente que hace contratos con Microsoft. Cuando le pregunto por qué cree que lo hacen, responde:

> La gente que hace servidores con Windows es: Uno, porque está bien pendeja, simplemente le gusta esa decisión, o porque tiene un contrato con Microsoft. O porque es una pinchi generación anciana, pues. [...] Que no tiene nada de malo, cada quién puede elegir con quién trabajar y todo eso, pues, pero yo digo que es muy pendeja en el sentido de que todos hemos sufrido a base de Windows, pues.

Concede el derecho que tienen esas otras personas de elegir software de Microsoft, por más que le moleste y le parezca irracional. Pero le parece una pésima decisión. Cuando Andy me comentaba que en Noisebridge buscaban tener el menor número de reglas posible, le pregunté si ese modelo se podría transportar a otras áreas de la sociedad.

> Posiblemente. Noisebridge ha construido un ejemplo positivo y un ejemplo negativo en ese espacio. [...] Estábamos intentando al principio establecer qué reglas tener, qué permitiremos que la gente haga. Siempre hubo un argumento muy fuerte a favor de quitarnos del camino, dejar que la gente haga lo que quiera y dejar que te sorprendan. Y eso fue educativo para mi, verlo suceder y verlo funcionar. Pero al mismo tiempo ves que gente hace cosas totalmente inútiles y que no coinciden con mis propias metas, por ejemplo.

Para Andy el modelo comunitario permite que las personas se desarrollen, aunque no evita que otras hagan cosas inútiles, o simplemente estorben a los demás. El mundo, simple y sen-

cillamente, no está lleno de *hackers*. Las actitudes que van en contra de sus valores abundan, y en ocasiones chocan con la realidad que los rodea.

¿Cuál es su solución? ¿Cómo transformar este mundo, entonces? Aunque los tres *hackers* coinciden en su percepción de los objetos, la libertad y la comunidad, cada uno toma una postura diferente para defender sus ideales. Andy, me contó una anécdota de cuando iba en la preparatoria que me parece representativa de su actitud personal:

> Había un niño en mi salón de la preparatoria que era *gay*. En retrospectiva, era muy obvio. Y vi que lo molestaron varias veces. Éramos buenos amigos en ese momento. No participé en el acoso pero tampoco ayudé mucho más, sólo reprobaba la actitud de la gente que lo acosaba.

Esta conducta de no oponerse activamente pero tampoco participar es similar a la que toma con respecto a sus ideales. Cuando se inició la discusión en Noisebridge sobre cerrar las puertas debido a las personas que entraban y no aportaban a la comunidad, Andy no estaba de acuerdo. Según él, las puertas debían estar abiertas para recibir a todo mundo. Pero en las discusiones al respecto su actitud era similar: "Todo lo que puedo hacer es decir: Yo no haría eso, no creo que debamos hacer eso, pero no puedo impedir que la gente se porte como idiota".

De manera similar, Andy busca influenciar el mundo que lo rodea de acuerdo a sus valores. Pero esto no implica una protesta activa ni perturbar el orden, más bien expresar su desaprobación o apoyo a ciertas ideas. Por supuesto, participar en medida de lo posible, pero no tanto en contra de ideas negativas, si no a favor de valores positivos (comunidad, libertad, justicia).

> Una de las compañías para las que trabajé... Era una compañía de hardware. Hacen chips que se incrustan en los productos de otra gente, y tenían un problema con sus drivers de Linux, los cuales eran simplemente terribles y con una licencia muy mala. Durante el tiempo que estuve ahí dije: "¡Hey! ¡Deberíamos mejorar

> esto!" No ayudé con ningún otro problema que tuvieron ni trabajé en ello, pero varios años después finalmente mejoraron... Creo que al principio de 2011 liberaron su código fuente y tuvieron una gran conferencia de prensa por el hecho de que sus drivers eran... Que estaban intentando trabajar mejor con la comunidad Linux, hacer mejores drivers *open source* específicamente para los productos que tenían problemas. De cierta forma mejoraron y tuve algo que ver en esa mejoría, pero no... No me tomaría ningún crédito por ello.

