

Universidad de las Ciencias Informáticas

Facultad 10



Título: Contribución a la Epistemología del Software Libre

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Diplomante

Yanicet Aveleira Rodríguez

Tutor

MSc. Tomás López Jiménez

**Ciudad de La Habana
Julio 4 del 2008**

“Nuestro pasado ha sido una cultura libre, pero solamente lo será en nuestro futuro si cambiamos el rumbo en el que vamos”

Lawrence Lessig

Declaración de Autoría

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2008.

Yanicet Aveleira Rodríguez
Autora

MSc. Tomás López Jiménez
Tutor

Datos de Contacto

Tomás López Jiménez. Guantánamo, Cuba. 1943. Graduado de Ingeniero Electricista en la Universidad de la Habana en 1974. MSc en Informática Aplicada en el ISPJAE en el 2007. Investigador Titular desde el año 1985. Profesor Auxiliar en el 2007. Asesor del Rector desde el 2005 y Director de Estrategia y Calidad de la Universidad de las Ciencias Informáticas desde el 2008, con una larga trayectoria laboral, de ella 42 años en la Informática. Su actividad docente la desarrolla en la UCI, impartiendo las asignaturas de Historia de la Informática y Metodología de la Investigación Científica en la Facultad 10 en pregrado, así como profesor de postgrado y colaborador de los Departamentos Docentes Centrales de Práctica Profesional y Sistemas Digitales. Ha cursado numerosos postgrados de la especialidad y de Técnicas de Dirección, incluyendo un diplomado en Dirección y Gestión Empresarial. Es integrante de la Comisión de Carrera, del Consejo Científico y de otros órganos colegiados de la Universidad. Ha sido miembro de numerosas organizaciones profesionales y científicas, siendo actualmente integrante de la Sección 9.7 sobre Historia de la Informática de la IFIP.

E-mail: tlopezj@uci.cu

A mis padres, por todo el apoyo que siempre me dan, por la dedicación y los consejos con los que me han hecho cada día más fuerte, a mi familia, a mi tutor que ha sido para mi como un padre en la UCI, a Adonys por todo su cariño, a mi compañera de tesis Dayaisis por toda la ayuda que me ha brindado, a la profe Marlen por su apoyo, a mis colegas de estos cinco años. En fin a todos los que han hecho posible de alguna forma que haya llegado hasta aquí.

A mis padres...

Resumen:

El informe de la investigación “Contribución a la Epistemología del Software Libre” como Trabajo de Diploma en opción al título de Ingeniero en Ciencias Informáticas ofrece elementos básicos sobre el origen y evolución del software como producto de la creación intelectual y de su industria derivada. Examina la interacción hardware-software en sus efectos recíprocos y sus diferencias esenciales como industrias, identificando el rol y continuidad de los sistemas operativos. Presenta los procesos de desarrollo, mejora e intercambio colectivo del software por las comunidades actoras, sus motivaciones y factores de cambio, mostrando evidencias de ello. Expone cómo ha evolucionado el concepto “libre” que le es inherente al software a lo largo de su historia, con énfasis en las diferencias, razones del cambio y necesidad de su perfeccionamiento, enfatizando sus matices sociales e ideológicos crecientes; de forma entrelazada analiza el origen y evolución del Software Libre desde la década de 1980, junto con la diversificación de ese concepto, con énfasis en sus diferencias y motivaciones principales. Examina sistematizadamente la base jurídica correspondiente a ambos tipos de productos, sus relaciones y restricciones, proponiendo enfoques sobre maneras de actuación y los contenidos comunes a conocer y utilizar por los actores de ese proceso desde la etapa de proyecto. Finalmente ofrece los resultados del diagnóstico realizado al nivel de conocimiento sobre el software en general y el libre en particular por los colectivos de la universidad, mediante lo que se corrobora la situación problemática y los objetivos de la investigación, ofreciendo las conclusiones y recomendaciones para alcanzar mejores desempeños en las actividades de formación y producción.

ÍNDICE GENERAL

Introducción: 1

Origen, Naturaleza y Evolución del Software 13

1.1 Introducción..... 13

1.2 Un poco de historia 13

 1.2.1 El inicio..... 14

 1.2.2 Nacimiento del BASIC 17

 1.2.3 Papel de Bill Gates 19

 1.2.4 Inicios de UNIX..... 21

 1.2.5 Stallman y el nacimiento de GNU y FSF 27

 1.2.6 Historia del MINIX y algunos de sus roles 29

 1.2.7 El Kernel Linux 30

 1.2.8 Aparición de Internet 32

1.3 Polémica entre software libre, código abierto y software gratis 33

 1.3.1 Software Libre 35

 1.3.2 Software gratis 36

 1.3.3 Software de Código Abierto o de Fuente Abierta (Open Source)..... 37

 1.3.4 Diferencias entre Software Libre y Fuente Abierta o Código Abierto 40

1.4 Algunos conceptos 43

 1.4.1 Hardware 43

 1.4.2 Software 43

 1.4.3 Sistema Operativo 43

 1.4.4 Software propietario, privativo o cerrado 44

 1.4.5 Dominio público..... 44

Aspectos Legales del Software con Énfasis en el Libre 46

2.1 Introducción..... 46

2.2 Aspectos legales 46

 2.2.1 Derechos de autor..... 47

2.2.2 Patente.....	50
2.2.3 Marca.....	50
2.2.4 Secreto comercial.....	50
2.2 Licencias de software	51
2.4.1 Licencias libres.....	52
2.4.1.1 Licencia Pública General de GNU (GNU-GPL)	54
2.4.1.2 La Licencia Pública General Menor de GNU (GNU LGPL).....	55
2.4.1.3 Licencia BSD	55
2.4.1.4 Licencia Pública de Mozilla (MPL)	56
2.4.1.5 Licencia X11	57
2.4.1.6 Licencia de la Fundación Apache	57
2.4.1.7 Licencias de documentación libre o FDL.....	57
2.4.2 Licencias propietarias.....	58
2.4.3 Compatibilidad de licencias	63
2.2 La tecnología de la información puede sustentar una revolución social	66
2.2.1 Herramientas colaborativas	67
2.1.1.1 Wiki.....	67
2.1.1.2 Blog	68
2.4.2 Creative Commons.....	70
Diagnóstico del Nivel de Conocimiento sobre SWL en la UCI	73
3.1 Introducción.....	73
3.2 Sistema de referencia para la evaluación e interpretación de los resultados	73
3.3 Resultados y análisis del diagnóstico	74
3.3.1 Categoría del Nivel de Conocimiento	74
3.3.1.1 Subcategoría del Nivel de Conocimiento Preciso	76
3.3.1.2 Subcategoría del Nivel de Conocimiento Impreciso	77
3.3.2 Categoría del Nivel de Desconocimiento	79

Conclusiones	82
Recomendaciones	84
Bibliografía.....	85
Bibliografía Consultada	85
Bibliografía Referenciada	89
Glosario de términos.....	91
Apéndices del documento.....	93
Apéndice A - Cronología	93
Apéndice B - Elementos del escenario internacional de la industria de software.	97
Anexos del documento	100
Anexo 1	100
Anexo 2.....	102
Anexo 3.....	108

ÍNDICE DE TABLAS Y FIGURAS

Figura 1 Representación gráfica del Nivel de Conocimiento.....	76
Figura 2 Representación gráfica del nivel de Conocimiento Preciso	77
Figura 3 Representación gráfica del nivel de Conocimiento Impreciso	78
Figura 4 Representación gráfica del Nivel de Desconocimiento	80
Figura 5 Representación gráfica del Nivel de Desconocimiento Consciente.....	80
Figura 6 Representación gráfica del Nivel de Desconocimiento Inconsciente	81
Tabla 1 Libertades del SWL	35
Tabla 2 Nivel del Conocimiento por Grupos - en % por pregunta	75
Tabla 3 Nivel del Conocimiento Preciso por Grupos – en % por pregunta.....	77
Tabla 4 Nivel del Conocimiento Impreciso por Grupos – en % por pregunta	78
Tabla 5 Nivel de Desconocimiento por Grupos – en % por pregunta.....	79
Tabla 6 Nivel de Desconocimiento Consciente por Grupos – en % por pregunta	80
Tabla 7 Nivel de Desconocimiento Inconsciente por grupos en % por pregunta.....	81

Introducción:

El concepto de Software Libre (SWL) se utiliza muy generalizadamente sin que aún se pueda hacer referencia a un sistema de conocimientos depurados y compartidos por una comunidad académica y profesional que le ofrezca el necesario respaldo. Además de la gran cantidad de conceptos y hechos correctamente establecidos sobre este, frecuentemente se leen, escuchan o se actúa sobre la base de preconcepciones, mitos y criterios de personas, grupos y escuelas que divergen en muchos de sus aspectos medulares. Al no disponerse de fuentes reconocidas y sistematizadas con el rigor necesario, resulta muy difícil y de no poca incertidumbre el actuar con certeza y seguridad en este campo.

Muchas veces ocurre que las organizaciones y personas que desarrollan SWL o que piensan que así lo hacen, no tengan claros aspectos fundamentales como los relativos a sus comunidades y formas de desarrollo y producción, integración, mantenimiento y otros soportes, distribución comercial o colaborativa, formas de propiedad, mecanismos de protección legal, variedad de licencias y sus incompatibilidades, diferencias entre el derecho de autor, el derecho de copia o reproducción (copyright) y otras formas de propiedad industrial o intelectual, en dependencia de las normas internacionales, así como de las culturas y bases jurídicas regionales y nacionales.

La industria del software con apenas cincuenta años de historia, a principios del presente siglo se ubicó como la cuarta rama de mayor importancia económica en EE.UU, ocupando también lugares importantes en otros países. Este producto que es un típico resultado del proceso de creación intelectual, junto con la infraestructura tangible que lo soporta (las redes como expresión distribuida, ubicua y omnipresente del hardware) se ha convertido también en el proceso más universal, acelerador y eficiente de la creación y desarrollo del conocimiento, deviniendo esto quizás en su rasgo y potencial más importante.

Sus implicaciones horizontales y universales para el desarrollo humano y en especial del conocimiento como rasgo distintivo de la especie, lo sitúan también al centro de los procesos de la creación y desarrollo cultural y del disfrute intelectual y espiritual. Estas singularidades lo llevaron desde sus orígenes a situarse gradualmente al centro de la lucha por las fuentes

abiertas (open source) de la cultura, incluso antes de que apareciera el conocido movimiento explícito del software libre, el que a su vez proclama entre sus libertades la garantía del acceso al código fuente.

Algunos de los principios fundamentales del software libre proceden de las filosofías de apertura, honestidad y colaboración, establecidos desde mucho antes en los campos de la investigación académica y científica, en este caso concreto en el de la ciencia de la computación.

Comunidades de software, que ahora pueden compararse con las actuales comunidades del software libre, antecedieron a este movimiento y a su término “software libre” (“free software”). De acuerdo a Richard Stallman, la comunidad de software compartido en MIT existió por “*muchos años*” antes de que él se involucrara en ella en 1971.

Prácticamente desde sus orígenes el software se producía por investigadores académicos y corporativos trabajando en colaboración. Inicialmente este no se veía por sí mismo como un producto o mercancía básica o primaria independiente del hardware. Simplemente constituía un acompañamiento o valor añadido incorporado inicialmente por los fabricantes del hardware o distribuido libremente entre las comunidades o asociaciones de “usuarios” de su línea de hardware.

Sistemas Operativos tales como las primeras versiones del UNIX, fueron ampliamente distribuidos y mantenidos por las comunidades de usuarios. El código fuente, la versión del software legible por los seres humanos, se distribuía con éste porque los usuarios frecuentemente lo modificaban por sí mismos para solucionar errores o añadir nuevas funcionalidades y porque para los programadores no resultaba posible crear código de máquina ejecutable para la amplia variedad de hardware que existía. Por lo tanto, en esa era el software fue libre en ese sentido, no porque existiese algún tipo de esfuerzo concertado entre sus usuarios y sus creadores, sino porque este se desarrollaba por la comunidad de usuarios.

AT&T distribuyó gratuitamente o a muy bajo precio versiones tempranas del UNIX a

investigadores gubernamentales y académicos, pero sin concederles el necesario permiso para su redistribución o para distribuir las versiones modificadas por ellos, por lo que no constituían SWL en la acepción moderna de esta frase o concepto.

