

Historia del software libre

Movimientos Open Access (Máster Propiedad Intelectual EOI)

Miguel Vidal – Juan José Amor

GSyC/LibreSoft

8 de mayo de 2010



© 2010 Miguel Vidal - Juan José Amor.
Algunos derechos reservados. Este artículo se distribuye bajo la licencia
“Reconocimiento-CompartirIgual 3.0 España” de Creative Commons, disponible
en <http://creativecommons.org/licenses/by-sa/3.0/es/deed.es>

Agenda

- Qué es el software libre
- Historia del software libre
- Quién es quién

Qué es el software libre

¿Por qué estudiar la historia del software libre?

- Permite comprender de dónde proceden los principios de la cultura libre
- Permite conocer un modelo de desarrollo/innovación exitoso alternativo al modelo restrictivo y cerrado.
- Permite situar históricamente algunos conflictos en torno a la propiedad intelectual.
- Permite comprender por qué las TIC han llegado a ser como son: el PC, Internet...
- Permite despejar algunos malentendidos sobre los orígenes de la informática moderna.

¿Qué es el software libre?

Quien lo recibe puede:

- usarlo como mejor le parezca, donde mejor le parezca.
- redistribuirlo a quien quiera, por los medios que quiera.
- modificarlo (y mejorarlo o adaptarlo).
- redistribuir las modificaciones
- Imprescindible: disponibilidad del código fuente.

Software libre NO es igual a software gratis

¿Qué es el software libre?

Quien lo recibe puede:

- usarlo como mejor le parezca, donde mejor le parezca.
- redistribuirlo a quien quiera, por los medios que quiera.
- modificarlo (y mejorarlo o adaptarlo).
- redistribuir las modificaciones
- Imprescindible: disponibilidad del código fuente.

Software libre NO es igual a software gratis

¿Por qué es esto y no otra cosa?

- Tradición (científica, académica, cultura hacker).
- Definición fijada por Stallman en 1984 (Manifiesto GNU).
- Plasmado en las primeras licencias libres de software (GPL, 1989)
- Definición asumida por todas las variantes del movimiento de software libre.
- Tres definiciones canónicas: FSF, OSI y DFSG

Comparemos las definiciones

- **FSF:** <http://www.gnu.org/philosophy/free-sw.html>
- **DFSG:** http://www.debian.org/social_contract.es.html#guidelines
- **OSI:** <http://www.opensource.org/docs/osd>

Consecuencias de la libertad del software

- **Coste:** modelo de costes radicalmente distinto al del software privativo.
- **Apertura:** puede modificarse, puede inspeccionarse, puede estudiarse.
- **Distribución:** nuevos canales, nuevos métodos.
- **Desarrollo:** modelos de desarrollo “sorprendentes”.
- **Mantenimiento y soporte:** Verdadera competencia.
- Combinación de dos poderosos mecanismos:
 - **Competencia** (pudiendo usar el mismo programa base)
 - **Cooperación** (incluso involuntaria)

Diversos actores, diversas perspectivas

- El usuario final (individual o empresa).
- El desarrollador (o productor de software).
- El integrador.
- El que proporciona mantenimiento y servicios.

¿Software libre “underground” ?

- Nacido en el corazón mismo de la economía capitalista.
- Antimonopolios (un mal de la industria del software): Libertad y competencia como pre-requisitos.
- Bien público pero no estatal (*commons*).
- Se basa también en:
 - la libre concurrencia y la cooperación social (libre mercado)
 - el rechazo a la apropiación de la fuentes de innovación

Competencia + Cooperación: es distinto de una “alternativa no mercantil” .

¿Software libre “underground” ?

- Nacido en el corazón mismo de la economía capitalista.
- Antimonopolios (un mal de la industria del software): Libertad y competencia como pre-requisitos.
- Bien público pero no estatal (*commons*).
- Se basa también en:
 - la libre concurrencia y la cooperación social (libre mercado)
 - el rechazo a la apropiación de la fuentes de innovación

Competencia + **Cooperación**: es distinto de una “alternativa no mercantil” .

Impacto del software libre

- Empresas como IBM, Oracle, Amazon, Yahoo, Google, la NASA o el centro de control aereo de USA usan y desarrollan software libre.
- Gobiernos como Brasil, Finlandia, Noruega, China, Extremadura, Andalucía. Ayuntamiento de Munich.
- Hay directivas europeas y leyes españolas (11-2007 LAECSP) que instan al uso de código abierto en las instituciones públicas.