Hay una diferencia en su participación en Noisebridge y Noisetor, la cual es mucho más activa. En esos casos, como iniciador de los proyectos, tuvo oportunidad de que su voz tuviera más peso, de tomar decisiones y ajustarlos fielmente a sus valores. Aunque intentó no ser impositivo y trabajar en comunidad, como de costumbre. En una ocasión Andy me explicó claramente su postura acerca imponer ideas sobre la tecnología. Me lo comentó justo después de su anécdota de cuando abandonó BSD decepcionado.

> Incluso en esa época tenía mucho... Tenía la misma idea acerca de... O falta de interés en la evangelización. No estoy interesado en decirle a la gente cómo usar sus computadoras. Eso no es valioso para mi. Si alguien coincide conmigo o tiene preguntas interesantes estoy feliz de responderlas, pero los fanáticos me hacen realmente infeliz. [...] La actitud de: "Tienes que hacer esto" sin comprender lo que intentas hacer y por qué tus concesiones pueden ser diferentes a las de alguien más. La habilidad de comprender que las nociones de valor de la gente son diferentes y que las concesiones serán diferentes como consecuencia es uno de los núcleos de mi... de mi visión del mundo. Así que los fanáticos, o sea, simplemente... Es algo que odio.

Él trabajará a favor de sus valores, pero nunca será un activista en el sentido de Richard Stallman. Cree que la postura de otros usuarios es respetable, sobre todo por que tienen valores diferentes, y por tanto no se puede juzgar absolutamente sobre el uso que otros den a

la tecnología. Sin embargo, mostrará su desaprobación cuando ocurra algo que él considera injusto. Como en el caso del reproductor de BluRay o cuando la comunidad intentó cerrar la reja de Noisebridge para evitar el acceso a ciertas personas.

Por otra parte, Carlos Iván me contó que desde que llegó a San Francisco, ha participado en varios eventos llamados *hackatones*. El formato varía de acuerdo al tipo de evento, pero normalmente son reuniones donde diferentes equipos se trabajan en proyectos de tecnología. Se realizan en todas partes del mundo, pero en el área de la bahía hay muchísimos.

> Hay veces que la gente se junta y es nomás como para que... para que... estás como escribiendo código o *hackeando* o hardware o lo que sea, pero hay más gente a tu alrededor, pues que es de la misma como... El mismo círculo, del mismo cotorreo de la misma práctica, pues. Y hay veces que algunos *hackatones* si tienen como, este... sus reglas o tienen sus metas, pues, para el siguiente día. Y casi siempre son como 24 horas o inclusive 42 horas. Sin dormir, o con dormir como seis horas y siguiendo escribiendo código.

Le pregunté qué ganaban los asistentes a esos eventos: Conocer gente, obtener conocimiento y desarrollar habilidades. Básicamente las bondades que los *hackers* ven en el trabajo comunitario. Pero muchos *hackatones* también tienen otros fines. También ayudan a proyectos de software libre, principalmente a resolver los errores que los mismos desarrolladores no tienen tiempo de parchar.

> Como [en] muchos *hackatones* de San Francisco se está dando, que en vez de poner dinero o donar dinero a los proyectos de software libre y de open source, donas los *commits*[78]. ¿Ya vez que todo ahorita está en el GitHub?[79] [...] Entonces muchos desarrolladores no tienen el tiempo para arreglar cosas y lo que hacen en

[78] Un *commit* es el cambio a un software que queda registrado mediante un software de control como git o GitHub.

[79] GitHub es una red social de programación. Los usuarios suben su código fuente y se encuentra disponible para buscar y copiar. Un *commit* se refiere a un cambio hecho a un software determinado. Cualquier usuario puede hacer cambios al software, pero los cambios sólo pueden ser aplicados por los administradores del repositorio. A la solicitud de hacer cambios para que el administrador los apruebe se le llama *pull-request*.

