

PROPIEDAD, S.L.: Propiedad y Software Libre

Autor José J. Grimaldos

Ponencia presentada en las III Jornadas Andaluzas de Software Libre, celebradas en Granada el 14 y 15 de Noviembre de 2003

Resulta ya tópico decir "los tiempos están cambiando", "cada día surgen nuevos retos",... pero, tal vez como ocurre con la mayoría de tópicos, algo tienen de verdad.

En el caso del propósito de estas reflexiones, qué duda cabe, cualquier observador curioso debe sentirse intrigado ante el devenir del fenómeno asociado al software libre. No pretendemos, en absoluto, una tarea de divulgación de esta ¿filosofía?, tampoco es nuestra meta el proselitismo, ni tan siquiera la formulación de unas tesis de contenido político, económico, jurídico o social; sólo pretendemos acercarnos con cierta curiosidad a este movimiento y plantear algunas reflexiones acerca de sus implicaciones en el pensamiento actual, los valores educativos que encierra, en definitiva, más dudas que certezas.

Para ello, huiremos de los profundos, y desconocidos para el autor, análisis académicos o técnicos donde el software libre está siendo objeto de análisis, estudios, ensayos y... hasta simulaciones iterativas usando modelos matemáticos inspirados en la "Teoría de Juegos" y trataremos de enfocar el tema en los valores educativos que encierra el software libre en sí mismo, desde las necesarias precisiones sobre el concepto de propiedad que sitúen con claridad el carácter de libre.

Clarificando el lenguaje

En principio, hemos de aclarar ciertos vocablos, la mayoría heredados o adaptados del idioma inglés, dado el carácter tecnológico del tema, a fin de que la semántica no sea un problema añadido, y con objeto, además, que la persona no muy familiarizada con la informática y su particular jerga tenga la posibilidad de acceder sin dificultad a las ideas que se exponen.

Tal vez, desde principios de siglo, la información ha alcanzado tal dimensión que el exceso de la misma requiere de herramientas auxiliares para ser procesada. En este sentido, las técnicas de computación han auxiliado al hombre en esta tarea hasta el punto de convertirse casi en imprescindibles hoy día. Bien es cierto que, en honor a la verdad, no son condición necesaria ni, por supuesto, suficiente para que un estudioso, investigador o un simple ciudadano, sin más pretensiones, acceda al conocimiento que desee, necesite o busque. En cambio, sí es habitual que la mayoría de personas, en mayor o menor medida, acudan a los ordenadores cotidianamente para aliviar parte de su actividad.

Un ordenador es, simplemente, un conjunto de piezas ensambladas en torno a un procesador con una gran capacidad operativa, aritmética y lógica. Sin embargo, todos estos componentes, convenientemente conectados, son incapaces de realizar tarea alguna por sí mismos, necesitan un soporte lógico que les dote de cohesión y les haga funcionar coordinadamente con algún propósito específico.

Es necesario además que el ser humano tenga la posibilidad de comunicarse con él, por una parte para demandarle el tipo de procesamiento deseado y por otra, para recibir el resultado del proceso. Básicamente necesitamos dos tipos de programas: Uno específico que sea capaz de realizar la tarea concreta que pueda demandar un usuario (procesar texto, gestionar una base de datos,...) y otro genérico que cohesione los diferentes elementos físicos del ordenador y, a su vez, se entienda con el programa específico utilizado por el usuario. Este último tipo de programa se conoce con el nombre de sistema operativo y a los primeros se les llama aplicaciones. Ambos se construyen generalmente sobre una secuencia de comandos en un determinado idioma que constituyen el código fuente de cada programa.

El código fuente

Sobre este código fuente gira toda la controversia. Las personas con los conocimientos necesarios para escribirlo son conocidas como hackers o programadores y, es conveniente distinguirlos de los crackers pues éstos últimos son también programadores de grandes conocimientos, pero con intenciones maliciosas o dañinas que a los primeros no se les suponen.