Un nuevo movimiento social

Es también un nuevo movimiento social basado en:

- La proliferación de los ordenadores personales.
- La universalización del acceso a Internet.
- La ética hacker, la cooperación horizontal y la “inteligencia colectiva”.

¿Hay conclusiones?

- Aún no hay suficiente casuística para estar seguros de la tendencia futura.
- El modelo es viable económica y técnicamente.
- El modelo favorece a los más competitivos.
- El modelo allana el terreno para los pequeños.
- Se está experimentando mucho: nuevos modelos de desarrollo, de negocio, de atención al usuario, de política tecnológica...
- Es un campo con mucha innovación, donde es preciso un buen (y actualizado) conocimiento del entorno.

¿Hay conclusiones? (2)

Este es uno de esos raros momentos en los que toda una industria puede estar cambiando de paradigma.

“El paradigma de las TIC proporciona las bases materiales de su extensión al resto de la sociedad” (Manuel Castells)

¿Hay conclusiones? (2)

Este es uno de esos raros momentos en los que toda una industria puede estar cambiando de paradigma.

“El paradigma de las TIC proporciona las bases materiales de su extensión al resto de la sociedad” (Manuel Castells)

Historia del software libre

Historia del software libre

- Se remonta a los orígenes de la informática.
- Como movimiento consciente, nace en 1984 con el Proyecto GNU.
- En los 90, Linux y el modelo bazar suponen su culminación.

Orígenes: “Real Programmers”

- 1945 hasta 1970: los *Real Programmers* fueron la cultura técnica dominante en el ámbito de la computación.
- Desde el primer computador ENIAC, existió una cultura técnica, consciente de sí misma, creaban y manipulaban software por pura diversión.
- Los *Real Programmers* eran ingenieros o físicos, y a menudo radioaficionados.
- Seymour Cray, diseñador de la gama de supercomputadoras Cray, fue uno de los más brillantes.
- Son los pioneros de la informática actual.

Un “Real Programmer”: Seymour Cray



Figura: Seymour Cray, con un supercomputador CDC 1604 diseñado por él, en 1958

Las décadas de 1950 y 1960

- Durante los años 1960 el software venía como acompañante del hardware, no se considera un elemento independiente.
- El software se distribuía con su código fuente: grupos de usuarios lo comparten, y lo mejoran.
- Código fuente a disposición de quien lo pide: los clientes generalmente no pagan por el software.
- Relación con el software muy parecida a la que hoy tenemos con el software libre.
- Todo cambia con el *unbundling* de hardware, software y servicios de IBM (1969)

Los primeros hackers

- 1961: el MIT adquiere la primera PDP-1. La usa el Tech Model Railroad Club (TMRC), núcleo del IA Lab del MIT.
- La cultura en torno a las computadoras del MIT adopta el término “hacker” y crean su propio SO para PDP-10 (ITS, “Incompatible Timesharing System”, sin permisos ni contraseñas).
- Allí se forma Stallman, y surge la cultura de Arpanet (*jargon file*).
- ARPANET (principalmente una red de computadoras DEC) interconecta a hackers de toda Norteamérica y es la génesis de Internet.
- Otro nodo importante en la cultura hacker fue el PARC de XEROX, en Palo Alto (California): allí se inventaron el ratón moderno, las ventanas y la interfaz gráfica de iconos.

¿Hackers?

¿Piratas informáticos?

Definición original de hacker

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

¿Hackers?

¿Piratas informáticos?

Definición original de hacker

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

Ética hacker

- Un buen programador debería contribuir con su trabajo a la comunidad
- Un buen programador debería poder aprovechar el trabajo de otros buenos programadores
- Un buen programador debería poder “arreglar” y mejorar cualquier programa
- Un buen programador debería sentirse orgulloso de su propio código y de que otros lo usen, sin otras contraprestaciones.

Buen programador == hacker

Ética hacker

- Un buen programador debería contribuir con su trabajo a la comunidad
- Un buen programador debería poder aprovechar el trabajo de otros buenos programadores
- Un buen programador debería poder “arreglar” y mejorar cualquier programa
- Un buen programador debería sentirse orgulloso de su propio código y de que otros lo usen, sin otras contraprestaciones.