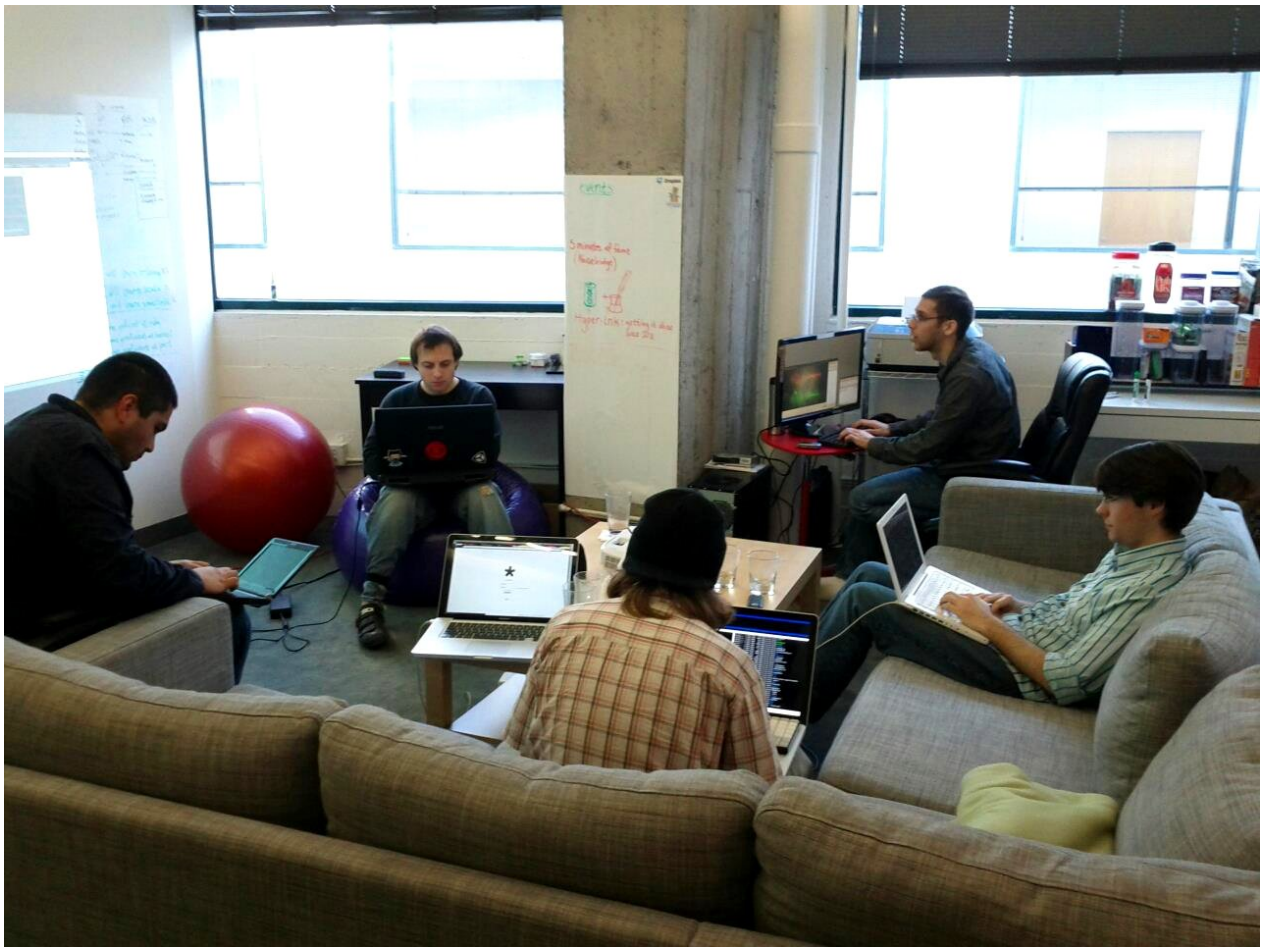


Figura 44. **:** Carlos Iván (derecha) durante el *hackatón* CodeSF (`http://codesf.org/index.html`) en las oficinas de la empresa Hyperlink. (La fotografía fue proporcionada por Carlos).