El código fuente de un programa es, por tanto, un simple texto que contiene las instrucciones para que el procesador, junto a los periféricos, realicen una determinada tarea, sea ésta simple o de gran complejidad. Sin embargo, para que las órdenes que contiene sean comprendidas por la máquina es necesario traducirlo (compilarlo) a un lenguaje denominado genéricamente código máquina que es propio de la estructura interna que ésta posea; una vez realizado este paso, el texto que inicialmente contenía el código fuente deja de ser legible para el humano y pasa a ser accesible sólo para la máquina. Además, este código fuente permite ser cifrado y, por tanto, ocultado ante lectores no deseados.

Es lo que ocurre con los programas comerciales, donde únicamente el propietario posee el código fuente de éste y sólo a él le está permitido hacer modificaciones en el comportamiento de su programa. Un poco de historia

Pese a que en la actualidad, el fenómeno del software libre está generando un eco y una repercusión considerable, lo cierto es que su génesis podría datarse casi en los inicios de la computación moderna. No es fácil señalar una fecha ni un lugar concretos, como es lógico, raras veces las ideas, las tendencias o los movimientos sociales vienen acompañadas de partidas de nacimiento en toda regla y este caso no es una excepción.

El primer sistema operativo digno de recibir ese nombre, tal y como lo concebimos hoy, nace en los laboratorios Bell fruto de la curiosidad investigadora del ser humano. Nació libre y fue bautizado con el nombre de Unix, al poco tiempo pasó a ser comercial puesto que fue patentado, registrado y distribuido por una considerable cantidad de dinero, sólo al alcance de las grandes firmas comerciales. Aún quedaba muy lejos el PC doméstico. Por aquel entonces los computadores estaban reclusos en los centros de cálculo e investigación, en los departamentos contables de las grandes organizaciones comerciales y centros similares. La informática era, pues, una disciplina científica de aplicación, todavía, lejos del gran público, lejos de la sociedad y oficiada por una privilegiada élite de investigadores capaces de instruir al computador.

El movimiento GNU

Un grupo de hackers, abanderados por Richard Stallman (RMS) se rebelaron ante esta situación porque defendían que el software era conocimiento científico y como tal no debía ser patentado ni comercializado, sino todo lo contrario, difundido como motor del avance del progreso.

Se consideraban, ante todo, investigadores, y como tales, frecuentaban el libre intercambio de información entre la comunidad científica. Con este objetivo crearon la Fundación para el Software Libre (FSF) con el objeto de recaudar fondos que permitieran la creación y distribución de programas de código abierto. Así nacieron distintas aplicaciones de software libre (como Emacs, gcc,...) que aún hoy día gozan de gran difusión.

Para ajustarse al marco legal vigente e impedir que sus productos fueran patentados y comercializados por alguna empresa que se apropiase de su código libre, la FSF los protegió bajo la Licencia Pública General (GPL), una ingeniosa fórmula de copyright expansivo, conocido como copyleft, en contraposición a las restricciones de las licencias de copia habituales.

Las condiciones de esta licencia, calificada también como contagiosa, se aseguran que un producto amparado por ella no evolucionará a versiones propietarias o restrictivas, pues hereda las características de la licencia original. Esta astuta forma de proteger sus creaciones ha convertido a la FSF en el foco más significativo de la comunidad del software libre durante varios años, sobre todo, en la década de los 80 y principios de los 90, llegando a crear lo que muchos han calificado de ideología hacker. Linux entra en escena

Paralelamente, Linus Torvald, un estudiante de informática finlandés, escribía, a principios de los años 90, un sistema operativo compatible con Unix con el objeto de poder practicar en su ordenador doméstico.

Este rudimentario sistema se llamó Linux y su creador difundió el código fuente en la red Internet para que fuese probado y mejorado. Este hecho causó una revolución sin precedentes en el ámbito informático, propiciando que muchísimos hackers se entusiasmaran con el proyecto hasta el punto que sobre el año 93, Linux era ya un sistema operativo casi tan potente y robusto como su inspirador Unix y superaba holgadamente en muchos aspectos a la mayoría de sistemas comerciales.