Buen programador == **hacker**

Años setenta

- El software empieza a ser privativo “por defecto”
- Esfuerzos “aislados”: TeX, Spice, etc.
- En general, el objetivo es hacer una herramienta determinada
 - A veces, motivos éticos (ej: costumbre en la comunidad matemática)
 - A veces, motivos prácticos (ej: difusión de una nueva tecnología)

Años setenta: El surgimiento de Unix

El nacimiento de Unix, una auténtica revolución del software:

- 1969: Ken Thompson inventó Unix (mismo año que Arpanet).
- Surge de los desechos de Multics, en AT&T (Bell Labs).
- Dennis Ritchie inventa un nuevo lenguaje llamado C para usarlo en el Unix de Thompson.
- Primer sistema operativo portable y modular (KISS), frente a anteriores sistemas incompatibles y costosos.
- Se extiende rápidamente y de forma no oficial por AT&T. Y por Arpanet (hardware distinto, gracias a C).
- Acuerdo judicial (*antitrust*) de 1956 impide a AT&T comercializar Unix: debe licenciarlo (con fuentes) a quien se lo solicite.

Años setenta: Unix y Berkeley

- CSRG (Computer Systems Research Group) de Berkeley:
 - Importancia de compartir fuentes (cultura Unix “original”).
 - Limitado por la licencia AT&T (poco desde el punto de vista práctico, todos la tenían).
 - Financiado por DARPA (DoD).
 - Utilizado por mucho software propietario (SunOS, Ultrix, etc.)
- Primera Internet:
 - Implementaciones de referencia, disponibles para todos: la base de los estándares actuales.
 - La Red como herramienta de cooperación (News, ftp, e-mail).
 - La comunidad de usuarios proporciona el mejor soporte.
 - Falso mito de los ataques nucleares.

Historia de Unix

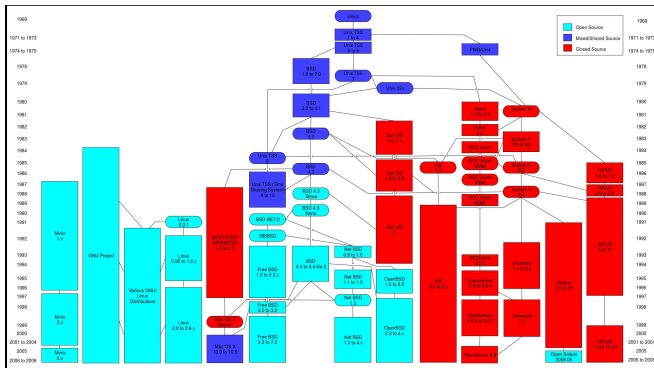


Figura: Historia de Unix.

Años setenta: Nacen los PCs

- El primer computador personal (Altair 8800) sale al mercado en 1975 como “kit”.
- Se les llamaba Microordenadores. Su lenguaje era el BASIC.
- Atrajo a otra nueva generación de jóvenes hackers libertarios: “computers for the people”.
- Nace una industria: Apple se fundó en 1977. Microsoft en 1975 (para vender intérpretes de Basic a los usuarios de Altair).
- “Carta abierta a los aficionados”
- La gran industria lo ignora hasta muy tarde: IBM lanza su PC en 1981.

La revolución de los microordenadores



Figura: Portada de *Popular Electronics*, enero de 1975.

La revolución de los microordenadores



Figura: El MIPS Altair 8800 con disquetera de 8".

Resumen: Los años setenta y la cultura hacker

Confluyen tres grandes subculturas hacker a finales de los setenta, en torno a tecnologías muy dispares:

- **La cultura de las PDP-10 y ARPANET**, ligada a TOPS-10, a LISP (SO del PDP-10), Macro (su lenguaje ensamblador), a ITS, al MIT y al SAIL (IA Lab de Stanford);
- **Las gentes de Unix y C** con sus PDP-11, sus VAXen y sus conexiones telefónicas rudimentarias (UUCP). Berkeley y Bell Labs.
- **Aficionados de los primeros microordenadores**, decididos a acercar el potencial de las computadoras “al pueblo”. *Popular Electronics*, Altair, Basic, Apple...