> los *hackatones* es que buscan los... los *bugs* que gente pone y la gente se pone a desarrollar esos *bugs* y luego ya hace como un *pull-request*, que es... que le manda el parche, pues, al desarrollador original.

Un ejemplo de este tipo de *hackatones* seria *Bits of kindness*. Pero no todos siguen el modelo comunitario. Existen otros eventos virtualmente idénticos, pero que tienen fines comerciales.

> Los *hackatones* que tienen premios o que están vestidos con dinero realmente se les llama "eventos". [...] Y hay gente que si les llama *hackatones*, pero no se da tanto, pues. Se ve mal. [...] Es como... Como busca talentos. Esos eventos se manejan de

> esa manera, pues. El morrillo que *hackee* más o que haga más proyectos o más de ese tipo de cosas es al que le ofrecen un trabajo. ¿Sabes cómo?

De los *hackers* entrevistados, Miguel Eduardo tiene la postura más combativa. Con ELUG dio talleres para convencer a la UABC de migrar a tecnologías libres, pegó carteles de protesta contra el gobierno y la última vez que lo entrevisté se involucró en talleres y conferencias para convencer a la gente de compartir la información.

> Pues es que no es campaña, es más que todo una difusión, vaya, de... De decir a la gente que tiene... Que hay otras propuestas en la red, de compartir, ¿no? Por que a fin de cuentas no somos un partido, nada que ver con eso. No estamos casados con un partido, no. Al contrario, lo que nosotros hacemos es darles herramientas a los artistas, dar talleres de Inkscape, dar talleres de Ubuntu a la gente donde sepa pues ver comparativas de qué pueden bajar, qué pueden ofrecerles en la red, tantos programas.

Los tres tienen posturas diferentes sobre cómo actuar conforme a sus valores. Pero los tres coinciden en dos puntos muy importantes: Son persuasivos. No quieren obligar a nadie a convertirse a sus valores, no desean romper la ley para conseguir sus propósitos. Creen que pueden influir el mundo a través del poder de convencimiento. Creen que la comunidad crecerá en cuanto la gente vea los beneficios y decida unirse. Hay eventos que ayudan a esto, como los *installfests*, los *hackatones* o conferencias, pero siempre son persuasivos.

Otro punto notable es que para los tres el medio para llegar a ese mundo mejor es a través del trabajo duro y bien hecho. Estos *hackers* tienen una imagen muy negativa de la gente que no cree en su trabajo, de aquellos que no lo hacen bien, de los que no se comprometen. Recordemos que Carlos habla de los "empleados Godínez", aquellos que sólo se interesan por el dinero y reciben órdenes. Miguel habla de la "gente normal", aquellos que no buscan alternativas y sólo les preocupa sacar adelante su trabajo sin cuestionarlo. Andy marca su diferencia entre la gente hábil e inhábil, y especifica todavía más: ¿Eres buena o

mala persona? Los tres, en algún momento y sin yo preguntarlo, encontraron etiquetas para aquellos que no comparten su actitud.

Y esa es básicamente la diferencia entre estos *hackers* y los no-*hackers*: Actitud. Al trabajar, al congregarse en LUGs, *installfests*, *hackatones* y *hackerspaces*, no sólo juegan con la tecnología, no sólo se desarrollan profesionalmente. Trabajan para construir, poco a poco y con perseverancia, una utopía. No se despiertan cada día deseando iniciar una revolución. No luchan la guerrilla contra Microsoft, Apple o el capitalismo. Saben que los cambios se logran poco a poco y persuasivamente, no por la fuerza.