No sería esa la única expansión que sufriría. A finales de los años 90, con la integración de los entornos gráficos de escritorio más avanzados (fundamentalmente GNOME y KDE) Linux abandonó los círculos propios de hackers para extenderse al usuario final, convirtiéndose así en una seria amenaza para los intereses comerciales de los fabricantes de programas de código cerrado, siendo en este momento cuando la polémica salta también a las tribunas de la sociedad.

Sobre la propiedad

El fenómeno del software libre está poniendo en cuestión numerosos conceptos normalmente asumidos, unas veces de manera natural, otras, no tanto, en la sociedad occidental actual. El primero en tambalearse visiblemente es el de propiedad, en el sentido genuino del término y, como efectos colaterales, los derechos de copia, propiedad intelectual, libertad de información, así como otras cuestiones menores como libertad de mercado, monopolios, etc.

Siempre que usamos el concepto de propiedad, en general, tendemos a la identificación de un bien material o inmaterial pero asociado a la exclusividad, es decir "poseemos algo", bien sea total o parcialmente; de modo que tenemos plena capacidad y absoluto dominio del bien poseído. Propiedad posesiva

Si adquirimos un refresco, podemos consumirlo, compartirlo o regalarlo, pero debemos fijar nuestra atención en unos rasgos fundamentales que acompañan a esta idea de propiedad.

En primer lugar, el producto adquirido no ve alteradas sus características esenciales dependiendo del precio pagado, incluso si nos lo han regalado tenemos pleno dominio sobre nuestra posesión y nada condiciona nuestra decisión acerca del destino posterior del refresco. Cuando compramos software propietario este dominio no es absoluto y se nos cercena la posibilidad de adaptar, mejorar, copiar y/o compartir el producto.

En segundo lugar, si ejercemos el derecho de decisión sobre el refresco, en cualquier sentido, perdemos todo o parte de nuestro dominio, tanto si lo consumimos, lo compartimos o lo regalamos.

La idea de posesión lleva implícita, pues, la idea de renuncia. No ocurre así con el software. Si poseemos un programa informático, podemos cederlo y/o compartirlo sin que perdamos ni un ápice de nuestra primitiva capacidad de dominio sobre la posesión. Esta característica, por sí misma, justificaría al menos una profunda reflexión sobre el peligro de extender la noción de propiedad -y por tanto el derecho sobre ella- alegremente a este terreno.

Una observación detallada y reflexiva de esta cuestión bajo el prisma educativo puede llevarnos, sin mucha dificultad, a la idea que instruir en este ambiente y con estos planteamientos puede fomentar el egoísmo innato en el ser humano, alentando el hecho de poseer en sí mismo, sin importar la legitimidad o el reconocimiento general y, sobre todo, sin margen para el intercambio tan necesario y enriquecedor en el ámbito del conocimiento.

La propiedad en el contexto de la comunidad del software libre tiene, y así debe tenerlo, un significado totalmente distinto al sentido de propiedad convencional. El propietario de un proyecto es aquél que tiene la capacidad exclusiva de redistribuir versiones modificadas del mismo. También ésto se origina dada la estructura organizativa y de trabajo que posee la comunidad hacker productora de software libre. Cuando surge un proyecto de programa o se aborda uno existente el grupo de hackers participantes se coordinan trabajando cooperativamente, en primer lugar, para evitar redundancias innecesarias como que varios escriban la misma parte del código y, en segundo lugar, para repartirse las tareas y sumar los talentos individuales. Cuando el resto de la comunidad acepta a ese grupo como responsable del proyecto, son los únicos autorizados para modificar y certificar las variantes de código introducidas, no por una cuestión reglamentaria, sino para evitar la muerte del proyecto por dispersión. Mientras tanto, cualquier usuario, hacker o no, es libre de usar o modificar la aplicación. Consideramos que es un proceder consecuente.