A finales de los años de 1970 e inicios de los 80, las empresas comenzaron rutinariamente a imponer restricciones a los programadores mediante el derecho de copia o de reproducción (Copyright). En ocasiones ello se debió a que esas libertades les cortaban una vía para hacer dinero, por lo que bloquearon esos derechos y procedieron a vender esos productos. Bill Gates marcó el cambio de los tiempos en 1976 cuando escribió su famosa Carta Abierta a los Aficionados (Open Letter to Hobbyists), en la que señalaba que lo llamado “*compartido*” por los hackers era, en sus propias palabras, “*robado*”. En 1979 la AT&T comenzó a reforzar sus licencias restrictivas cuando llegó a la conclusión de que podía obtener beneficios monetarios vendiendo el sistema UNIX.

Adicionalmente el advenimiento de Usenet a principios de los años de 1980 conectó a la comunidad de programadores, proporcionándoles así una vía simple para que compartieran su software y contribuyeran al desarrollo y mejora del escrito por otros.

¿Qué permanece de aquellos tiempos?

Algunos productos libres de software que fueron desarrollados en la década de 1970 y primeros años de 1980, entre los que continúan utilizándose por largo tiempo se encuentran SPICE, TeX (desarrollado por Donald Knuth), y el Sistema X Window. Los desarrollos originales del Sistema X Window fueron realizados dentro del proyecto W Window System, llevados a cabo en su mayoría después del lanzamiento del proyecto GNU, sin que este último fuera en manera alguna su responsable.

El Software (SW) en sus orígenes nació bajo ambientes libres, no se protegía como producto comercializable y se desarrollaba por comunidades o grupos colaborativos de una manera u otra. En su evolución, precisamente las corrientes más fuertes de convertir al SW en un producto propietario, muy destacado el caso de Microsoft, en no pocos casos partieron de apropiarse de alguna forma y proteger con el copyright (CR) a productos de SW que

nacieron libres o pertenecientes a determinadas comunidades. El BASIC constituye uno de los primeros ejemplos de ello; cuando este se concluyó y se dio a conocer en 1964, sus autores lo declararon expresamente como un producto del “dominio público”, incluido su código fuente. Unos 12 ó 14 años después es precisamente ese resultado el que se convierte en el primer producto de la naciente Microsoft, constituyéndose así en la base para el desarrollo del cada vez más férreo sistema de licencias de SW propietario, mediante lo que casi es imposible distribuir y reutilizar nuevos productos y aplicaciones sin una dependencia extrema de las organizaciones poseedoras del CR.

La propia evolución del hardware y del software y su interacción recíproca en su desarrollo, constituyen de por sí una de las componentes más estratégicas del avance de la Informática Moderna. Lenguajes de programación, sistemas operativos, herramientas, ambientes y plataformas en su devenir histórico han ido enfatizando las contradicciones y vaivenes de libre a propietario a libre.

El movimiento GNU a partir de 1984 con su Manifiesto originó un escenario que explícitamente intentaba catalizar las contradicciones anteriores, dándole mayor claridad de objetivos, organización y coherencia al movimiento colaborativo entre las comunidades desarrolladoras y productoras del SWL. En paralelo éstas contradicciones e intereses necesariamente han devenido mucho más complejos, incrementándose y diversificándose las modalidades, principios, conceptos y objetivos, haciendo muy variado y difícil su comprensión, organización, relaciones y gerencia, junto con el incremento de las preconcepciones y confusiones sobre esta forma de desarrollar, producir y distribuir el ya insustituible SW en la vida de las personas de la sociedad moderna.

Junto a las implicaciones científicas y tecnológicas del desarrollo interactivo del software y el hardware, de la propia Ingeniería del Software y de otras componentes de la Informática Moderna como cosa en si, su convergencia e integración con las comunicaciones dando lugar a la Tecnología de la Información, gestaron y dieron vida a la Sociedad de la Información y por su vía, a la del Conocimiento. Ese desarrollo constituye la Revolución Científica que pudiera ser portador del cambio del modo social capitalista de producción de

conocimientos a un modo colaborativo de ese proceso, como una de las formas más acabadas de la creación de valor.

Para ello ha sido decisivo el desarrollo de las redes, propiciador de la aparición y consolidación de Internet, en especial a partir de las WWW, induciendo el desarrollo de las herramientas colaborativas como una de las características sociales del desarrollo de la Informática. Entre otras referencias, vale la pena citar el libro del profesor Yochai Benkler de Harvard, “La riqueza de las Redes”, en cierto análisis paralelo y analogía comparativa con el clásico de Adam Smith “La riqueza de las Naciones” como base teórica inicial del modo de producción capitalista.

La fuente de contradicciones y confusiones en la interpretación de ese desarrollo singular de la ciencia y la tecnología y su generalización en la sociedad humana han ido aumentando con el paso de los años. En esas condiciones el término de SWL constituye un importante problema que conceptualmente no está tolo lo claro que se requiere. Es estratégico para la UCI y para el país que esa evolución se investigue y esclarezca con el mayor rigor posible, de manera que se ofrezcan elementos básicos de conocimientos sobre este campo, como contribución a una fuente de aprendizaje y guía para la acción de los colectivos profesionales y estudiantiles.

Declarándose como **Pregunta de la investigación** la siguiente:

¿Cómo contribuir a esclarecer el concepto de Software Libre estudiando metódica y rigurosamente el desarrollo del software, identificando y seleccionando el saber verdadero resultante de su evolución, para ofrecer una mejor aproximación al criterio de su verdad, ofreciendo una contribución al necesario saber organizado y fundamentado, despojado en todo lo posible de preconcepciones y otras deformaciones e inconsistencias?

Esta investigación tiene como **Objetivo General** ofrecer un modesto sistema de conocimientos organizados y fundamentados sobre el SWL, que pueda tomarse como una fuente introductoria más fiable y segura, en apoyo al proceso de enseñanza-aprendizaje en la UCI, ofreciendo elementos básicos sobre su origen y evolución, así como otras

generalidades indispensables para concebir, organizar y desarrollar proyectos en este controvertido y estratégico campo de actividad.

Como **Objetivos Específicos** se establecieron los siguientes:

1. Identificar las características esenciales sobre el origen, desarrollo y evolución del hardware y del software y su interacción recíproca, con énfasis en sus modalidades de producción y desarrollo vertiginoso y cambiante de estas industrias y su influencia en la Informática Moderna.
2. Ofrecer con la profundidad y el mayor rigor posibles los elementos principales del desarrollo y evolución del SWL como un modesto aporte en atención a su importancia estratégica para la universidad y, por consiguiente, para el país y la sociedad.
3. Esclarecer y extraer de la abundante y controvertida información internacional sobre el SWL, aquellos hechos y conceptos que ofrezcan una referencia clara y sistematizada como fuente de aprendizaje y guía para la acción de los colectivos profesionales y docentes.
4. Realizar un diagnóstico del grado de conocimientos básicos y generales sobre SWL en la UCI con el propósito de identificar y proponer las acciones esenciales que contribuyan a que este se eleve a los niveles necesarios en los diferentes colectivos.

Se define como **Objeto de Estudio** del presente trabajo la Epistemología del Software.

De aquí que el **Campo de Acción** queda enmarcado en la Epistemología del Software Libre.

Preguntas Científicas:

1. ¿Cuáles son los elementos principales que han caracterizado al SW y HW en su desarrollo desde su origen?

2. ¿Cuál ha sido la influencia del HW y SW en el desarrollo de la Informática Moderna?
3. ¿Cuál sería un concepto recomendable de SWL, para el aprendizaje de los colectivos afines con el tema?
4. ¿Qué escenario básico pudiera ofrecerse sobre el alcance legal que rodea al entorno del SW y su desarrollo?
5. ¿Cuál es el nivel de conocimiento básico y general sobre SWL existente en la actualidad en los colectivos de la UCI?

Tareas de la investigación:

1. Estudio de la evolución del hardware y del software.
2. Análisis detallado de los conceptos que han marcado la evolución del software.
3. Valoración de los sistemas legales del mundo del SW.
4. Fundamentación de elementos que apuntan a la certeza tecnológica de que la Información es un modelo científico que sustenta a la nueva sociedad.
5. Diseño y aplicación del instrumento de medición para diagnosticar el estado del conocimiento sobre el SWL en la UCI. Evaluación y valoración de los resultados de la muestra. Preparación del resumen y las recomendaciones.

Métodos teóricos utilizados en esta investigación

Analítico – sintético

Mediante este método se identificaron y diferenciaron los conceptos básicos sobre el SW, las modalidades y variantes de su desarrollo y producción, su evolución como producto y servicios, contribuyendo ello también a identificar los elementos relacionados con los mitos y preconcepciones que giran alrededor de la categoría SW y de su derivada identificada mediante el término SWL. Ello permitió un análisis ordenado de los conceptos que pudieron

identificarse dentro del alcance de un proyecto de investigación para un diploma de pregrado. Posteriormente, con esta información, se procedió a diferenciarlos y sintetizarlos, además de extraer los elementos que se consideraron más importantes, teniendo en cuenta el objeto de estudio. Estas operaciones no son independientes, toda vez que el análisis de un objeto se realiza a partir de la relación que existe entre los elementos que lo conforman y a su vez, la síntesis se produce sobre la base de los resultados previos del análisis.

Histórico – lógico

A partir de elementos fidedignos de la historia del SW, se estudiaron los fenómenos de su trayectoria, lo que permitió establecer la sucesión cronológica para el estudio investigación de sus antecedentes, evolución y tendencias, esclareciéndose el proceso de aparición y diversificación de lo que hoy se conoce de forma general como Software Libre (SWL), sus momentos iniciales, evolutivos y cumbres, así como los desafíos o retos en su devenir y en el presente, lo que pudiera servir para proponer elementos de visión sobre sus perspectivas. Con ello fue factible estudiar particularidades básicas de su esencia, siguiendo una línea lógica en la investigación sobre su desarrollo y funcionamiento.

Sistémico

El análisis sistémico del SW como objeto de investigación posibilitó modelarlo como fenómeno singular de la producción intelectual. Se adoptó el enfoque de verlo como una superposición de abstracciones que extiende “la máquina real – la computadora” mediante “máquinas virtuales – los programas como capas superpuestas e interrelacionadas” en su capacidad de máquina de aplicación universal. De esta forma se comprende que la “producción” de esas capas involucra, estimula y requiere de la participación de personas especializadas en diversas etapas del proceso desde la capa del HW hasta la aplicación final.

Este modelo del SW facilitó identificar en su proceso de desarrollo y distribución las componentes que dieron y continúan dando lugar a la complejidad de relaciones existentes entre los grupos de personas, sus intereses, necesidades y motivaciones individuales y

sociales, de manera que pudiera apreciarse su estructura y la dinámica seguida por esta hasta el presente.

Métodos Empíricos

Como el objeto de estudio de esta investigación en su origen y evolución depende esencialmente de fenómenos subjetivos y del movimiento e intereses sociales de sus actores principales, en interacción que gradualmente se ha ido extendiendo a toda la sociedad, la información estadística sobre su comportamiento en el tiempo ha sido clave. Esta se extrajo de fuentes históricas cuidadosamente seleccionadas desde el punto de vista de su certeza y fiabilidad.

Por otra parte, en atención al objetivo específico de realizar un diagnóstico del grado de conocimientos básicos y generales sobre el SWL en la UCI, cuyos resultados sirvan de base a un programa de acciones que lo eleve a planos superiores, fue indispensable concebir y diseñar un instrumento que permitiera medirlo y evaluarlo, diseñando la estructura de individuos e indicadores más apropiados, representativos y consecuentes con un diagnóstico inicial en una población, que aunque heterogénea, debe constituir un espacio donde este conocimiento esté muy por encima de la media nacional. Esta investigación se llevó a cabo mediante una encuesta en cuyo diseño se consideraron cuidadosamente los requerimientos antes descritos.

Encuesta

Se elaboró y aplicó un cuestionario que permitió diagnosticar el grado de conocimientos sobre elementos básicos del SW como fenómeno genérico y sobre el SWL como una de sus expresiones particulares, sin entrar en aspectos demasiado profundos y especializados respecto al estado del arte.

Se tomaron a los colectivos de la universidad como la población a diagnosticar. Se llegó a la conclusión que el establecimiento de una muestra intencionada ofrecería una medición

mucho más adecuada que otra técnica, optándose así por el muestreo no probabilístico, estableciendo cuotas intencionadamente representativas.

Se trabajó con una muestra de 134 personas, distribuidas en tres grupos que se diferenciaron sobre la base del nivel de conocimientos que debieran tener acerca de la industria del SW en general, y en particular sobre el SWL, en dependencia del área de trabajo, el cargo, las funciones y tareas a desempeñar y los objetivos a lograr por cada uno. De esta manera se definieron los tres grupos siguientes:

Grupo 1: Integrado por individuos que un bajo nivel de conocimientos sobre el SW en general y sobre el SWL en particular sería suficiente para el buen desempeño de sus roles.