Cuando uno de los *hackers* toma una decisión en su trabajo, cuando elige comprar una computadora, cuando decide emprender un proyecto o no, se pregunta: ¿Qué tanto me acerca esto al mundo en el que quiero vivir? ¿Ayudará esto a mejorar la sociedad? ¿Contribuyo con esto a mis metas? Si la respuesta es negativa, buscarán una alternativa que se acerque más. Y su herramienta más poderosa, su habilidad más desarrollada, es la programación. Cuando programan, no sólo crean software, programan un mundo mejor.

Esta es la ética de los tres *hackers* presentados en esta tesis. A lo largo de dos años la vi en acción, participé de ella a través del software libre, aportando a la comunidad como me fuera posible. La comunidad me ayudó mucho: Me dio Linux, sistema operativo que utilizo diario. Me dio otros programas que uso diariamente: BibTeX, Firefox, LaTeX y muchas más. Espero que esta tesis pueda ser también un aporte a la comunidad *hacker*. Soy un miembro de la comunidad.

No puedo decir que la ética de estos *hackers* no me afectó. Repensé la forma como me relaciono con la tecnología. Intento promover el software libre y recomendarlo como alternativa en medida de lo posible. Cuando hablo de mi tema, me es difícil lidiar con los prejuicios de la gente. Escuchan *hacker* y piensan que hablo de delincuentes, ladrones, gente que husmea en las cuentas de correo de los demás para ver qué se encuentra de provecho. ¿Cómo explicar en cinco minutos todo el mundo que acabo de describir? Es muy difícil. Pero al menos puede hacerse en alrededor de 200 páginas.

Índice de figuras

Referencias

Abagnale, F. (2000). *Catch me if you can: the true story of a real fake.* Nueva York: Broadway.

Alasuutari, P. (1995). *Researching culture: Qualitative method and cultural studies.* London: SAGE publications Ltd. Disponible en `http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:Researching+culture\#0`

Alexander, J. C. (1991, abril). Sociología cultural: lo sagrado y lo profano en el discurso tecnológico. *Revista Mexicana de Sociología*, *53*(2), 283. Disponible en `http://www.jstor.org/stable/3540806?origin=crossref`

Aristóteles. (2007). *Ética nicomaquea.* Buenos Aires: Ediciones Colihue. Disponible en `http://books.google.com.mx/books?id=8lctITxKx4sC`

Badham, J. (1983). *Wargames.* Metro-Goldwin-Mayer.

Barker, C., y Galasinski, D. (2001). *Cultural studies and discourse analysis: A dialogue on language and identity.* London: Sage publications Ltd. Disponible en `http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:Cultural+studies+and+discourse+analysis\#0`

Basalla, G. (1994). *La evolución de la tecnología.* Barcelona: RBA editores.

Bauman, Z. (2006). *Ética posmoderna.* México, D.F.: Siglo veintiuno. Disponible en `http://books.google.com.mx/books?id=nkiwwGCtDFMC`

Bijker, W. E., Hughes, T. P., y Pinch, T. J. (1987). *The social construction of technological systems.* Massachusetts: The MIT press.

Blackburn, S. (2003). *Ethics, a very short introduction.* New York: Oxford University Press.

Castells, M. (2003). Internet, libertad y sociedad: Una perspectiva analítica. *Polis, revista de la universidad bolivariana*, *1*(004).

Castells, M. (2006). *La era de la información: Economía sociedad y cultura. Volumen I: La sociedad red.* México, D.F.: Siglo veintiuno.

Castells, M. (2008). *La era de la información. Economía sociedad y cultura. Volumen 1: La sociedad red* (Primera ed ed.). México, D.F.: Siglo veintiuno.

Cheney, G., Lair, D., Ritz, D., y Kendall, B. (2010). *Just a Job? Communication, Ethics, & Professional Life* (Vol. 61) (n.º 1). New York: Oxford University Press.

Clifford, J. (1999). *Itinerarios transculturales* (Primera ed ed.). Gedisa.