Movimiento contemporáneo del software libre

Años ochenta: El fin de los viejos tiempos

- 1983: DEC cancela la línea PDP-10. ITS ya no tiene futuro (no era portable).
- Se extienden los acuerdos de no-divulgación
- Comienza a despegar la gran industria del software privativo, basada en el secreto (binarios), en la venta de licencias y en la privatización de los fuentes.
- Unix e Internet: choca el modelo privativo (AT&T) contra el modelo abierto (BSD).

Declina la ética hacker

Stephen Levy, en *Hackers: Heroes of the Computer Revolution* (1984), acuña la expresión “ética hacker” de forma retrospectiva:

- 1 Acceso ilimitado a los ordenadores y a todo aquello que puede enseñarte algo.
- 2 Toda la información debe ser libre
- 3 Es necesario promover la descentralización
- 4 Los hackers no deben ser juzgados por sus títulos académicos, su edad o posición.
- 5 Se puede crear belleza con una computadora.
- 6 Los ordenadores pueden cambiar la vida a mejor.

El software libre es el heredero directo de estos principios.

Declina la ética hacker

Stephen Levy, en *Hackers: Heroes of the Computer Revolution* (1984), acuña la expresión “ética hacker” de forma retrospectiva:

- 1 Acceso ilimitado a los ordenadores y a todo aquello que puede enseñarte algo.
- 2 Toda la información debe ser libre
- 3 Es necesario promover la descentralización
- 4 Los hackers no deben ser juzgados por sus títulos académicos, su edad o posición.
- 5 Se puede crear belleza con una computadora.
- 6 Los ordenadores pueden cambiar la vida a mejor.

El software libre es el heredero directo de estos principios.

Años ochenta: El Proyecto GNU

- Stallman abandona el MIT en 1984 para poder dedicarse al Proyecto GNU (GNU's Not UNIX!).
- 1985: Stallman publica el Manifiesto GNU: sienta los fundamentos éticos del software libre
- Meta: construir un sistema completo libre, alternativo a Unix.
- Crea la infraestructura básica: editor (emacs), compilador (gcc), depurador (gdb), gmake...
- Crea la Fundación de Software Libre (1985) para apoyar el Proyecto GNU.
- Fundamentos legales: la GPL (1989)
- Trabajo muy estructurado y con metas claras.
- A principios de los 1990 GNU tenía su sistema casi completo, faltaba el núcleo.

Final de los 1980, primeros 1990

CSRG de Berkeley:

- Liberaron la parte de UNIX (implementación de TCP/IP) que desarrollaron ellos, no AT&T (Net/1, 1989)
- Reescribieron el código del UNIX original que no era suyo y liberaron el código (Net/2, 1991)
- Los hermanos Jolitz portan el código a i386 como 386BSD, liberado por Internet con licencia BSD.
- Rápidamente: sistemas completos, similares a SunOS en funcionalidad.
- Importancia de X Window (MIT): cientos de individuos de decenas de empresas colaborando.

El juicio AT&T vs BSD

- USL (AT&T) denuncia a la Universidad de Berkeley (1992) por explotar Unix.
- Berkeley contraataca denunciando a AT&T por incumplir la licencia BSD (la menos restrictiva del mundo).
- Berkeley gana el litigio, USL es vendido a Novell y llegan a un acuerdo en 1993.
- Pero juicio deja exhausto a BSD, supone un retraso de dos años en un momento crítico...
- Otro proyecto sin problemas legales empieza a adquirir masa... el núcleo Linux.
- Tras el juicio, se libera una última versión completa de Unix BSD y el CSRG de Berkeley desaparece.

La herencia de BSD

- Desde la distribución de 386BSD el desarrollo es rápido y se consigue un sistema estable.
- Las distribuciones NetBSD, FreeBSD y OpenBSD surgen a partir de la adaptación original de 386BSD.
- Modelo catedral, en paralelo al desarrollo de Linux.

Los años noventa: El nacimiento de Linux

- Linux es un kernel
- Lo inicia Linus Torvalds, en 1991, y *just for fun*
- Existían ya sistema operativos libres casi completos (GNU y Unix BSD)
- Desde que liberó la primera versión (0.01) se van uniendo cientos de desarrolladores
- Se adopta la licencia GPL
- Marzo 1994: versión 1.0

Los años noventa: GNU/Linux

- Linux es *solo* un kernel: necesita algo más para funcionar.
- Al proyecto GNU le falta un núcleo en 1990.
- Desarrollo del proyecto Hurd, arquitectura de microkernel (Mach): sin resultados
- Se adopta *temporalmente* como núcleo para GNU
- Proliferan las distribuciones GNU/Linux: Slackware, Debian, Red Hat, SuSE, Gentoo, etc.