Grupo 2: Se integró con personas que para el logro satisfactorio de sus objetivos al menos debieran tener un nivel medio de conocimiento en el referido campo.

Grupo 3: Formado por individuos que por su rol y objetivos debieran tener un alto nivel de conocimientos sobre estas temáticas.

Como criterio para asegurar por defecto la maximización de la representatividad estadística de cada grupo, la distribución de las personas en estos se hizo cuidadosamente en función de la intención preestablecida. Para el Grupo 1 se seleccionó personal de servicios de las diferentes áreas de la universidad, dándole preferencia a aquellos que probablemente tuvieran el mejor conocimiento del tema. En los grupos 2 y 3 se cuidó de incluir estudiantes y profesores de 4º y 5º años, especialistas y cuadros de las áreas docentes, producción, IP e investigación. Como parte de la IP se incluyeron también los asesores jurídicos y representantes de ALBET, integrándolos en el Grupo 3; en este se incluyeron estudiantes de 5º año (por ejemplo del colectivo de NOVA), profesores y cuadros de la F10, así como otros cuadros del primer nivel de dirección de la universidad. Los estudiantes y profesores incorporados al Grupo 2 no podían ser de la F10.

La estructura lógica de la encuesta o cuestionario se preparó siguiendo el diseño y criterios de investigación antes descritos. De esta manera, sus 14 preguntas corresponden a dos

grupos. El primero con cuatro preguntas, se refiere a aspectos muy generales de la industria del SW respecto a su origen como producto comercial, primeras formas de desarrollo y producción, los fenómenos que pudieron incentivar a la generalización del SW propietario y formas generales de su protección legal como producto. El segundo, presentó diez preguntas centradas en cuestiones generales sobre el SWL, entre ellas aspectos como si es gratis o no, si se debe proteger, relación elemental entre SWL y Open Source, generalidades sobre las licencias, derechos generales y algunos criterios básicos que deben tenerse en cuenta al desarrollar un proyecto de SWL.

En ambos grupos de preguntas la complejidad aumenta ascendentemente, sin rebasar las temáticas y conocimientos generales ya citados. Como método de respuesta predominó la selección de alternativas entre Sí, No ó No sé. En algunos casos la selección había que realizarla eligiendo entre más opciones, pero incluyendo siempre la de No sé. En el cuestionario se orienta que se parta de que no se evalúa a la persona que responde la encuesta, sino que se trata de que contribuyan a realizar un diagnóstico del estado de ese conocimiento en la universidad, por lo que se le recomienda fuertemente que ante la duda frente a una pregunta se seleccionara la opción de No sé. Además, al entregar las encuestas esta orientación se reiteró verbalmente, con énfasis persuasivo.

Estructura del informe

El presente informe está compuesto por la Introducción, Capítulos 1, 2 y 3, Conclusiones, Recomendaciones, Bibliografía, Glosario de términos, Apéndices y Anexos.

En el **Capítulo 1 – “Origen, Naturaleza y Evolución del Software”**, se exponen los elementos principales de la historia del software, desde su origen hasta el presente, su interacción con el hardware, de manera que se evidencien las relaciones sistémicas que contribuyen a explicar la naturaleza y evolución de este producto intelectual como extensión del producto industrial que lo soporta, el hardware, así como del surgimiento y evolución de su peculiar industria, con énfasis en los factores que la mueven y diferencian, especialmente en lo relativo a la colaboración en el desarrollo y las contradicciones con su propiedad y comercialización. Se destaca el rol especial de los Sistemas Operativos como productos

vertebradores de los demás y la influencia del UNIX como hilo conductor del proceso. También expone la evolución del concepto genérico de Software Libre, considerando sus diferentes momentos, modalidades y demás aspectos de interés. Concluye con el examen de algunos conceptos claves respecto al objetivo principal de esta investigación.

El **Capítulo 2 – “Aspectos Legales del Software con Énfasis en el Libre”**, se centra en los aspectos legales más comunes inherentes al software en general y en particular al libre, con el objetivo de ofrecer los contenidos y conocimientos comunes que sobre esos aspectos deben dominarse por los especialistas desarrolladores, productores y comercializadores del mismo. También examina en su evolución la forma creciente en que estos aspectos entrañan un proceso filosófico y de lucha social, que promueve y soporta la naturaleza intelectual de esa producción y su implicación en los procesos colaborativos que inducen, promueven y soportan como modalidad avanzada de producción y acceso colectivo al conocimiento.

En el **Capítulo 3 – “Diagnóstico del nivel de Conocimiento sobre SWL en la UCI”**, mucho más breve, se exponen y analizan los resultados del diagnóstico sobre el estado del conocimiento de los colectivos de la universidad acerca de los conceptos básicos y generales de la industria del software y en particular del software libre. Con ello también se contrastan los principales aspectos que caracterizan a la situación problemática en la que se centró esta investigación, y sirve de base para su introducción en la práctica de la Universidad al posibilitar la recomendación de acciones concretas y bien definidas para elevar el nivel de conocimientos actual, proceso para el que los contenidos de este informe pudieran ser de alguna utilidad inmediata.

Origen, Naturaleza y Evolución del Software

1.1 Introducción

La nueva sociedad es abierta, espléndida, en red, interconectada e informacional. Esa es la nueva realidad que se ha encumbrado en el horizonte, la Sociedad de la Información. En el contexto del libre acceso a la información y la generalización del conocimiento, el presente capítulo se centra en el análisis de los principales procesos de colaboración e intercambio mundial de SW.

Se partió desde sus más tempranos orígenes en la década de 1950, buscando esencialmente los hechos que originan al movimiento del SWL, caracterizado por un voluntarismo y fraternidad intelectual que lo definen como un modelo de perfeccionamiento informacional afín con las corrientes culturales de estos días. Movimiento cuyos principios son la base para la instauración de un ciberespacio público, apoyado en servicios de información a bajo costo y bibliotecas digitales de alcance mundial.

1.2 Un poco de historia

Aunque se desee ser breve hablando de SW, es difícil conseguirlo. Sin embargo, se sintetizará mucho su historia, enfatizando solo en determinados momentos que han marcado hitos principales en su evolución, especialmente en aquellos que evidencian la génesis de sus diversas formas de desarrollo, producción y distribución, con el objetivo de contribuir, en la medida posible para un trabajo de diploma de pregrado, a esclarecer conceptos y criterios que giran en torno a estos procesos.

Tal vez para muchas personas el SW propietario es la variante más natural de esta singular rama de productos, probablemente muchos consideren que es la más segura y la que proporciona la mayor calidad y evolución. Sin embargo, el SW nació en ambientes de colaboración e intercambio entre grupos de usuarios que compartían determinados intereses y permaneció así durante no pocos años, aunque con el paso del tiempo el escenario fuera diversificándose sensiblemente, cobrando también mucha fuerza la modalidad propietaria.

En paralelo se mantuvieron las formas colaborativas entre comunidades de investigadores académicos y corporativos, surgiendo movimientos que potenciaron esas alternativas, diversificándose sensible y complejamente en sus criterios, motivaciones, enfoques y objetivos, en la medida que el acervo de conocimientos en este campo, el avance de las tecnologías, especialmente de las redes. Por ejemplo, desde el surgimiento de Usenet a inicios los años de 1980, contribuyó a que esas comunidades se interconectaran más estrechamente, colaborando e intercambiando mucho más fácil y ampliamente. Ello fue potenciado grandemente por el abaratamiento de las computadoras personales y el surgimiento de la www de Internet a principios de los 90 del siglo pasado, propiciando también la incorporación de cantidades importantes de personas de manera individual a los diversos grupos, el incremento y diversificación de estos últimos, rebasando con creces y diversificando los límites, posibilidades e intereses de los grupos académicos y corporativos que les antecedían desde los orígenes.

En el presente, después de alcanzar un mayor grado de conocimientos, perfeccionamiento y variantes de organización, se encuentran vías y posibilidades muchos más claras de alcanzar objetivos mucho más avanzados para que las herramientas fundamentales del SW puedan ser fuentes abiertas con las libertades necesarias para poder llegar a cada necesidad del desarrollo humano, desde las individuales como forma de aprendizaje y disfrute intelectual, hasta las más sociales en sus diversos entornos y formas organizacionales. Esas ideas y cambios que continúan buscando libertad, tienen su génesis en el mismo origen del SW, pasando por modalidades crecientemente más sociales y progresistas en las décadas de 1960 y 1970, con un importante hito en la primera mitad de los 1980, cuando apareció formalmente el concepto del “Software Libre” y su correspondiente movimiento, al que le ha seguido una gran diversificación, incremento de objetivos y modalidades.

1.2.1 El inicio

Durante los primeros años el mundo de la informática estuvo dominado por las grandes computadoras o computadoras centrales (mainframes) como se les conocía, básicamente al

servicio de empresas y centros gubernamentales. Cuando se adquiría una computadora, esta venía acompañada del software, término que en dicha época era poco conocido o quizás se desconocía por completo (Ver Apéndice A).

Mientras se pagase el contrato de mantenimiento, se disponía de la documentación y de los pocos programas que suministraba el fabricante. No se veía al software como algo independiente, al menos en el mundo comercial. Se tomaba sólo como un añadido, no se compraba ni se vendía por separado, sencillamente se entregaba como un complemento necesario para dotar al hardware de mayor funcionalidad, aplicabilidad y utilidad, sin la presencia de contratos o licencias u otras especificaciones y restricciones legales relativas al mismo.

El mercado negocio estaba en la venta de las costosas computadoras. La motivación fundamental de las grandes compañías productoras de hardware como IBM, Remington Rand -devenida en Sperry Rand y luego en Unisys-, General Electric, Control Data Corporation, Digital Equipment Corporation (DEC) y otras, era el desarrollo y ventas crecientes del hardware (HW). Este cambiaba continuamente como ha caracterizado al sector en sus cortos sesenta años de historia, con su característica relación inversa y casi exponencial de la reducción de sus costos y precios e incremento de su capacidad de almacenaje de información y velocidad de operación. El gasto en computación era esencialmente del hardware.

La diversificación del HW hasta mediados de los años de 1970 y los varios desarrolladores y fabricantes se erigían en un problema más serio para el desarrollo del software y las aplicaciones en general, toda vez que cada uno creaba y modificaba sus arquitecturas propias, las que si eran protegidas legalmente, restringiendo el acceso a su know how como forma de intentar mantener la competitividad e incremento del segmento de mercado dentro de la competencia empresarial característica de la sociedad capitalista.

El fenómeno anterior hacía mucho más complejo el desarrollo del SW, toda vez que el HW era incompatible desde el repertorio o conjunto de instrucciones (ISA por sus siglas en inglés) y demás elementos de su arquitectura, incluyendo incompatibilidad entre las

computadoras de un mismo fabricante. En buena medida esta enorme dificultad fue uno de los factores principales de la llamada crisis del software.

Inicialmente la programación se realizaba directamente en código de máquina. Los ensambladores del lenguaje nemotécnico constituyeron una de las primeras herramientas de SW desarrolladas por los fabricantes. No fueron pocos los casos de creación de ensambladores y macro-ensambladores por parte de los usuarios. Promovido fuertemente por la IBM, la aparición del FORTRAN constituyó un hito importante para el desarrollo de las aplicaciones. Suministrarlo gratuitamente, con su código fuente y toda la documentación se le convertía en un añadido promocional de mucha fuerza para el incremento de las ventas del HW. Hechos como estos ponían en tensión la competencia entre los fabricantes y especialmente a sus fuerzas de ventas, por ello no resulta raro que la creación del primer grupo de usuarios de la IBM para el desarrollo y el intercambio gratuito de software con toda su documentación fuera una iniciativa de uno de sus agentes de ventas.

La génesis de la creación de una asociación cooperativa para el desarrollo de programas y aplicaciones de computadoras fue la propuesta de R. Blair Smith, un agente de ventas de la 701 de la IBM de la oficina de Santa Mónica, quien había vendido esas máquinas a la RAND Corporation y a la Douglas Aircraft Company, a quienes les preocupaban mucho los costos en que tenían que incurrir para programar sus aplicaciones. Thomas J. Watson Sr., presidente de la IBM, fue inicialmente hostil a esa idea, porque no veía en un grupo de usuarios una verdadera posibilidad de superar el “cuello de botella” que representaba la programación, el futuro SW, accediendo finalmente a su creación. De esa manera en noviembre de 1952 se realizó la sesión inaugural del primer grupo colaborativo para el desarrollo del SW, bajo el nombre de Digital Computer Association (DCA). Esto estimuló el surgimiento también de las primeras modalidades de empresas de programación, bajo la forma de industrias contratistas de programación o software [Campbell-Kelly, 2004].