Coleman, E. (2005). Three ethical moments in Debian. *Retrieved May*, *23*, 2007. Disponible en `http://papers.ssrn.com/sol3/Papers.cfm?abstract\_id=805287`

Coleman, E. G. (2010, octubre). Ethnographic Approaches to Digital Media. *Annual Review of Anthropology*, *39*(1), 487–505. Disponible en `http://www.annualreviews.org/doi/abs/10.1146/annurev.anthro.012809.104945`

Coleman, E. G., y Golub, A. (2008, septiembre). Hacker practice: Moral genres and the cultural articulation of liberalism. *Anthropological Theory*, *8*(3), 255–277.

Coleman, G. (2004). How free became open and everything else under the sun. *M/C: A Journal of Media and Culture*, *7*(3). Disponible en `http://kb.cospa-project.org/retrieve/2466/colemanhill.pdf`

Coleman, G. (2005). The Social Production of Ethics in Debian and Free Software Communities: Anthropological Lessons. En *Free/open source software development* (pp. 273–295). Idea Group.

Coleman, G. (2009, agosto). Code is speech: Legal tinkering, expertise, and protest among free and open source software developers. *Cultural Anthropology*, *24*(3), 420–454.

Coleman, G. (2010). The Hacker Conference: A Ritual Condensation and Celebration of a Lifeworld. *Anthropological Quarterly*, *83*(1), 47–72.

Coleman, G. (2012). *Coding freedom*. Princeton: Princeton University Press.

Contreras, P. (2004). *Me llamo Kohfam. Identidad hacker: Una aproximación antropológica* (Primera ed ed.). Barcelona: Gedisa.

Denning, D. (1996). Concerning hackers who break into computer systems. En *High noon on the electronic frontier: Conceptual issues in cyberspace* (pp. 137–164). Bradford.

Evans-Pritchard, E. E. (1990). *Ensayos de antropología social* (tercera ed.). México, D.F.: Siglo veintiuno.

Fabian, J. (1983). *Time and the other: How anthropology makes its object.* New York: Columbia University Press.

Feenberg, A. (2002). *Transforming technology. A critical theory revisited.* Oxford: Oxford University Press.

Flick, U. (2004). *Introducción a la investigación cualitativa.* Madrid: Morata.

Foucault, M. (1992). *El orden del discurso* (Vol. 18) (n.º 70). Buenos Aires: Tusquets.

Gates, W. H. (1976). An open letter to hobbyists. *Homebrew Computer Club Newsletter*, *2*(1), 2. Disponible en `http://www.digibarn.com/collections/newsletters/homebrew/V2\_01/index.html`

Geertz, C. (2005). *La interpretación de las culturas.* Barcelona: Editorial Gedisa.

Gibson, W. (1984). *Neuromancer.* Penguin group.

Giménez, G. (2005). *Teoría y análisis de la cultura. Volumen I.* México, D.F.: CONACULTA.

Goldstein, E. (2001). *Freedom Downtime.* 2600 productions.

Goldstein, E. (2008). *2600: A hacker odissey.* New York: Wiley publishing.

Gordon, S. (1996). The generic virus writer II. En *6th international virus bulletin conference* (pp. 1–16). Disponible en `http://www.research.ibm.com/antivirus/SciPapers/Gordon/GVWII.html`

Guber, R. (2001). *La etnografía. Método, campo y reflexividad.* Bogotá: Norma.

Harris, M. (2001). *Antropología cultural* (segunda ed.). Madrid: Alianza Editorial.

Hebdige, D. (1979). *Subculture, the meaning of style.* New York: Routledge.

Himanen, P. (2001). *La ética del hacker y el espíritu de la era de la información.* Barcelona: Destino libro.

Latour, B. (2007). *Nunca fuimos modernos, ensayo de antropología simétrica.* Siglo veintiuno.

Latour, B. (2008). *Reassembling the social: an introduction to actor-network-theory.* Oxford University Press. Disponible en `http://books.google.com/books?id=-NKGPwAACAAJ`

Levy, S. (1984). *Hackers, heroes of the computer revolution.* New York: Penguin books.

Lin, N. (2002). *Social capital: a theory of social structure and action.* Nueva York: Cambridge university press.