Los años noventa: el modelo bazar

- La principal aportación de Linu[xs]: su modelo de desarrollo, el llamado “modelo bazar”
- Gran número de voluntarios coordinados a través de Internet.
- La calidad se mantenía, no con estándares rígidos o autocracia, sino publicando cada semana y obteniendo el *feedback* de cientos de usuarios pocos días.
- “*Release Early, Release Often* (and listen to your customers)”: propicia selección darwiniana rápida sobre las mutaciones presentadas por los desarrolladores.
- Para sorpresa de casi todo el mundo, esto funcionó bastante bien.
- “La ventaja más importante de Linux no fue de carácter técnico, sino sociológico” (ESR – CATB).

Finales de los 1990

- Netscape anuncia la liberación del código de su navegador:
- “La catedral y el bazar”.
- Cada vez más cerca del usuario estándar: KDE, GNOME.
- GNU/Linux penetra en Universidades (y en casa de los estudiantes).
- La mejor opción es libre en muchos ámbitos (Apache, infraestructura de Internet, XFree, GCC, Gnat).
- Empresas como RedHat consiguen capital-riesgo. Nasdaq: IPOs de récord.
- La prensa comienza a atender al software libre: compite con Windows NT.
- Grandes empresas tecnológicas invierten en Software Libre: HP, IBM, Sun, Google, Yahoo!...

Principios de los 2000: Madurando poco a poco

- El software libre empieza a estar listo para el escritorio (GNOME 2.x, KDE 3.x, OpenOffice), y es simple de instalar por el usuario final.
- El software libre se incorpora a la estrategia de grandes empresas (IBM, HP, Sun)
- Otras (como Microsoft) prefieren una estrategia de enfrentamiento parcial (FUD).
- Dificultades financieras como resultado de la crisis de las puntocom
- Comienza la penetración en Administraciones públicas y grandes empresas
- Aumento grande del número de desarrolladores, de la cantidad de software libre disponible, etc.

Principios de los 2000: Madurando poco a poco

Productos con éxito:

- Servidores: Apache, Postfix, Tomcat, Proftpd...
- Navegadores: primero Mozilla, luego Firefox...
- Correo: Thunderbird, Evolution, Kmail...
- Ofimática: OpenOffice.org, Koffice, AbiWord...
- Escritorio: KDE, Gnome, Compiz/Beryl...
- Sistemas Operativos: Sun libera Solaris (2005) y todas sus tecnologías punteras (ZFS, DTrace, etc.).
- Formatos abiertos: ODF (ISO/IEC 26300), OGG...

Principios de los 2000 (2)

- Nuevas disciplinas estudian el software libre: comenzamos, poco a poco, a entender cómo funciona
- Comienzan a verse efectos de la “deslocalización” del desarrollo de software libre: países periféricos hacen cosas interesantes.
- Ciertos mercados, ciertos sectores, ya consideran al software libre como una opción natural
- El entorno legal va cambiando de forma ambivalente: ¿se convertirá en hostil para el software libre?

Actualidad (finales de los 2000)

- Software libre es estratégico para muchas empresas (ej: Google)
- Conjuntos de aplicaciones muy completos para muchos entornos
- Empresas probando nuevos modelos de colaboración (ej: ObjectWeb, Morfeo)
- Software libre como propuesta para dominar mercados (ej: Android, Symbian, Maemo en móviles)
- Nuevos modelos de negocio, modelos para nuevos negocios
- Software libre parte del análisis de competencia en sectores (ej: MySQL en la compra de Sun por Oracle)
- Software libre se analiza en las Escuelas de Negocios. ;-)
- El software libre se va convirtiendo en algo “normal”.

El futuro: ¿una carrera de obstáculos?

La evolución futura del software libre se encuentra con varios obstáculos:

- Técnicas FUD (miedo, incertidumbre, duda): hasta ahora han mostrado no ser muy problemáticas.
- Disolución: confusión (llamar libre a lo que no lo es), división de la comunidad, pérdida de las ventajas del modelo...
- Desconocimiento (pérdida de visión): ¿por qué es interesante el software libre?
- Impedimentos legales y tecnológicos: patentes de software, mecanismos de control de acceso a la información, leyes contrarias (anti-elusión), etc.
- ¿Cuán sostenible es el desarrollo de software libre?