Casi dos años después, en mayo de 1954 IBM anunció la entrada al mercado de su nueva computadora, la 704 como sucesora tecnológica superior de la 701. La DCA se reemplazó en ese momento por una modalidad más avanzada de grupo de usuarios, cuya condición

necesaria y suficiente para pertenecer y recibir todos sus beneficios consistía en haber adquirido o ser propietario de una IBM 704. El nombre de la DCA se sustituyó por el de SHARE (Compartir), cursándose las invitaciones para su sesión inaugural en agosto de 1955 por su secretario pro tempore, Fletcher Jones de la RAND Corporation [Campbell-Kelly, 2004].

A esa iniciativa le siguieron otras parecidas, cada una con su propia plataforma. Una que alcanzó mucho prestigio y cuyos resultados se utilizaron ampliamente en Cuba en la década de los 1970 fue DECUS (Digital Equipment Computer User Society) (Parte de DECUS existe en la actualidad dentro de la HP y opera en Europa con otras modalidades). Sus materiales se caracterizaban por ofrecer el código fuente comentada de los programas, junta con otras características de mucha utilidad. De modo que en esos primeros años de la informática el trabajo era colaborativo, relativamente libre; todos los que tenían acceso a él podían disponer normalmente del código fuente; era común compartirlo, modificarlo y distribuir las modificaciones [Pascual, 2007].

1.2.2 Nacimiento del BASIC

En 1964 vio la luz el lenguaje BASIC original (Beginners All-purpose Symbolic Instruction Code) desarrollado por los profesores John George Kemeny y Thomas Eugene Kurtz en el Dartmouth College. Concebido con fines docentes, para que sus estudiantes pudieran aprender programación más fácilmente, de manera que lograran compartir, modificar y distribuir sus trabajos, para conseguir mejoras, fruto del trabajo en conjunto. Destinado al uso colectivo por estudiantes y profesores en primer lugar, consecuentemente al registrarlo bajo copyright (CR) lo declararon directamente en el “*dominio público*”, divulgando su código fuente y demás características, garantizando así que estuviera al acceso libre de las comunidades a las que les pudiera interesar. Ello facilitó que en los años siguientes aparecieran otras versiones de BASIC, distinguiéndose del original de Kemeny y Kurtz que era conocido como BASIC Dartmouth.

En su momento este lenguaje satisfacía las necesidades para la escritura y ejecución de programas. Además, es bueno recordar que esa época las computadoras eran de gran

tamaño y sus precios muy altos, lo que limitaba su uso a una relativamente pequeña minoría. Las características del BASIC permitían acceder simultáneamente a una misma computadora desde varias terminales mediante sencillos sistemas de tiempo compartido (time sharing systems), para realizar sus programas y simplificar la complejidad del acceso a las computadoras, a diferencia de los demás lenguajes y sistemas disponibles, al mismo tiempo que así se abarataban sensiblemente los costos y precios de la hora de computadora al repartirse entre varias terminales.

El extraordinario desarrollo científico y tecnológico de la computación desde 1945-46 ha producido transformaciones inimaginables desde un punto de observación a 10-15 años de las futuras transformaciones e impactos en la ciencia, tecnología, economía y la sociedad en este campo. Las primeras computadoras eran enormes, extraordinariamente caras, muy consumidoras de energía, inestables y difíciles de comandar. Fueron reduciendo su tamaño, aumentando su eficiencia y su velocidad de procesamiento hasta aparecer las minicomputadoras a mediados de la década de 1960, con las que sus costos por ejemplo se redujeron en más de diez veces respecto a las mainframes del momento. Con la aparición del microprocesador unos diez años después de las minis, surgieron las microcomputadoras como la Altair 8800 y muchas otras. Un poco después se comenzó a hablar de computadoras personales o PC según sus siglas en inglés. Los precios de estas comenzaron a oscilar alrededor de los 2000 dólares norteamericanos, en promedio más de diez veces más baratas que las minis, produciéndose así una verdadera y creciente explosión de usuarios. A este último hito se sumó el estándar PC de la IBM, rompiendo así el paradigma tradicional del HW, junto con el rol jugado por la industria microelectrónica, y especialmente por Intel.

Hoy día además de hablarse de la laptop de 100 dólares (OLPC - con la idea que pueda llegar a una laptop por niño), recientemente las empresas ASUS y Acer lanzaron la llamada mini-laptop, de excelentes prestaciones, pero con una singularidad y un interesante llamado a las comunidades del SWL: *Han iniciado con un precio de unos 500 USD si se adquieren con el SO Windows y alrededor de 350-400 en el caso de optarse por la versión con el alguna de las distribuciones con el kernel Linux; uno de los mensajes y lectura implícita de*

esto último es que la licencia del SO casi alcanza el 50% del precio al que puede adquirirse el hardware con un sistema operativo de una distribución bajo licencia de SWL.

Regresando la mirada a mediados de la década de 1970, debe recordarse que la irrupción de las microcomputadoras no fue diferente a las generaciones anteriores, lo hacían sin o con muy poco software. Ello junto a las características del paso a memorias semiconductoras y la no disponibilidad de memorias externas (discos por ejemplo) a precios razonables, hizo pensar a muchos que el futuro de las micro no era claro en un plazo razonable. Sin embargo, todo este desarrollo singular de la microelectrónica y el HW de las microcomputadoras propició la diseminación casi masiva del BASIC, que naciera de la idea original del Dartmouth College. Este fenómeno tuvo su máxima expresión en la ya mencionada Altair 8800 a mediados de la década de 1970.

1.2.3 Papel de Bill Gates

Es aquí donde entra a escena el hoy muy conocido y célebre hombre de negocios William Henry Gates III¹. Para ese entonces no le resultó muy difícil a Bill Gates, junto a Paul Allen, ambos estudiantes de Harvard, darse cuenta de lo que necesitaba la Altair: *SW, un lenguaje para crear los programas*. Sin perder tiempo establecieron contacto con la pequeña compañía de Albuquerque, fabricante de Altair, convenciendo a su director de que serían capaces de elaborar lo que su micro necesitaba. Para cumplir su compromiso contractual escogieron el BASIC, del que ya se ofrecieron los elementos sobre el origen y destino para el que fue concebido. Resultaba el lenguaje ideal para programas cortos, además de su más fácil implementación y uso, por tratarse también de ser Interpretativo desde su versión original.

Fue esa la oportunidad que, aprovechada oportuna y brillantemente, convirtiera al BASIC que surgió totalmente libre, en el primer producto exitoso del software de la naciente

¹ Es conveniente señalar que William Henry Gates III desciende en segunda generación de familias dedicadas a la actividad bancaria. Aunque estudiaba informática y sabía programar, sin dudas que desde hogar recibió inspiración y formación en negocios y finanzas, como ambiente natural de background de alguna manera.

Microsoft (MS). Esta compañía se lo apropió e inició la extraordinaria batalla egoísta de patentar o proteger legalmente su versión mediante el CR (versión norteamericana del derecho de autor), convirtiéndose este hecho en la *“nave insignia”* de la nueva modalidad del software propietario de propósito crecientemente masivo y fuente inicial de la increíble obtención de riquezas personales, iniciándose así la carrera de la mayor empresa del mundo dedicada al software empaquetado para uso masivo. Es conveniente aclarar que existe el mito de que MS es la mayor empresa de SW del mundo y hasta que domina ese mercado global, sin embargo la realidad es que al cierre del año fiscal del 2007 sus ventas solo representaron el 10% de de las ventas totales de las 500 mayores empresas de software, ocupando el segundo lugar de esa lista [Software Magazine, 2008].

Los aficionados al desarrollo de SW, que se sentían parte del movimiento colaborativo de su distribución y mejora, estuvieron rotundamente en desacuerdo con las acciones de Bill Gates. En réplica a ello obtuvieron una copia del MS BASIC que se distribuía en “cartuchos” para la Altair y otras, procediendo a diseminarlo de forma gratuita, argumentando que ese SW había sido y debía seguir siendo un bien público, de acceso ilimitado, etc. El presidente de MS en total divergencia con esas posiciones y acciones, denunció abiertamente a los que compartían y contribuían a la difusión del Altair BASIC sin proporcionarle a él la remuneración monetaria establecida, y los llamó *“piratas”*, calificando a ese software de *“robado”* en vez de *“compartido”*. Esto lo hizo a través de una carta abierta a los aficionados sobre “piratería”, donde los acusaba de robo y exponía que sin un pago adecuado nadie podría ser capaz de producir SW profesional y de calidad (Ver Anexo 1).

El marco de esa comunicación le fue propicio para exhortar a estos amantes del SW a colaborar con él y unírseles para ofertar al mercado lo que el llamaba *“buen software”*. Así se dispersaron de esas comunidades muchos de sus programadores, para dedicarse al mercado y al software propietario, fuertemente influenciados e incentivados por Bill Gates, capaces a partir de entonces de propagar el *“nuevo descubrimiento”*: esa modalidad egoísta, desleal y altamente restrictiva del comercio del SW “cerrado, sin garantías y derechos” de un bien intelectual de necesidad masiva creciente, incorporándose de esta manera a uno de los sectores en los que se enmarcaba la industria del SW, el de los desarrolladores de productos

de software para el mercado en masa o masivo (Ver Apéndice A). Parecía que el SW pagado había pasado a ser la única elección para muchos que decidieron apoyar esta nueva forma de intercambio remunerado.

Lo que Bill Gates olvidaba es que el BASIC que empleó, había nacido bajo el dominio público, de modo que no tuvo que pagar un solo centavo para hacerse de una copia, creado por informáticos que no reclamaron compensación monetaria por esa vía, y que además había sido continuamente perfeccionado por centenares de programadores anónimos, sin ánimo de lucro ni reconocimientos especiales. Pero a él poco le importaban los basamentos de esa *“ética comunitaria”* [Burke, 1999].

1.2.4 Inicios de UNIX

El origen y desarrollo de los Sistemas Operativos (SO), además de ser la componente clave de las plataformas y herramientas del SW, constituye quizás la fuente principal de la continuidad del desarrollo colaborativo del software, el objetivo que ha dado continuidad desde los primeros de alguna importancia, que fueron demandados por las características y potencialidades del HW de mediados de la década de 1960 en adelante.

La mayor y principal parte de la historia del SW colaborativo, el surgimiento del movimiento del SWL, así como su evolución y modalidades hasta el día de hoy puede perfectamente derivarse y extraerse sin mayores dificultades de la historia de los SO y en especial del UNIX, aun cuando las versiones de hoy sean muy diferentes y evolucionadas con respecto a sus versiones originales de finales de la década de 1960, hecho indispensable e inevitable ante la necesidad y excepcionalidad del desarrollo de la ciencia y la tecnología de la computación o informática, como acepción más genérica que se utiliza en este trabajo.

Retomando lo señalado en el epígrafe 1.2.1 sobre el factor clave de la incompatibilidad del HW entre fabricantes diferentes e incluso dentro del mismo, ello además de constituir uno de los factores clave de competitividad, a su vez inducía a que esas empresas estimularan la colaboración en el SW y no se motivaban fuertemente a convertirlo en líneas de productos comerciales.

El fenómeno anterior se manifestaba de diversas maneras, siendo causa importante de varias dificultades y controversias. En gran medida contribuía a la llamada crisis del SW. Por otra parte, se desarrollaba un determinado elitismo entre los especialistas de la ciencia y la tecnología de la computación, favoreciendo a los dedicados al HW, lo que no estimulaba favorablemente a la dedicación y al apoyo del desarrollo del SW, especialmente dentro de las organizaciones y empresas que incluían el desarrollo del HW, mayoritarias durante largo período. Este último fenómeno se manifestó también fuertemente en Cuba entre 1969 y mediados de la década siguiente.

Para las computadoras de la primera y segunda generación apenas eran muy necesarios los SO. Su importancia fundamental era aprovechar al máximo el costoso tiempo del procesador central ante la demora que representaban los procesos de entrada y salida (E/S). Quizás el primero de esos SO lo fue el FSM (FORTRAN System Monitor) al cual le bastaban con algunos cientos de líneas de código, no rebasando las 10000 líneas en los casos más desarrollados de esas generaciones. Con el advenimiento de la tercera generación y en especial con los sistemas 360 y 370 de la IBM esa situación cambió dramáticamente, aun cuando no rebasaban mucho a los sistemas en bache.