Littman, J. (1996). *The fugitive game: Online with Kevin Mitnick.* New York: Little, Brown and company.

Littman, J. (1997). *The Watchman: The Twisted Life and Crimes of Serial Hacker Kevin Poulsen.* New York: Little, Brown and company.

Lizama Mendoza, J. A. (2002, abril). Hackers: De piratas a defensores del software libre. *Revista Mexicana de Ciencias Políticas y Sociales*, *XLV* (185), 91–108.

Lizama Mendoza, J. A. (2005). *Hackers en el contexto de la era de la información.* Tesis Doctoral no publicada, Universidad Nacional Autónoma de México.

Malinowski, B. (1986). *Crimen y costumbre en la sociedad salvaje* (Ariel, Ed.). Barcelona.

Martínez Miguelez, M. (2006). *Ciencia y arte en la metodología cualitativa.* México, D.F.: Trillas.

Marx, K., y Engels, F. (1966). *Obras escogidas I.* México, D.F.: Editorial Progreso.

Mead, M. (2000). *Antropología, la ciencia del hombre.* elaleph.com.

Meyer, G., y Thomas, J. (1990). The baudy world of the byte bandit: A postmodernist interpretation of the computer underground. En *Computers in criminal justice* (pp. 31–67). Bristol: Wyndham Hall. Disponible en `http://scholar.google.com/scholar?hl=en\&btnG=Search\&q=intitle:A+Postmodernist+Interpretation+of+the+Computer+Underground\#3`

Meyer, G. R. (1989). *The social organization of the computer underground.* Tesis Doctoral no publicada, Northern Illinois University.

Mishkin, N. (1983). Hackers Vindicated. *Science News*, *124* (23), 355. Disponible en `http://www.jstor.org/stable/3968225`

Mitnick, K., y Simon, W. (2002). *The art of deception. Controlling the human element of security.* Indianapolis: Wiley publishing.

Mitnick, K., y Simon, W. (2010). *Ghost in the wires: My adventures as the world's most wanted hacker.* New York: Little, Brown and company.

Morley, D. (1998). *El posmodernismo: una guía básica.* Paidós Ibérica. Disponible en `http://dialnet.unirioja.es/servlet/articulo?codigo=2127187`

Mungo, P. (1992). *Approaching zero.* London: Faber and Faber.

Oré, C. (2007). Reseña de "Me llamo Kohfam. Identidad hacker: Una aproximación antropológica"de Pau Contreras. *Papeles del CEIC*, *1* (Marzo), 1–6. Disponible en `http://www.ehu.es/CEIC/pdf/critica2.pdf`

Ortiz Marín, M. (2010). *Cómo hacer una rica sopa con la metodología: caminos y veredas de la investigación en ciencias de la comunicación.* Mexicali: Universidad Autónoma de Baja California.

Polgár, T. (2008). *Freax, the brief history of the computer demoscene. Volume 1.* Winnenden: CSW-Verlag.

Raymond, E. S. (1998). *Goodbye, 'free software'; hello, 'open source'.* Disponible en `http://www.catb.org/~esr/open-source.html`

Raymond, E. S. (1999). *The new hacker's dictionary.* Cambridge: The MIT press.

Raymond, E. S. (2001). *The cathedral and the bazaar, musings on Linux and open source by and accidental revolutionary.* Sebastopol: O'Reilly.

Reckwitz, A. (2002). Toward a Theory of Social Practices A Development in Culturalist Theorizing. *European Journal Of Social Theory*, *5*(2), 243–263.

Ritzer, G. (2005). *Teoría sociológica clásica.* México, D.F.: McGraw Hill.

Rosenzweig, R. (1998, diciembre). Wizards, Bureaucrats, Warriors, and Hackers: Writing the History of the Internet. *The American Historical Review*, *103*(5), 1530.

Rosteck, T. (1994). Computer Hackers: rebels with a cause. En *Montreal: Concordia university, dept. of sociology and anthropology, honours seminar-soci* (Vol. 409).