Quién es quién

Quién es quién

Porque el Software Libre también tiene sus héroes.

Quién es quién



Linus Torvalds

- Finlandés.
- Creador de la primera versión del núcleo Linux.
- En enero de 1992 adoptó la GPL.
- Prototipo de la visión ingenieril y pragmática del software libre.
- Tiene un asteroide con su nombre.

¿Fue determinante que usara la licencia GPL?

Linus Torvalds

- Finlandés.
- Creador de la primera versión del núcleo Linux.
- En enero de 1992 adoptó la GPL.
- Prototipo de la visión ingenieril y pragmática del software libre.
- Tiene un asteroide con su nombre.

¿Fue determinante que usara la licencia GPL?

Quién es quién



Richard Stallman

- Nexo con la vieja cultura hacker.
- Desarrollador: emacs, gcc, gdb...
- Creador del Proyecto GNU.
- Creador de la GPL, de la FSF.
- "Inventor" del concepto de copyleft.
- Faro estratégico e incansable promotor de la visión ética/filosófica del software libre.

¿Existiría el Software Libre sin Stallman?

Richard Stallman

- Nexo con la vieja cultura hacker.
- Desarrollador: emacs, gcc, gdb...
- Creador del Proyecto GNU.
- Creador de la GPL, de la FSF.
- "Inventor" del concepto de copyleft.
- Faro estratégico e incansable promotor de la visión ética/filosófica del software libre.

¿Existiría el Software Libre sin Stallman?

Quién es quién



Eric Raymond

- Autor de “La catedral y el bazar” y los primeros ensayos antropológicos y culturales sobre la comunidad.
- Desarrollador de fetchmail, bogofilter.
- Memoria de la cultura hacker (*jargon file*)
- Co-fundador de la Open Source Initiative.
- Visión libertaria y pragmática del software libre.

¿Hubiera liberado código Netscape o SUN sin que Raymond escribiera “La catedral y el Bazar”?

Eric Raymond

- Autor de “La catedral y el bazar” y los primeros ensayos antropológicos y culturales sobre la comunidad.
- Desarrollador de fetchmail, bogofilter.
- Memoria de la cultura hacker (*jargon file*)
- Co-fundador de la Open Source Initiative.
- Visión libertaria y pragmática del software libre.

¿Hubiera liberado código Netscape o SUN sin que Raymond escribiera “La catedral y el Bazar”?

Quién es quién



Alan Cox

- Desarrollador de Linux desde 1991
- Primera instalación de Linux importante
- Ramas 2.2 y 2.4 del kernel
- Activista contra patentes de software
- Trabaja para Red Hat

Quién es quién



Bruce Perens

- Segundo *Project Leader* de Debian (1996-97). Fundó la SPI. Escribió las DFSG y el Contrato Social de Debian.
- Creó la Definición de Open Source (OSD).
- Co-fundador junto a Raymond de la Open Source Initiative.
- Presentó el Open Source/Free Software ante Naciones Unidas.
- Evangelizador del software libre y su viabilidad económica en el ámbito empresarial.

Quién es quién



Ian Murdock

- Fundó Debian GNU/Linux (1993) y su manifiesto: compromiso social, mantenida por la comunidad, y una de las distribuciones más importantes
- Emprendedor.
- Fichó por Sun Microsystems en 2007 para liderar el Proyecto Indiana, que produjo la primera distribución binaria de OpenSolaris.

Quién es quién



Miguel de Icaza

- Fundador de Gnome, creador de Midnight Commander, Gnumeric, componentes Bonobo, Mono (reimplementa plataforma .NET)
- Fundador de Ximian
- Ahora vicepresidente de desarrollo de Novell
- Polémico por su defensa de tecnologías Microsoft.

Quién es quién



Larry Wall

- Lingüista y de fe cristiana (suele aludir a ello en sus conferencias).
- Creador del lenguaje de programación Perl (pegamento de Internet por muchos años).
- Lidera desde hace dos décadas la comunidad en torno a perl.
- Creador del programa patch, usado universalmente, y de rn (viejo lector de news).
- Hacker *tout-court*, ganador dos veces del concurso de C ofuscado (IOCCC).
- Visión filosófica de la programación: “There’s More Than One Way to Do It”