El escenario y requerimientos cambiaron radicalmente al desarrollarse la cultura de tiempo compartido y la multiprogramación. Adicionalmente los retos aumentaron increíblemente con la incorporación gradual de la conexión a líneas de comunicaciones y las primeras redes muy elementales. El MIT desarrolló uno de los primeros proyectos serios de SO en tiempo compartido, el CTSS (Compatible Time Sharing System) para una IBM 7094 especialmente modificada. Estos sistemas no ganaron en popularidad hasta que el hardware no garantizó la seguridad de la memoria compartida, hecho que se logró con la tercera generación [Tanenbaum A. S., 2006].

A partir del relativo éxito del CTSS de MIT, la General Electric, uno de los principales fabricantes de computadoras en aquel entonces, intentando competir fuertemente con la IBM y otros, decidió unirse con los grupos de SO de MIT y los Bell Labs para desarrollar lo que llamaron un sistema de “utilitarios de computación” (computer utility), consistente en una

máquina que pudiera atender simultáneamente a cientos de usuarios en tiempo compartido. Como modelo tomaron a los sistemas de distribución de electricidad –cuando usted necesita potencia y energía eléctrica, simplemente conecta un tomacorriente a la pared y, dentro de lo razonable, allí usted encontrará la potencia y energía que necesita. Este fue el origen del MULTICS (MULTiplexed Information and Computing Service), concebido para una máquina poderosa que fuera capaz de suministrar potencia de cálculo a todos los que la necesitaran en el área de Boston. Este proyecto no triunfó, aunque finalmente el MULTICS se concluyó, vendiéndose decenas de copias, muchas de las que trabajaron hasta la década de los 90 en EE.UU. y la última funcionó en Canadá hasta el año 2000. La idea de una máquina, que fuera mucho más potente que la mainframe GE-645, que pudiera venderse alrededor de mil dólares a millones de usuarios solo uno 30 años después de ese momento, hubiese sido tomada como pura ciencia ficción, algo así como la idea de concebir que fuera posible diseñar y construir un tren submarino supersónico y trasatlántico [Tanenbaum A. S., 2006].

Independientemente del fracaso de los objetivos iniciales del MULTICS y de la retirada de la GE, algunos de los científicos de MIT y Bell Labs persistieron en la idea y concepción de tan novedoso sistema. Ken Thomson de Bell Labs fue el principal exponente de esa idea. Alrededor de 1967-68, cuando le rechazaron su propuesta de comprar una computadora para desarrollar una versión de simple usuario del MULTICS, encontró en los laboratorios una pequeña minicomputadora PDP-7 en desuso, en la que se sentó a escribir su versión simple, la que un poco después desarrolló y designó como el SO UNIX, el que devino muy popular en el mundo académico, así como en muchas agencias gubernamentales y compañías privadas [Tanenbaum A. S., 2006].

A la primera versión del UNIX surgida en 1969 le llamaron UNICS y se ejecutaba, como ya se dijo, en una vieja minicomputadora DEC PDP-7. Este primer UNIX estaba escrito en un lenguaje llamado B. El trabajo de Thompson impresionó a sus colegas de los laboratorios Bell de tal forma que pronto se le unió Dennis Ritchie y más tarde todo el departamento. Lo

primero que hicieron fue portar el UNIX de la obsoleta PDP-7 a las modernas PDP-11/20, PDP-11/45 y PDP-11/70².

En aquel entonces Dennis Ritchie se había convertido en compañero de Thompson en los laboratorios Bell, dándose a la tarea de diseñar un sucesor del lenguaje B, al que llamaron C, ofreciendo así un lenguaje que pudiera usarse para escribir una versión portable del sistema UNIX que no estuviera completamente escrito en ensamblador. Con este lenguaje, en el año 1973 Ritchie y Thompson rescribieron el UNIX, propiciando una visión distinta al diseño del software: Solucionar un problema interconectando herramientas sencillas en vez de crear grandes programas monolíticos. Adicionalmente se ofrecía una plataforma avanzada para garantizar la portabilidad de los SO con poco esfuerzo adicional, brindando así una vía para la solución por SW al problema de la incompatibilidad del HW.

De esta manera, a la retirada de la GE de un proyecto de SW “colaborativo” para vender más HW, la relevó la clara colaboración entre investigadores académicos y corporativos, la que rompía en cierta forma las fortalezas competitivas de los vendedores de HW, como modalidad que se volvía obsoleta. Ese actuar y resultados constituyen un buen ejemplo de la transformación y continuidad del desarrollo del SW en modalidad colaborativa y buenas prácticas, de lo que el nacimiento del UNIX constituye un modelo crucial y sin lugar a dudas el de la continuidad del desarrollo de los SO. Este sistema nació con el propósito de demostrar que era posible construir un sistema operativo que ofreciera un ambiente de trabajo cómodo y mucho más sencillo. Es el mismo UNIX que siguió inspirando a numerosos grupos colaborativos.

En 1974 se da a conocer el nuevo Sistema Operativo UNIX, generándose de esta manera un gran entusiasmo en la comunidad académica, pues lo ven como una alternativa para la

² DEC creó el concepto y arquitectura de minicomputadoras. Desarrolló varios modelos incompatibles, hasta llegar a la PDP-11, familia modelar que mantuvo compatibilidad ascendente del HW, desde la pionera PDP-11/20. La 11/70 fue una megamini en la que potenciaron el desarrollo del manager de memoria. Esto fue seguido por la familia VAX 11/780 (Virtual Address extensión), manteniendo compatibilidad con la arquitectura PDP/11 y de mucha influencia en el Intel 8086 y demás derivaciones hasta la actualidad.

enseñanza. Por problemas legales a la compañía AT&T se le impide comercializarlo. Debido a esto AT&T distribuye copias a numerosas universidades y agencias gubernamentales para propósitos educativos y de investigación, incluyendo una cantidad significativa de código.

Así llegó el UNIX a la universidad de Berkeley donde se empezó a utilizar con frecuencia y a hacersele mejoras, apareciendo un tiempo después una de sus alternativas más importantes, la *Berkeley Software Distribution* (BSD). La universidad poco a poco le fue incorporando mejoras significativas a su distribución, entre ellas el soporte para memoria virtual y el stack TCP/IP para comunicación en red.

Esto propició el desarrollo de dos versiones UNIX con culturas diferentes, que en un momento determinado se enfrentaron (la del UNIX del System Labs de AT&T con sus mecanismos de innovación y la del Computers Systems Research Group (CSRG) de Berkeley con su proceso cooperativo de innovación). Para la AT&T el éxito de BSD constituyó un problema, al punto que en 1992 llevó al CSRG a los tribunales por haber sobrepasado los términos de la licencia acordada y haber hecho públicos "secretos industriales". Este proceso jurídico solo tuvo fin en el año 1994, con la venta por parte de AT&T de los UNIX System Labs a la empresa Novell, la que no perdió tiempo para llegar a un acuerdo con el CSRG, aunque con ello no se logró que el proyecto BSD superara el proceso jurídico, pero si le permitió que distribuyera una última versión del sistema BSD llamada "4.4 BSD Lite", bajo licencia libre y desembarazada de toda huella del código perteneciente a la AT&T.

UNIX fue sin dudas un ensayo temprano de lo que se potenció años después con la iniciativa GNU y con la aparición y desarrollo del kernel Linux y su integración a las diversas distribuciones varios años más tarde, por mencionar momentos y grupos singulares. Estaba enmarcado en una comunidad mucho más pequeña, y era necesaria la licencia de AT&T, pero en otros aspectos su desarrollo fue similar en un mundo mucho menos comunicado. Con el tiempo, UNIX fue también un ejemplo temprano de los problemas que podían presentar los sistemas propietarios que a primera vista tenían alguna característica del SWL.

Es importante precisar que en la primera mitad de la década de 1980 aparecieron, de forma organizada y consciente, los primeros proyectos para la creación de sistemas compuestos bajo la iniciativa de software libre conocida como GNU, surgiendo así el concepto de SWL como movimiento explícito. Probablemente lo más importante de este hecho son sus fundamentos éticos, legales y económicos-comerciales. Su desarrollo se ha mantenido hasta el presente, cada vez con mayor fuerza y sustentabilidad, no solo por sus fundamentos originales, sino por la diversificación y actualización de ese movimiento, lo que hace mucho más complejo y difícil el establecimiento claro de su verdad actualizada. Es importante destacar que desde el principio de la historia de UNIX, la posibilidad de tener acceso a las fuentes de los programas ha sido una constante, y que en gran medida el éxito de este SO se debe a ese movimiento y sus derivaciones [Pascual, 2007].

No obstante, desde mediados de la década de 1970 era completamente usual encontrarse con SW propietario en cualquier ambiente informático. Ello propició un gran cambio cultural y mucha confusión entre los profesionales del SW, a la vez que un gran número de empresas florecían en torno al nuevo negocio. Mucho antes de que Microsoft comenzara a jugar su rol en la industria del SW, hubo numerosas iniciativas que mostraron características de lo que luego se consideraría SWL, solo que no existía un concepto claro de lo que se hacía. Sus prácticas tributaban a un SW colaborativo bien semejante al SWL de la actualidad, y quizás un poco más entregado al dominio público que el que se promueve en el presente. Faltaba aún casi una década para que empezase a aparecer de forma organizada y como respuesta a esta situación lo que hoy se denomina SWL. Durante todos esos años el movimiento continuó cultivando la idea del SW como bien común, de libre acceso al código fuente y de intentar que lo libre no pudiera convertirse en propietario.

El término “libre” asociado al SW al surgir en EE.UU., en idioma inglés el término “free” tiene dos acepciones correctas en dependencia del contexto en que se utilice. Una de ellas es “gratis-exento de pago” y la otra es “libre-de libertad”. Realmente libre en el contexto del SW y según los principios del movimiento que lo hizo nacer, significa libertad y no gratis. Ello no se opone a que si alguien quisiera distribuir el SWL de manera gratuita, pueda hacerlo.

Quizás el hecho de que cada vez son más personas la que necesitan algún producto de SW, genera ese deseo romántico de que sea gratis. Ello hace reflexionar en la necesidad de que estos productos puedan llegar de manera razonable a quienes los necesiten, independientemente de su mayor o menor solvencia económica.

Por otra parte, esa aparente ambigüedad del término no debe confundir induciendo a la falsa idea de que el SWL no sea comercializable, independientemente de que ello se realice en actividades no lucrativas, menos lucrativas o del nivel de lucro que resulte razonable. Lo importante es la libertad para su desarrollo y reutilización, dentro de las garantías y restricciones que deban garantizarse, porque además es una situación que también corresponde al campo de la ética y de los mejores valores humanos.

1.2.5 Stallman y el nacimiento de GNU y FSF

De acuerdo a Richard Stallman, un programador del Massachusetts Institute of Technology (MIT), la comunidad de software compartido en MIT existió por “muchos años” antes de que él se involucrara en ella en 1971. Con el paso del tiempo abandonó su trabajo para comenzar un nuevo proyecto. Él mismo se consideraba un hacker de los que disfrutaban compartiendo sus inquietudes tecnológicas y su código. Notaba con enfado cómo su negativa de firmar acuerdos de exclusividad y no compartimiento le estaban convirtiendo en un extraño dentro de su propio mundo, y cómo el uso de SW propietario en su entorno le dejaba impotente frente a situaciones que antes podía solventar fácilmente.

La causa principal por la que Stallman decide abandonar el MIT para iniciar el proyecto GNU se derivó de una impresora que tenían en el laboratorio, la que presentaba problemas con la provisión de papel, atascándose habitualmente y sin otra forma de descubrirlo que trasladándose hasta ella. Intentando solucionar el problema Stallman contactó a los fabricantes, con la idea de modificar el SW de control del equipo de manera que enviara una señal cuando se atascara, evitando así la pérdida de tiempo de trabajo. Los fabricantes se negaron a facilitarle el código fuente del driver de la impresora. Este suceso colmó su contrariedad e hizo que terminara de consolidarse su idea de que el código fuente de los programas tenía que ser de libre acceso para todo el mundo. De esta manera se consolidó

su idea de renunciar al MIT para construir un sistema de SW completo, de propósito general, pero completamente libre. El sistema y el proyecto que se encargaría de hacerlo realidad se llamó GNU (acrónimo recursivo, GNU 's Not UNIX).