Esta forma de entender la propiedad consagra un principio de legitimidad, es decir, el propietario lo es no por una cuestión meramente legal, sino porque la comunidad reconoce en él unos rasgos que le acreditan y son moralmente aceptables para todos. Propiedad abierta

Por lo tanto, para la comunidad hacker de software libre no existe el concepto de propiedad convencional, de hecho toda su producción está amparada por la Open Source Definition (OSD) que, al menos en teoría, permite a cualquiera tomar el código fuente de una aplicación, duplicarlo y evolucionarlo en cualquier sentido.

No obstante, los usos y costumbres de los programadores de software libre hacen que su configuración y funcionamiento, tanto a nivel individual (Los proyectos también los puede mantener una sola persona) como a nivel de grupo "propietario" de un proyecto, les confieren unas características muy particulares.

Según Eric S. Raymond, en un ensayo sobre el comportamiento de estos hackers llamado "Cultivando la noosfera" pone de manifiesto la contradicción entre la teoría de la propiedad asumida por la comunidad del software libre y la práctica tan diferente que resulta de sus formas de proceder, ateniéndose a unas costumbres, a menudo, con matices heredados del mundo del "software propietario". Concretamente defiende que el modelo generativo de software libre es homólogo a la teoría de Locke sobre posesión de tierras, para ello destaca tres formas en las que un hacker puede adquirir la propiedad de un proyecto para crear un programa libre.

En primer lugar, fundando el proyecto. Lógicamente en este caso la "propiedad" no es discutida. En segundo lugar, heredándolo de su anterior propietario. Finalmente, proponerse como propietario de un cierto proyecto si éste necesita atención porque el propietario haya desaparecido o se encuentre abandonado. En este caso el protocolo requiere un anuncio previo a la comunidad, la espera prudencial antes de repetir su ofrecimiento y si no hay objeción a su propuesta, puede adueñarse del proyecto, aunque no será reconocido completamente mientras no aporte una innovación sustancial.

En cualquier caso, en todos los supuestos está presente una idea legitimadora que evita el sentimiento únicamente

egoísta y propicia el esfuerzo a la búsqueda del mérito. Sin duda, un valor formativo nada desdeñable.

Estructura de la propiedad libre

Esta forma de proceder, según Raymond, es totalmente lockeana y recuerda la ley común anglo-americana sobre tenencia de tierras, sistematizada y racionalizada por el filósofo inglés John Locke a principios de la era moderna.

Hay otras consideraciones acerca del proceder de la comunidad del software libre como Faré Rideau quien sostiene que el hacker no ejerce su dominio en el territorio de las ideas puras, sino en los proyectos de programación, incluyendo por tanto, una labor material que lleva asociada elementos como la confianza, reputación, etc. para él sería pues, más preciso, situar sus dominios en el terreno de la ergosfera o esfera del trabajo, si bien esta matización no perturba demasiado el concepto de propiedad mismo que se manifiesta en este ambiente.

Este modelo de propiedad es el que podríamos denominar, de un modo no exento de paradoja, "propiedad libre" porque en él no son fundamentales los conceptos de heredad o perpetuidad, sino que el control sobre lo poseído está, por la propia dinámica del sistema, siempre en cuestión. Un "propietario" que se negara a liberar nuevo código, que se mostrara reticente a la incorporación de mejoras, que su obstinación rebasara lo razonable o que la intransigencia pertinaz fuese una rémora para el progreso del proyecto, sería finalmente cuestionado por el resto de la comunidad y relegado de sus funciones, o bien, otro hacker asumiría el proyecto virando en la dirección adecuada y condenando, de paso, al ostracismo al líder dictatorial. Por tanto la propiedad se fundamenta en el respeto, en el prestigio, en la actitud y en la aptitud, siendo así fruto del consenso general. Cuestiones totalmente al margen del concepto de propiedad convencional.