Ruiz Torres, M. A. (2008). Ciberetnografía: comunidad y territorio en el entorno virtual. En *Epistemologías y metodologías: perspectivas antropológicas* (pp. 373–404). Servicio de Publicaciones. Disponible en `http://www.euskomedia.org/PDFAnlt/antropologia/11/05/05117132.pdf`

Rutiaga, L. (2004). Prólogo. En *Lo bello y lo sublime: fundamentación de la metafisica de las costumbres* (pp. 5–10). México, D.F.: Grupo Editorial Tomo.

Rybas, N., y Gajjala, R. (2007). Developing cyberethnographic research methods for understanding digitally mediated identities. En *Forum qualitative sozialforschung/forum: Qualitative social research* (Vol. 8, pp. 1–15). Disponible en `http://www.qualitative-research.net/index.php/fqs/article/viewArticle/282`

Shimomura, T., y Markoff, J. (1996). *Takedown. The pursuit and captur of Kevin Mitnick, America's most wanted computer outlaw-by the man who did it.* New York: Hyperion.

Sinnott, E. (2007). Introducción. En *Ética* (pp. VII–LXXVI). Buenos Aires: Ediciones Colihue.

Softley, I. (1995). *Hackers.* United Artists.

Stallman, R. (1996). *La definición de software libre.* Disponible en `http://www.gnu.org/philosophy/free-sw.es.html`

Stallman, R. (2009). *Free Software, Free Society: Selected Essays of Richard M. Stallman.* Boston: CreateSpace.

Sterling, B. (1992). *The hacker crackdown. Law and disorder in the electronic frontieer.* New York: Bantam books.

Stoll, C. (1989). *The cuckoo's egg.* Doubleday.

Swift, J. (1980). Modesta proposición. En *Ensayos científicos* (2da. ed.). CONACYT.

Taylor, P. A. (1999). *Hackers.* New York: Routledge.

Thomas, D. (2002). *Hacker culture* (primera ed.). Minneapolis: University of Minessota Press.

Torvalds, L., y Diamond, D. (2001). *Just for fun, the story of an accidental revolutionary.* New York: Harper Business.

Touraine, A. (2000). *Crítica de la modernidad* (segunda ed.). México, D.F.: Fondo de cultura económica.

Valles, M. S. (1999). *Tecnicas cualitativas de investigacion social: Reflexión metodológica y práctica profesional.* Madrid: Síntesis.

Verton, D. (2002). *The hacker diaries: Confessions of teenage hackers.* Berkeley: McGraw-Hill/Osborne.

Wark, M. (2004). *A hacker manifesto.* Cambridge: Harvard university press.

Wayner, P. (2000). *Free for all: how Linux and the free software movement undercut the high-tech titans.* New York: HarperBusiness. Disponible en `http://books.google.com.mx/books?id=gLxQAAAAMAAJ`

Weber, M. (2008). *La ética protestante y el espíritu del capitalismo.* Barcelona: Ediciones Península. Disponible en `http://books.google.com/books?hl=en\&lr=\&id=gJNS8\_3zyTQC\&oi=fnd\&pg=PA7\&dq=La+\'{e}tica+protestante+y+el+esp\'{\i}ritu+del+capitalismo+Max+Weber\&ots=N1t\_fesTNl\&sig=pTuW3in05tpYZ5kdCEOT7IBbNfE`

Weber, S. (2005). *The success of open source.* Harvard University Press.

Williams, R. (1983). *Sociología de la cultura.* Barcelona: Paidós.

Williams, S. (2002). *Free as in freedom: Richard Stallman's crusade for free software.* Cambridge: O'Reilly. Disponible en `http://books.google.com.mx/books?id=LUJcSjkYEikC`

Zittrain, J. (2008). *The future of the Internet and how to stop it.* Harrisonburg: Yale university press. Disponible en `http://adam-hazdra.webz.cz/download/zittrain\_future.pdf`

Índice alfabético