Desde el principio del proyecto GNU, Stallman estaba preocupado por las libertades que tendrían los usuarios del SW. Le interesaba que no sólo los que recibieran los programas directamente del proyecto GNU, sino que quien lo recibiera después de cualquier número de redistribuciones y quizás modificaciones, siguiera disfrutando de los mismos derechos: modificación, redistribución, etc. Para ello, escribió la licencia GPL, probablemente la primera licencia de SW diseñada específicamente para garantizar que un programa fuera libre en este sentido. Al mecanismo genérico que utilizan las licencias tipo GPL para conseguir estas garantías, Richard Stallman lo llamó Copyleft, que hoy día es el nombre de una gran familia de licencias de SWL. Así fundó una organización en 1984 que denominó Free Software Foundations (FSF) con el fin de conseguir fondos para el desarrollo y la protección del SWL y sentó los fundamentos éticos del mismo, con documentos como “El Manifiesto GNU”.

A principios de la década de 1990, el proyecto GNU estaba muy cerca de tener un sistema completo similar a UNIX. Aun así, hasta ese momento no había producido una de las piezas fundamentales: el kernel del SO (el núcleo del SO que se relaciona con el hardware y permite que todo funcione). Sin embargo, el SW de GNU era muy popular entre los usuarios de las distintas variantes de UNIX, por aquella época el sistema operativo más usado en las empresas. Además, el proyecto GNU había conseguido ser relativamente conocido entre los profesionales informáticos, y muy especialmente entre los que trabajaban en universidades. En esa época, sus productos ya gozaban de una merecida reputación de estabilidad y calidad.

Stallman siempre había soñado con sustituir el UNIX por algo que fuera igual de bueno, acompañado además del código, pero era una larga y dura tarea. Lentamente pero de forma segura, el proyecto GNU iba ensamblando las partes para hacer que funcionase. A finales de la década de los ochenta, principios de los noventa había centenares de pequeñas utilidades y herramientas mayores donadas al proyecto GNU, y todos esos pequeños fragmentos se

iban añadiendo. Ya entonces muchas aplicaciones libres eran las mejores en su campo (utilidades UNIX, compiladores). Sin embargo, para terminar el rompecabezas faltaba la pieza esencial: el núcleo del sistema operativo. El proyecto GNU estaba buscando esa pieza con un proyecto llamado Hurd, que pretendía construir un kernel con modernas tecnologías [Pascual, 2007].

1.2.6 Historia del MINIX y algunos de sus roles

Cuando el UNIX era joven, (Versión 6) su código fuente estaba ampliamente disponible, bajo una licencia de la AT&T, por lo que era posible estudiarlo cuando se necesitara. John Lions, de la Universidad de New South Wales en Australia, escribió un folleto explicando sus operaciones línea por línea. Este folleto se utilizaba como texto de la asignatura y cursos sobre sistemas operativos en muchas universidades, bajo la necesaria licencia de AT&T [Tanenbaum A. S., 2006].

Cuando esta compañía comenzó a distribuir la Versión 7 del UNIX, lo hizo bajo una modalidad propietaria, mediante la que prohibía el uso del código fuente y cesaba su distribución por esa vía. Muchas universidades se plegaron a esa prohibición y dejaron de impartir esa enseñanza. Desafortunadamente enseñar solo teoría, sin poder desarrollar casos de estudio es algo de limitado valor y alcance. Para remediar esa situación el profesor Tanenbaum decidió escribir un nuevo SO que fuera compatible con el UNIX desde el punto de vista de los usuarios, pero completamente diferente a su estructura interior y códigos fuente y de máquina. Al no utilizar ni un solo renglón de código de la versión propietaria de AT&T, este sistema evitaba las restricciones de licencia impuestas. Al igual que el UNIX, el MINIX 1.0 fue escrito en C y liberado en 1987 con su código fuente a la disposición de todo el que quisiera o necesitara utilizarlo. El nombre MINIX sugiere la idea de mini-UNIX porque es lo suficientemente pequeño para que cualquiera no experto pueda comprender como este trabaja [Tanenbaum A. S., 2006].

Además de resolver las graves restricciones de la nueva licencia propietaria y propiciar la continuidad de la enseñanza de esos conocimientos acumulados y su desarrollo ulterior, el MINIX tiene la ventaja de que al ser creado dos décadas después del originario y una

posterior a las versiones más generalizadas, de ser estructurado y mucho más modular. Por ejemplo, a partir de sus primeras versiones el sistema de gestión de ficheros y la administración de memoria ya no eran parte del SO, corriendo bajo el modo usuario. Estas y otras modularidades permiten que sea mucho menos exigente en los recursos del HW que se necesitan [Tanenbaum A. S., 2006].

Sin dudas que la obra del MINIX constituyó un aporte significativo al desarrollo de la ciencia de la computación y al SWL en cualquiera de sus modalidades del concepto quizás más restringidas o más avanzadas, al menos mucho más modernas, como es el caso del de código abierto (open source). Para personas con algún conocimiento del MS-DOS, la existencia de MINIX y la disponibilidad de su código fuente es una buena motivación para disponer de una PC e intentar desarrollar y aportar nuevas piezas y mejoras al SW. Pudiera decirse que esa disponibilidad, junto a poder acceder a la red USENET crearon condiciones muy favorables para que Linus Torvald potenciara su creatividad y produjera el notable resultado que posteriormente se incorporó al movimiento GNU, incentivando también una gran diversificación y promoción del SWL, a la vez que ponía a la disposición de todos el kernel que no había logrado el movimiento en sus años de existencia.

1.2.7 El Kernel Linux

Linus Torvalds, estudiante finlandés de informática con 21 años, anuncia que ha empezado a trabajar en un núcleo libre tipo UNIX, usando herramientas de GNU, como GCC. Su meta en esa época era construir un MINIX de mayor alcance. Este estudiante no podía permitirse comprar una máquina con sistema UNIX, por lo que había decidido crear su propia versión. Sólo se podía permitir una PC básica con un procesador 386. Lo más importante quizás fuera que se sumaba al movimiento colaborativo de producir e intercambiar SW, incorporándose finalmente al GNU.

Hasta aquel momento UNIX era uno de los mayores sistemas operativos del mundo. Lo utilizaban las computadoras de IBM, Sun, Apple y la mayor parte de los fabricantes de máquinas de gran potencia. Todos tuvieron la necesidad de unificar el mercado

estandarizando UNIX, por lo que utilizaron el interface estándar UNIX del IEEE conocido como POSIX.

Torvalds continuó trabajando con su proyecto hasta crear un sistema que funcionaba con un compilador, tomando componentes del proyecto GNU. En 1991 Linus Torvald creó el núcleo o kernel de Linux. A principios de 1992, Linux ya no era simplemente un hobby de un estudiante finlandés. Varios programadores influyentes se habían interesado por el código que era libre y relativamente manejable. Ejecutaba gran parte del código GNU y permitía experimentar con algunas herramientas excelentes de una forma neta y barata. Crecía sistemáticamente la cantidad de usuarios que descargaba el sistema, y una parte significativa de ellos comenzaron a comunicar los errores y sugerencias a Torvalds. Estas personas se fueron comprometiendo con el proyecto, el que fue creciendo de manera no antes vista.

También puso en práctica la idea de recurrir a la inmensa comunidad de aficionados, programadores y hackers, invitándolos a poner en la red sus ideas y deseos de construir SW y compartirlo. El modelo de trabajo cooperativo, voluntario y difuso fue conquistando espacios cada vez con mayor popularidad y crédito. Un proyecto de tal magnitud no hubiese podido ser llevado a cabo por una sola persona.

Finalmente incorporó su proyecto a la modalidad de la licencia GNU - General Public License (GPL), la que atendiendo a sus principios, establece que todo el código fuente que Torvalds o cualquier otra persona escribiera para este sistema, debía ser accesible con entera libertad y compartido por todos los usuarios. Todo el SWL GNU anterior podía ser fácilmente conectado para que trabajase con el núcleo del sistema operativo, al que Torvalds ya llamaba Linux. Ése es el poder que tiene el código fuente al ser distribuido libremente.

Quizás parte de la genialidad de Linus Torvalds es el enfoque de ingeniería de SW con que manejó su proyecto en la etapa de la arquitectura de sistema, análisis, programación y prueba, en cuanto a métodos y roles más dinámicos a la hora de desarrollar el kernel de Linux, aprovechando todo el desarrollo que encontró a su disposición y las nuevas posibilidades que ofrecía el alcance que iba tomando Internet.

Linus llegó a la conclusión que para lograr hacer un sistema operativo que resuelva los problemas hay que involucrar a muchas personas haciendo pequeñas cosas, con una estrategia de ingeniería de SW en la que se puedan incorporar rápidamente las contribuciones de todo el mundo. La capacidad de monitorear, ordenar y estimular a muchos usuarios a que hagan más cosas en un proyecto, esto es un problema netamente de ingeniería de SW desde el punto de vista de organización, gestión y control.

Internet ha tenido mucho que ver en el desarrollo de Linux, ayudando a viabilizar que un montón de desarrolladores de todo el mundo trabajaran en colaboración para lograr su conformación. Indudablemente el desarrollo de Internet y de Linux de forma paralela no fue por casualidad, ambos se deben a las potencialidades y al éxito de la colaboración abierta y de la cooperación de muchas personas con un objetivo común [Pascual, 2007].

1.2.8 Aparición de Internet

Internet ha supuesto una revolución sin precedentes en el mundo de la informática y de las comunicaciones. Es a la vez una oportunidad de difusión mundial, un mecanismo de propagación de la información y un medio de colaboración e interacción entre los individuos y sus computadoras independientemente de su localización geográfica.

Desde su génesis Internet tuvo mucha relación con las colaboraciones que existían para el desarrollo del SW. Las comunidades que ayudaron a su aparición tenían claro muchos principios que un tiempo después se convirtieron en clásicos en el ambiente del SWL, entre ellos la depuración de errores y la compartición de código por citar algunos.

Este nuevo acontecimiento se generalizó rápidamente entre las comunidades del entorno informático, propiciando que sus costumbres y formas de actuar pasasen fácilmente de una comunidad a otra, y viceversa. Muchas de las aplicaciones básicas en el desarrollo de Internet, como Sendmail (servidor de correo) o BIND (implementación del servicio de nombres) se crearon por este movimiento, y en gran medida fruto de la colaboración entre comunidades.

Si se analiza detenidamente el desarrollo creciente que ha tenido Internet, no sería muy difícil percatarse que ocurre un fenómeno parecido al que suele ocurrir en los proyectos de SWL, donde mucha información es pública y existe un ambiente abierto a la colaboración entre personas o grupos que se encuentran geográficamente dispersos. El sistema de cooperación que respalda hoy al SWL se multiplicó significativamente con el nacimiento de Internet, aunque ya desde el origen del SW, existía colaboración entre personas e instituciones afines con estos temas como se ha ido reseñando en este trabajo.

1.3 Polémica entre software libre, código abierto y software gratis

Existen diversos conceptos e interpretaciones sobre software libre, código abierto, software gratis y otras expresiones o manifestaciones sobre este vasto, dinámico y revolucionario campo de actuación. Al final del epígrafe 1.2.4 se exponen algunas definiciones básicas que contribuyen a esclarecer estas confusiones y a orientarse mejor en estos conceptos. Buena parte de estas diferencias es necesario estudiarlas y comprenderlas desde sus aspectos jurídicos o legales, según el tipo de protección, restricciones y libertades que las diversas modalidades ofrecen.

Por otra parte, desde mucho antes de existir el SW apareció un fuerte movimiento por la cultura libre o de fuente abierta (open source). En la parte introductoria se explican algunos de estos elementos y se pone de manifiesto como el SW al ser una creación intelectual y convertirse en un factor recursivo para esos procesos, devino quizás en el centro principal del movimiento implícito o tácito por el código abierto, produciéndose también la ambigüedad que genera el término profesional relativo a la identificación del código legible para los humanos, que se designó como código fuente (source code). Inevitablemente el paso del tiempo necesariamente modifica, enriquece y perfecciona las ideas anteriores. Hoy constituye una polémica abierta en la que se discute si los factores de estímulo y metodología desarrollados por las comunidades de SWL serían de igual efectividad en los demás campos de la creación intelectual.

Algo más que una modesta contribución a la epistemología del SWL escapa con creces a los objetivos y modestas posibilidades de esta investigación. En el sentido más amplio es difícil

llegar ahora un poco más lejos, pero las fuentes básicas para esta investigación necesariamente hay que ubicarlas en los trabajos de la comunidad que de manera general debiera designarse como comunidad del Software Libre y de Código Abierto (SLCA o F/OSS como se está generalizando por sus siglas en inglés), donde son de importancia capital los trabajos de Richard M. Stallman y Eric S. Raymond, entre otros. También hay publicaciones de gran importancia como la revista Wired, la editorial O'Reilly and Associates entre otras muchas de importancia. Igualmente son fundamentales las contribuciones de las comunidades de investigación en ingeniería del software, cuyos ejemplos más relevantes se encuentran en la ACM y la Sociedad de Computación del IEEE [Joseph Feller, et all, 2005]. Similarmente en Europa y en otras latitudes pueden encontrarse fuentes de mucho valor y utilidad.