Como ocurre en la mayoría de casos, para avanzar, es necesario mirar atrás. El concepto de propiedad surge como un acuerdo para evitar las luchas fratricidas por el control de bienes escasos. La sociedad se dota de este instrumento regulador como escudo protector ante los desastres ocasionados por la disputa de elementos vitales, no suficientes para colmar las necesidades o aspiraciones de todos los miembros de la comunidad.

En su génesis, la propiedad se ejercía sobre un, llamémosle así, "objeto" público, las tierras, el agua, los refugios donde cobijarse. Más tarde, fue extendiendo su ámbito para acoger las creaciones individuales o colectivas: las cosechas, las rudimentarias armas,... En definitiva, como todos los consensos humanos, es fruto de la necesidad, del "mal menor", del "beneficio"... del beneficio social. Toda sociedad acota sus libertades individuales con el único objetivo de protegerse o beneficiarse colectivamente.

Llegado este extremo, cabe preguntarse, ¿estas consideraciones son de aplicación al software?, ¿qué debe proteger la sociedad, al creador de software o al comercializador?, ¿es el software un bien escaso susceptible de ser protegido?, si un programa se copia, se divulga, ¿se impide o se limita que el resto de personas puedan crear nuevas aplicaciones?, ¿qué es más beneficioso socialmente, el software comercial o el software libre? Desde luego, sería paradójico que la sociedad adoptase un marco regulador que le perjudicase a sí misma. No tendría mucho sentido y, por supuesto, estaría en contra de la finalidad primitiva de las estrategias normativas, incluso de las más evolucionadas, como afirma Jean-Jacques Rousseau en "El contrato social": Encontrar una forma de asociación que defienda y proteja de toda fuerza común a la persona y a los bienes de cada asociado, y gracias a la cual cada uno, en unión de todos los demás, solamente se obedezca a sí mismo y quede tan libre como antes.

Es evidente que el software no presenta las características de un bien poseible en el sentido primigenio del término, más bien, nos debemos aproximar al concepto de "acción apropiativa" fruto del trabajo, tal y como convergen las tesis liberales y socialistas. Sin embargo hay muchas opiniones que sostienen la inapropiabilidad del software, abanderadas por Stallman y la FSF.

¿Software? libre, por favor

Pese a todo lo expuesto, hay determinados factores que intervienen, tal vez, para añadir mayor confusión y margen de controversia o, tal vez, para contextualizar el fenómeno del software y presentarlo junto a su genuino marco de gestación, desarrollo y distribución. Entre estos factores es indudable que destaca el fenómeno de la Internet, tanto su aparición,[4] como, y sobre todo, su expansión hasta el nivel que conocemos. Aunque todavía carezcamos, quizás, de perspectiva o lejanía histórica suficiente para calibrar en su justa medida.

Otro factor a tener en cuenta es la propia estructura de los programas, compuestos de algoritmos, subrutinas, bibliotecas,... no necesariamente genuinas, sino que es la combinación de una serie de elementos, en un cierto orden, lo que proporciona cohesión y funcionalidad a una aplicación informática, pero fundamentalmente, las tecnologías de distribución, edición y copia han evolucionado hasta unas cotas y con una velocidad, sin precedentes, ocasionando un fuerte desconcierto entre una sociedad organizada en unas estructuras jurídicas sorprendidas ante este voraz crecimiento.

Todo este cambio tecnológico se ha ido produciendo en las últimas décadas bajo el paraguas normativo de las leyes de propiedad intelectual, copyright y patentes, por supuesto generadas en una época diferente y con unos objetivos también diferentes. Tan sólo en algunos países se han producido cambios, modificaciones o nuevas normativas que recogían aspectos relacionados con el software, las nuevas tecnologías o el comercio electrónico, pero, bajo la referencia del actual contexto jurídico, es decir, sin afrontar un marco legislativo nuevo, acorde con las peculiaridades del tema y los intereses de la sociedad.