Con un mayor alcance habría que ofrecer resultados en al menos cinco partes o divisiones que son inevitables cubrirlas exhaustivamente para ofrecer un conocimiento más acabado sobre el SWL y para poder proyectar sus perspectivas. De manera general son las siguientes [Feller, et all, 2005]:

1. Motivaciones para el desarrollo del SW SLCA o F/OSS – SWL a los efectos de la terminología de uso generalizado en este trabajo.
2. Evaluación del Desarrollo del SW de las comunidades y distribuciones SLCA.
3. Procesos y Herramientas que caracterizan al SLCA.
4. Modelos Económicos y de Negocios que son inherentes al SLCA.
5. Aspectos Legales, de la Comunidad y de la Sociedad relacionados con el SLCA.

Insistiendo en que el alcance presente es mucho más reducido, a continuación se intenta ofrecer una mejor precisión sobre las principales modalidades del SWL. En el Capítulo 2 se presta atención especial a sus diferencias desde el punto de vista de los instrumentos legales y otros aspectos que las diferencian adicionalmente. Debe decirse que en estos

residen buena parte de las mayores dificultades para lograr una referencia más unificada y mejor compartida.

1.3.1 Software Libre

La definición siguiente se ciñe a los conceptos originales postulados por la FSF para los programas de SWL, como punto de partida del movimiento explícito y organizado con personalidad jurídica que esta creara. Queda claro aquí que el significado de “free” se toma en términos de libertad, no de gratuidad en la distribución de los productos como única modalidad. Según dichos postulados, el SWL como tal se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el SW, basado en las cuatro libertades genéricas que especifican lo anterior.

Tabla 1 Libertades del SWL

Libertad 0	Libertad 1	Libertad 2	Libertad 3
La libertad para ejecutar el programa sea cual sea el propósito.	La libertad para estudiar el funcionamiento del programa y adaptarlo a las necesidades (el acceso al código fuente es condición indispensable para esto).	La libertad para redistribuir copias y ayudar así a los vecinos.	La libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad (el acceso al código fuente es condición indispensable para esto).

Se considera SWL a todo aquel programa que cumpla con las libertades mencionadas anteriormente. El usuario de este SW tiene derecho a ejecutar, compartir, modificar, estudiar, redistribuir copias, con o sin modificaciones, de forma gratuita o por alguna compensación

económica con cualquier persona por distante que ésta se encuentre. Debido a las libertades que brinda este tipo de SW puede ser usado deliberadamente en cualquier entorno informático por cualquier ente social o cualquier organización, con el fin que se desee y sin necesidad de notificar lo que se realizó con él.

Para poder llevar a cabo las libertades de estudiar el funcionamiento del programa y adaptarlo a las necesidades del que lo precise (libertad 2) y para mejorar el programa y luego publicarlo para el bien de toda la comunidad (libertad 4) es necesario contar con el código fuente del programa. Es decir la accesibilidad del código fuente es una condición necesaria para el SWL y en específico para estas dos libertades que en su concepto se exponen.

También es importante aclarar el tema de la comercialización, muchas personas piensan que las condiciones de libertad del SWL no le permiten ser un producto comercial y esto es un error pues cualquier programa libre estará disponible para cualquier uso, desarrollo y distribución comercial. El desarrollo comercial del SWL ya no es extraño y de hecho ese SWL comercial es muy importante.

Las libertades definidas anteriormente están protegidas por licencias de *Software Libre*, de las cuales una de las más utilizadas es la Licencia Pública General GNU (GPL). El autor conserva los derechos de autor "copyright", y permite la redistribución y modificación bajo términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo los términos más restrictivos de la propia GNU GPL. Esto hace que no sea imposible crear un producto con partes no licenciadas GPL: el conjunto tiene que ser GPL [Marín, 2006].

1.3.2 Software gratis

El SW gratis es el que se regala, no exige intereses económicos al distribuirse; este concepto apenas se ha desarrollado ni ha adquirido posición y prestigio. Algo más generalizada es la modalidad conocida como freeware, la que sin tener una definición clara de aceptación generalizada, se usa comúnmente para paquetes distribuidos en binario (sin

código fuente). Estos paquetes no son SWL, por lo tanto no es correcto usar la palabra “freeware” para referirse al SWL. El freeware generalmente es SW propietario en su esencia o conduce de alguna manera a este.

1.3.3 Software de Código Abierto o de Fuente Abierta (Open Source)

La creciente confusión provocada por los modelos y términos originales asociados al SWL influyeron significativamente en la aparición del concepto Open Source (OS) ó Código Abierto (CA). Con el objetivo de contribuir a aclarar esa situación, ofrecer otro alcance y nuevas alternativas, se empezó a manejar el término OS, el que un tiempo después fue la base de un nuevo movimiento del SWL.

El nacimiento del Open Source está muy influenciado también por un acontecimiento importante, gran parte de las ideas que conforman este movimiento provienen del artículo de Eric Raymond publicado en el año 1997, titulado “La Catedral y el Bazar”, el que recoge las observaciones y análisis antropológicos que hacen funcionar al mundo OS.

En el artículo se establece el contraste entre los dos modelos de desarrollo que implantaban estilos diferentes. Uno era cerrado, el estilo catedral en el que de manera estricta se tenían algunas especificaciones para pequeños proyectos que se ejecutaban de modo jerárquico y muy autoritario y con periodos de publicación sumamente largos. El otro dramatizaba el mundo de Linux más de igual a igual como un mundo de intercambio, más estilo bazar, el cual tenía ciclos de publicaciones muy cortos y constante solicitud de feedback, de personas que estaban fuera del proyecto. Es decir era un proceso de crítica muy intenso. Este proceso de crítica masiva era una ventaja relevante, al punto que parecía que con el se obtenían buenos resultados, y lo más alarmante era que valía la pena cambiar las supuestas ventajas del desarrollo convencional cerrado por la de crítica masiva.

La razón por la cual entra Netscape en esta historia fue por ser la primera compañía que sacó a la luz el código fuente de un producto y la que obligó a Eric Raymond, Bruce Pernees y a otros colaboradores a tomar la decisión de formar la organización llamada hoy Open Source Initiative (OSI) .

La compañía pionera en compartir su código tenía un motivo muy fuerte para adoptar esta alternativa, pues la competencia con otras compañías era muy grande; aunque el papel fundamental en todo esto lo jugaba Microsoft, con su navegador Internet Explorer, el cual estaba distribuyendo de manera gratuita pero sin dejar que nadie tuviera el código fuente y sin pedir que las empresas colaborasen. Esta gratuidad era relativa, era una manera de asegurar la venta de todo lo demás, de estimular el uso de sus demás plataformas propietarias: SO, plataformas de desarrollo, paquetes como el Office, etc.

Después de un exhaustivo análisis desarrollado por los responsables de estos temas en Netscape, análisis que jugó un importante papel el artículo de Eric Raymond “La Catedral y el Bazar” el cual tuvo una influencia significativa en la decisión de Netscape de hacer público el código fuente, aunque no fuera esta la única influencia, se decidieron lanzar el SW.

Anunciando Netscape de esta forma el 23 de enero de 1998 la distribución gratuita de su navegador web, así como la publicación de su código, actividad que llamó la atención de muchos. Este fue un momento de inflexión para el SWL, porque una empresa importante en aquel instante como Netscape, hacía una apuesta firme por el movimiento y ponía a disposición de la comunidad un navegador de Internet en forma de SWL con las últimas tecnologías.

A raíz del suceso de Netscape se realizó una reunión con la intención de reaccionar ante el plan de esta compañía de liberar el código fuente de su navegador 'Netscape Navigator'. Además, Eric Raymond se percató del problema que existía con el SWL y la proclividad de las personas a asociar este término con la libertad de beneficios. Esta reunión sesionó el 3 de febrero de 1998 en Palo Alto, California donde se llegó a este nuevo concepto de Open Source, el cual procedía del proyecto Debían y se había definido por Bruce Perens.

¿Qué es Open Source?

En la película Revolution OS Bruce Perens responde a esta pregunta de la siguiente manera: *“Open Source es una forma para conseguir que la gente colabore con el SW sin verse envueltos en problemas de propiedad intelectual, evitando tener que negociar contratos cada*

vez que compran SW, con muchos abogados de por medio. En general solo queremos que el SW funcione y queremos ser capaces de tener personas que contribuyan con soluciones, etc. Así que se sacrifican algunos derechos de propiedad intelectual, y simplemente se permite que todo el mundo use ese SW.”

Y define el concepto de Código Abierto (Open Source) como sigue:

1. Libre redistribución: el software debe poder ser regalado o vendido libremente.
2. Código fuente: el código fuente debe estar incluido u obtenerse libremente.
3. Trabajos derivados: la redistribución de modificaciones debe estar permitida.
4. Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas solo como parches.
5. Sin discriminación de personas o grupos: nadie puede dejarse fuera.
6. Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.
7. Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa.
8. La licencia no debe ser específica de un producto: el programa no puede licenciarse solo como parte de una distribución mayor.
9. La licencia no debe restringir otro software: la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.
10. La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

La Iniciativa Código Abierto responde a dos objetivos fundamentales: el primero consiste en publicar esta definición por la cual nació el movimiento y el segundo era registrar como marca el término “Open Source”, pero como este vocablo no es descriptivo, no fue aceptado como marca registrada. Entonces para poder indicar que el SW que cumple con la Definición de Código Abierto con esta nueva licencia, se creó la fundación OSI como entidad con personalidad jurídica y se registró la etiqueta “OSI Certified” (Certificado por la OSI). Esta certificación se aplica al SW que se distribuye bajo la licencia que cumple con la Definición de Código Abierto.

1.3.4 Diferencias entre Software Libre y Fuente Abierta o Código Abierto

La FSF con su licencia GPL y OSI con OS, son dos movimientos progresistas que defienden al SWL bajo cualquier concepto. Las ideas de la OSI pueden desde cierta perspectiva considerarse más avanzadas, al menos en algunos aspectos medulares. La diferencia que se puede observar entre ambas, está dada fundamentalmente en sus modelos y procesos de desarrollo, o sea desde el punto de vista de la Ingeniería del Software y los modelos de negocio que estos propugnan.

Por otra parte la FSF brinda buenas posibilidades para la libertad del software pero tiene un modelo de desarrollo más lento, manteniéndose aferrada a sus primeras ideas, que ya tenían cerca de 14 años cuando apareció el Open Source, hechos que, como en todos los casos, siempre es necesario perfeccionar y modificar en alguna medida, actualizar ciertos conceptos y alcances. Indudablemente que la FSF ha ido adoptando nuevas versiones de la licencia GPL, alcanzando justamente la GPL V3 en el año 2007, e incorporando lo relativo a la documentación entre otros casos, todo lo que constituye un reconocimiento tácito de las necesidades que pone en práctica el movimiento Open Source representado por la OSI.

Una buena fuente para contribuir a ejemplificar estas diferencias puede encontrarse el artículo relacionado con el tema, “La Catedral y el Bazar” de Eric Raymond, en el que pueden apreciarse claramente algunos de los elementos que se ofrecieron anteriormente.

Los proyectos de SWL, como GNU y NetBSD, tienen una metodología fuertemente centralizadas, mediante el que unas pocas personas realizan el diseño e implementación del software. Las tareas que desempeñan estas personas, así como sus roles, están perfectamente definidos y alguien que quisiera entrar a formar parte del equipo de desarrollo necesitaría que se le asignara una tarea y un rol según las necesidades del proyecto. También, las entregas de este tipo de programas se encuentran espaciadas en el tiempo siguiendo una planificación bastante estricta. Esto supone tener pocas entregas del software y ciclos largos entre ellas, a la vez que constan de varias etapas [Pascual, 2007].

En este modelo, la solución a los errores y problemas de desarrollo son procesos más complejos y algo inflexibles. Generalmente toma meses de revisión exhaustiva para alcanzar la seguridad de que han sido eliminados en la medida más adecuada. Por eso se dan los intervalos tan largos entre cada versión que se libera, y la inevitable desmoralización cuando estas versiones, largamente esperadas, no resultan perfectas [Raymond, 1998].