Es ciertamente interesante conocer la posición del movimiento por el software libre en relación con la regulación normativa existente y su evolución. En primer lugar, como hemos avanzado, se defiende la idea que el software no es patentable. Para ello se fundamentan en los principios básicos que generaron la adopción del concepto de propiedad en que se sustenta la teoría del liberalismo, incluso, la propia antropología: ante unos bienes materiales escasos, constituye la única salida civilizada al control de los recursos disponibles esquivando las luchas permanentes entre los miembros del colectivo, en cambio, el software no comparte esta característica, no forma parte de los bienes físicos o materiales sino que se encuadra en el ámbito de las ideas, constituyendo un tipo particular de éstas, más cercanas al conocimiento intelectual o, si se prefiere, a la aplicación práctica de éste.

En consecuencia, la escasez o no de ideas no está relacionada con que éstas sean propietarias o libres, puesto que es posible generar conocimiento simultáneamente, es decir, si dos personas tienen una idea, no impiden que la tenga también un tercero. La sociedad contra sí misma

Es el momento de plantearse, ¿por qué la sociedad ha llegado a ver con naturalidad el comercio de software?, ¿por qué se permiten las patentes sobre el conocimiento humano, e incluso se protegen jurídicamente?, ¿qué beneficio obtiene la sociedad con estas prácticas? Indudablemente, son muchos los factores que intervienen para explicar la situación actual.

En primer lugar, la propia naturaleza práctica del software [5] hace que vaya unido a la comercialización de servicios, a recursos físicos y a la realización de tareas que sí tienen mucho que ver con la economía de la sociedad. No es de extrañar, entonces, que, interesadamente, las empresas hayan tratado, y conseguido, llegar a confundir producto con servicio y comercializar todo un conjunto incluyendo el software en el lote.

En segundo lugar, se ha utilizado equívocamente la llamada propiedad intelectual y todo su marco legal relativo a derechos de copia y patentes, para colocar al software libre bajo su cobertura, cuando en realidad, ni por cuestiones históricas, prácticas, ni por interés social se justifica este hecho. Los derechos de copia surgen, y tienen su sentido, con la aparición de la imprenta que posibilitaba la producción en serie de libros, antes, ni existían, ni fueron necesarios, de hecho, era natural que los libros ser copiasen a mano sin despertar recelo social. Sin embargo, cuando la copia en serie se convirtió en realidad, la sociedad estimó la adopción de un instrumento que regulara esta nueva situación, naciendo así los derechos de copia. Es claro que la normativa restringía tan sólo a las imprentas, por lo tanto, a la sociedad en general [6] no se le cercenaba ningún derecho fundamental que le fuese propio y, en un principio, constituyó una normativa fácil de vigilar en su cumplimiento pues bastaba con controlar a las industrias de impresión. Sin embargo, a partir de la aparición de la fotocopiadora, pero sobre todo, con el tremendo desarrollo de la tecnología de autoedición y copia, este marco normativo está gravemente obsoleto pues para su control se están proponiendo y ejecutando una serie de medidas que colisionan frontalmente con derechos fundamentales de la libertad individual, síntoma claro de la no adecuación del ámbito jurídico a la necesidad social.

Otro tanto ocurre con la propiedad intelectual o "derechos de autor" que no constituyen un derecho fundamental, de hecho, en la Constitución de los Estados Unidos está claramente recogido que la protección al autor sobre sus derechos intelectuales debe ser por un tiempo limitado, en su origen diez años, pero con el único fin de favorecer el progreso, es decir, la sociedad protege al autor pero con el objetivo "egoísta" de beneficiarse a sí misma. No ocurre así en la situación actual, en que los derechos de autor se esgrimen por las grandes corporaciones, para justificar sus propios intereses, no los de los autores ni los de la sociedad en su conjunto.

En la época en la que surge la legislación sobre los derechos de copia, ningún ciudadano miembro de la sociedad renunciaba, de hecho, a ningún tipo de libertad "ejercible" pues no disponía de industria de reproducción, hoy día, esta renuncia sí es significativa. El desarrollo tecnológico ha hecho posible que una gran mayoría de ciudadanos tengan la posibilidad de acceder y copiar información de cualquier tipo, por lo tanto, parece necesario, un replanteamiento de la situación, a nivel político y jurídico. Richard Stallman, en una conferencia pronunciada en la Universidad de Burdeos, el 7 de julio de 2000, pone un ejemplo bastante ilustrativo. Viene a decir que, si en las circunstancias actuales, a un ciudadano le pagan por firmar un documento en el que renuncia a realizar viajes interestelares, sería un trato ventajoso para él ya que, de momento, no limitaría nada su libertad de actuación, sin embargo, puede que dentro de unos años, el mismo trato tenga unas connotaciones distintas y, entonces, sí suponga una merma sensible para sus derechos y libertades básicos, debiendo ser abordado desde otra posición.

Por otra parte, desde los tiempos de la FSF el software libre ha tenido que justificarse continuamente y tratar de argumentar sus ventajas sociales, tanto desde el punto de vista ético, filosófico, moral, económico o de capacitación

técnica. Es necesario también que nos detengamos a reflexionar sobre las mejoras que representan para la colectividad los programas comerciales y demandemos a las empresas de software propietario las explicaciones pertinentes.

Hasta ahora, todas las manifestaciones e informes en este sentido han carecido de argumentación sólida y de rigor científico. Se han basado en verdades a medias e hipótesis usadas como argumentos irrefutables para establecer unas conclusiones totalmente interesadas y alejadas de toda objetividad.

Sirva como ejemplo el informe elaborado por la empresa norteamericana NERA (National Economic Research Associates), conocido como Informe NERA, analizado por Ricardo Galli Granada, profesor de la Universidad de las Islas Baleares, en un artículo aparecido en *Bulma*, donde se pone de manifiesto la parcialidad y ausencia de rigor de este tipo de documentos elaborados al dictado de los grandes intereses comerciales.

El software libre como herramienta educativa

Pocos ponen en duda que, hoy día, el software tiene un protagonismo esencial en todos los ámbitos de la sociedad, especialmente en el educativo. Constituye un instrumento básico para todas las tareas relacionadas con la generación, transmisión, recuperación y almacenamiento de información. Este fenómeno que muchos califican de Tercera Revolución Industrial y que hemos dado en denominar como Tecnologías de la Información y la Comunicación (TIC), descansa, en buena medida sobre los dominios del software.

El manejo de aplicaciones, ya básicas, como el correo electrónico o el procesador de textos son un requisito para la mayoría de la población contemporánea. Nuestros escolares no pueden abandonar el colegio sin conocer, más aún, sin dominar estas destrezas.

Ahora bien, ¿qué se necesita para conseguir este objetivo? En general, muchas y muy variadas condiciones. Algunas pueden resultar polémicas y controvertidas, sin embargo otras parecen diáfanas y concitan bastantes acuerdos, sobre todo desde el sentido común. Cada persona, cada educador puede tener, y probablemente tendrá, una visión muy particular de esta cuestión. Puede discrepar en la edad de iniciación apropiada, puede cuestionar los métodos y los procesos de enseñanza-aprendizaje, puede acentuar en mayor o menor grado el aspecto tecnológico-científico del software o el aspecto instrumental del mismo, sin embargo, resulta evidente que algunas cuestiones están al margen de la discusión.

En primer lugar, para poder adiestrar en las TIC y convertirlas en un vehículo eficaz de transmisión de conocimientos junto a una herramienta formativa y útil, es necesario acercarlas a los centros educativos, de modo que el alumno crezca y se desarrolle junto a ellas, con ellas y, por qué no, por ellas. El esfuerzo necesario para este acercamiento que debe acometer la entidad competente no es baladí. El software libre se perfila como un aliado idóneo para esta tarea, aunque sólo seamos capaces de apreciar, obtusamente, el ahorro económico en el coste de licencias.

Por otra parte la educación debe tener un carácter igualitario y ser un factor esencial en la corrección de desequilibrios sociales. No parece, pues adecuado, formar en unos sistemas propietarios y comerciales alejados, en muchos casos, de las posibilidades económicas de una gran parte de la población, sino que debemos procurar la formación en entornos abiertos que garanticen el principio de igualdad de oportunidades.

Básicamente y de una forma general podemos considerar como necesidades sociales primarias en relación con las TIC, la posibilidad de generar información en diferentes soportes y formatos, la capacidad de transmitirla y compartirlas con nuestros semejantes, la recuperación, consulta y almacenamiento de la misma, en resumen, herramientas de gestión total del conocimiento disponible y/o generado. Puede quedar un resquicio para la polémica si cuestionamos las capacidades reales del software libre para estas tareas, sin embargo, no resulta demasiado convincente esta objeción en los tiempos actuales, donde numerosas aplicaciones libres han demostrado con creces, no ya su idoneidad para el objetivo de su diseño sino, en muchas ocasiones, un nivel de prestaciones muy por encima de otras soluciones comerciales del mismo propósito. El carácter educativo del software libre

Si dejamos a un lado las obvias capacidades del software para ser usado como una herramienta eficaz en los entornos educativos y nos centramos en sus valores intrínsecos o, incluso en su estructura generativa, aproximándonos a sus formas de organización y de relación, podremos observar una serie de rasgos con un tremendo valor pedagógico y formativo.

La educación, hoy día, no se concibe únicamente como un mero sistema transmisor de conocimientos preocupado por la alta cualificación y especialización de los individuos sino que también procura no desdeñar los aspectos formativos que rodean todo el proceso de enseñanza-aprendizaje acentuando los valores, actitudes y principios como ejes transversales de este proceso con la finalidad de conseguir individuos plenamente integrados, responsables, solidarios, en definitiva, una sociedad más rica y humana.

Con carácter general, los entornos de generación de software libre aportan tipologías muy aprovechables en este campo.

-

El esfuerzo cooperativo, el trabajo en equipo, donde cada miembro de un grupo siente la responsabilidad y, a la vez, la consideración de la importancia de su tarea para la consecución de un objetivo común es, en muchas ocasiones, un acicate y un elemento motivador amén de un magnífico instrumento de mejora de la autoestima.

-

El planteamiento de metas comunes y la labor personal necesaria para llevarlas a cabo fomentan la generosidad y mantienen a raya el, a veces, excesivo individualismo propio del ser humano.

-

El respeto surgido del consenso y no de la imposición supone un magnífico aliado para potenciar la cultura del esfuerzo y el afán de superación personal como medio para lograr el reconocimiento legítimo.

-

La revisión y transparencia del trabajo personal invitan claramente a la adquisición de las dosis necesarias de humildad propicias para el progreso y la formación individual, lejos de posturas intransigentes y sectarias propias de otros ambientes más oscurantistas.

-

La toma de decisiones basadas en la transparencia y en datos objetivos constatables constituyen un fuerte inhibidor de las tentaciones autoritarias y fomentan unos valores democráticos genuinos sustentados en la información y en la responsabilidad.

-

La igualdad y, a la vez, la diversidad está naturalmente aceptada, potenciada, respetada y convenientemente valorada. Nadie debe sentirse inferior ni superior a nadie. Cada persona tiene sus propias destrezas que se le respetan y reconocen de modo que se canalizan adecuadamente para que el proyecto, en su conjunto, resulte beneficiado.

-

El conocimiento circula libre. Es cuestionado, adquirido, mejorado y difundido en un esfuerzo conjunto imprescindible, para el progreso de la sociedad.

Todos estos rasgos se encuentran presentes en el movimiento del software libre. Son inherentes a él y constituyen unos pilares básicos que pueden ayudarnos a entender su vertiginoso desarrollo. Tal vez ha llegado el momento de preguntarnos si la sociedad actual puede permitirse desdeñarlos.