Otro ejemplo a estudiar es el desarrollo creciente del núcleo Linux, llevado a cabo con una metodología que sigue un esquema donde no existe una máxima autoridad que controle rígidamente los procesos en ejecución ni que planifique estrictamente lo que ha de suceder. También los roles de los participantes pueden cambiar de manera continua (los vendedores pueden pasar a ser clientes) y sin indicación externa, además las versiones se liberan de una manera muy rápida. La descripción del proceso que ha hecho de Linux un éxito dentro del mundo del SWL, refleja que este es una sucesión de buenas maneras para aprovechar al máximo las posibilidades que ofrece la disponibilidad de código fuente y la interactividad mediante el uso de sistemas y herramientas colaborativas de la informática. Esta forma de desarrollo que aplicó Linus en la construcción del núcleo Linux es la que pone en práctica y trata de generalizar y perfeccionar la OSI.

En esta metodología, se asume que los errores son fenómenos relativamente evidentes o, pueden tornarse así cuando se exhiben a miles de entusiastas desarrolladores que colaboran a la par sobre cada una de las versiones. En consecuencia, se liberan con frecuencia para poder obtener una mayor cantidad de correcciones, logrando como efecto

colateral benéfico el perder menos tiempo, cuando aparece un eventual obstáculo. Por eso esta última iniciativa es mucho más acogida por el mundo empresarial brindándole más beneficios y mayor rapidez a las comunidades que desarrollan SWL [Raymond, 1998].

Open Source promueve una colaboración con nuevos enfoques, que motive a la gente mucho más que el modelo clásico de la FSF, apreciándose que está presente un elemento de ingeniería y de participación diferente que atrae a otras personas. El movimiento Open Source ha ganado mucho campo con su punto fuerte, el modelo de desarrollo de tipo peer-to-peer o red entre iguales.

Una preocupación importante de los fundadores de la OSI era encontrar una manera de promover las ideas en torno al SWL que motivara a las personas que anteriormente habían rechazado este movimiento. Les preocupaba que las intenciones anti-negocios de la FSF fueran las de mantuvieran el mundo al margen de apreciar mejor el potencial del SWL.

La definición de la OSI es más flexible en lo relativo a la mezcla de licencias, facilitando así que un software de código abierto pueda tener enlaces a productos propietarios. Las licencias de la OSI son más “permisivas” que las GPL, pudiéndose encontrar que un software bajo una licencia del tipo OS puede ser SWL, pero también puede llegar a ser semi-libre o incluso completamente no libre.

A continuación se ofrece un buen ejemplo de la mayor flexibilidad y alcance de las licencias OS. A finales de 1998 hubo una importante disputa en la comunidad de SWL, provocada por la llegada de dos sistemas de software, GNOME y KDE, cada uno de los cuales tiene como objetivo construir una interfaz de escritorio orientada a objetos. Por una parte, KDE utilizaba la biblioteca Qt de Tecnología Troll Tech (una pequeña empresa noruega), una parte del código que era propietario, pero muy estable y maduro. Por otra, los usuarios de GNOME decidieron utilizar la biblioteca GTK +, que es una biblioteca completamente libre, aunque no tan avanzado como Qt. Antes de la variante OSI la tecnología Troll Tech hubiese tenido que elegir entre usar la GPL y mantener su postura de propiedad. La brecha entre GNOME y KDE habría continuado. Sin embargo, con la llegada del Open Source, Troll fue capaz de cambiar su licencia [Raymond, 1999].

1.4 Algunos conceptos

1.4.1 Hardware

El HW, como es bien conocido, está integrado por los componentes físicos tangibles que integran a los sistemas de computadoras y sus redes. A los fines de este trabajo lo que interesa destacar es que, aun cuando le caracteriza un proceso previo de creación intelectual, tiene que seguir un proceso industrial tradicional hasta alcanzar la producción seriada. Ha sido normal su protección mediante patentes y secretos industriales. Es la forma en que nació el comercio de la informática, e indujo crecientemente la necesidad del desarrollo del SW, incluyendo su conocida crisis. Su precio ha bajado exponencialmente en el tiempo, al mismo tiempo que su potencia ha crecido en forma similar. Nada de ello es extraño al modo capitalista de producción, engendrado por la revolución científica conocida como revolución industrial.

1.4.2 Software

El SW es aquel conjunto de instrucciones cuya finalidad es interactuar con el hardware y realizar tareas específicas para el usuario (sistemas operativos, hojas de cálculo, gestores de bases de datos, procesadores de texto, navegadores de Internet, etc.). Su evolución como producto e industria ha obedecido a modelos, conceptos y normas muy controvertidos, por no responder al modelo clásico industrial de la revolución que dio origen a la sociedad capitalista. Lo anterior no hay dudas que obedece a la naturaleza de la creación intelectual, cuya industrialización es totalmente diferente a la de los productos industriales clásicos. Desde sus momentos más tempranos ha inducido fuerte y crecientemente diversas modalidades de desarrollo y producción en cooperación.

1.4.3 Sistema Operativo

Un Sistema Operativo es una plataforma software, compuesta por un conjunto de programas que actúan como interfaces entre el usuario de la computadora, su hardware (incluida la red) y las demás componentes del software. Su objetivo principal consiste en facilitar y optimizar el uso de los recursos del sistema, por lo que este puede verse como un administrador de los

recursos de hardware y el encargado de presentar al usuario una capa de abstracción de software que lo aísla de la complejidad del sistema en su conjunto. Generalmente suelen dividirse en los cuatro administradores fundamentales siguientes: de procesos, de dispositivos, de información y de memoria.

1.4.4 Software propietario, privativo o cerrado

Los usuarios de SW propietario no pueden acceder al código fuente del mismo, por lo tanto se ven privados del conocimiento real de las utilidades y funciones que el SW realiza. Están privados del derecho a compartirlo, modificarlo, adaptarlo o corregir sus errores, convirtiéndose así en clientes cautivos de las empresas que los desarrollan, comercializan y ofrecen determinados servicios técnicos. Estos usuarios o clientes pierden así sus derechos básicos como consumidores.

El SW propietario se caracteriza por privar de derechos a sus usuarios, aspecto semántico por el que también es correcto denominarlo privativo. Este tipo de software ha sido diseñado para sostener un modelo de negocio que busca maximizar las ganancias sobre la base del monopolio sobre el código fuente y el permiso legal para hacer cambios en los programas. Además se caracteriza por dividir a los usuarios y les dificulta colaborar con su trabajo y conocimientos en la mejora del software y en pro del bien común [Saravia, 2005].

Este concepto es objeto de gran controversia terminológica, en la que ha influido mucho un planteamiento de Stallman del año 2003. En la Wikipedia en español aparece un estudio relativamente exhaustivo de este problema, en el que se comprueba que en idioma español tanto el término propietario como el privativo tienen una parte del significado que explica bien el fenómeno pero otra que no. A los efectos de esta investigación se dio preferencia al término “propietario”, considerando recomendable su uso y generalización en el español especializado que se habla en Cuba.

1.4.5 Dominio público

Al dominio público pertenece el conjunto de bienes que en cada país cualquier persona tiene derecho a usar sin necesitar permisos ni pagar por su uso. En relación a bienes intangibles

como lo son los protegidos por la propiedad intelectual, el dominio público comprende todos los bienes que no están protegidos particularmente y forman un tesoro de información y recursos de libre disposición. La creatividad e innovación de una sociedad se basan en gran medida en el dominio público, el cual incluye la herencia de todas las generaciones anteriores. Todas las obras intelectuales, artísticas y otras que caen en categorías protegidas por propiedad intelectual necesitan de un dominio público para poder ser creadas.

El autor renuncia absolutamente a todos sus derechos, en favor del común, lo cual tiene que estar declarado explícitamente en el programa, ya que si no se especifica explícitamente, el programa es propietario y no se puede hacer nada con él sin una autorización por la persona natural o jurídica que detenta ese derecho. En este caso, y si además se proporcionan los fuentes, el programa es libre [Pascual, 2007].

Para la sociedad es imprescindible contar con un dominio público lo más extenso posible, dado que cualquiera puede valerse de esta fuente para crear nuevas ideas y obras basándose en el conocimiento acumulado. Es por ello que es natural para una sociedad intentar maximizar el dominio público que tiene a su alcance. La propiedad intelectual consiste en un conjunto de herramientas legales que permiten a los creadores intelectuales abstenerse de agregar sus obras directamente al dominio público, al menos por el plazo en años que se establece en cada país, aun cuando se hubiesen beneficiado indudablemente de la riqueza del dominio público mundial. Esto se hace invocando a que en el largo plazo la cantidad de innovaciones creativas que pasen al dominio público sean mayores que en una situación sin esta protección. En otras palabras, parece razonable abstenerse de tener ciertos bienes en el dominio público de inmediato y esperar un tiempo, porque de lo contrario es probable que esa obra no hubiese sido generada siquiera, mucho menos podría haber pasado al dominio público [Hardings, 2003]. Sin embargo, la joven historia del SWL parece cuestionar lo expresado por Hardings y muchos otros, hecho que también lo refrenda la no corta historia de la lucha por una cultura abierta y la necesidad del desarrollo humano.

Aspectos Legales del Software con énfasis en el Libre

2.1 Introducción

La ley de la propiedad intelectual para el segundo milenio, los esfuerzos de numerosas empresas por controlar Internet especialmente, el dominio de Microsoft sobre parte importante de los programas las computadoras modernas y la voluntad de tribunales y legisladores, son las principales fuentes de limitación institucional sobre la libertad para usar los recursos lógicos necesarios para la comunicación en red.

El universo de la información, su contenido y la cultura existente, disponibles en el dominio de las redes, presentan una tendencia sistemática en la ley y una contra-tendencia en la sociedad. Los derechos de propiedad intelectual se aplican a más usos, las marcas son más fuertes y agresivas y las patentes se han extendido a nuevos ámbitos, contando con más eficacia. Ante este escenario ha aparecido una fuerte corriente potenciada por el desarrollo de las redes, las herramientas colaborativas, las licencias Creative Commons y otras alternativas que cada vez sustentan con más fuerza la presencia de una revolución social que caracterizada por un modo colaborativo de producción del conocimiento como forma más acabada de producir este valor. En la medida en que el modelo de información y cultura existente se vaya ocupando por estas alternativas libres, y se vaya complementando con el modelo de las técnicas de licencias libres, el conflicto basado en la propiedad intelectual pudiera cesar.

2.2 Aspectos legales

Las ideas no son un producto de individuos aislados o de las corporaciones. Estas se fundamentan en gran medida en las tradiciones, la sabiduría colectiva y el entendimiento de grupos sociales. También pueden estar basadas en creaciones y fenómenos naturales, sometidas a procesos de desarrollo. Es por ello que las personas interesadas tienen derecho social a utilizarlas en beneficio del público, fundamentalmente si éstas son la clave de la prosperidad de la sociedad.

La aparición de la propiedad intelectual como figura del derecho devino del surgimiento del mundo industrializado para regular el movimiento de conocimiento y para equilibrar el uso de cualquier idea portadora de algún valor económico-comercial. Estos se instituyeron a partir de supuestos que sugieren que los autores de las ideas tienen derecho material a una compensación económica por su esfuerzo y un derecho moral a no ver sus ideas deformadas, además de reconocidas.

El concepto de propiedad intelectual tiene varias acepciones según el contexto y quién lo utiliza. En esta pueden destacarse los derechos de autor, las marcas, los secretos industriales y las patentes. Sin embargo en muchas tradiciones legales, entre ellas la hispana, se distingue entre la propiedad intelectual, que se refiere exclusivamente a los derechos de autor y la propiedad industrial, que abarca las figuras restantes.

En las sesiones siguientes se puntualizarán algunas de las categorías de la propiedad intelectual.

2.2.1 Derechos de autor

En términos jurídicos el derecho de autor es el que describe los derechos concedidos a los creadores por sus obras literarias, artísticas y científicas.

Una obra puede definirse como la expresión formal de una idea o sentimiento que se quiere comunicar al público. El tipo de obras que abarca el derecho de autor incluye: obras literarias, como novelas, poemas, obras de teatro, documentos de referencia, periódicos y programas informáticos; obras artísticas; obras científicas; bases de datos; películas, composiciones musicales y coreografías; obras arquitectónicas; publicidad, mapas y dibujos técnicos [Bain,2004].

El origen de los derechos de autor se remonta a la época medieval cuando las obras se comenzaron a explotar mediante la reproducción en soportes físicos por los copistas medievales y posterior a ello, se empezaron a difundir mediante la imprenta. La imprenta transformó las obras en objetos comerciales permitiendo que los editores se beneficiaran directamente, aunque con este avance apareció por primera vez la piratería intelectual,

provocando que los editores ejercieran una fuerte presión sobre los legisladores para obtener protección y asegurar sus rendimientos económicos. De esta manera el primer objetivo de los derechos de autor fue regular y proteger a los editores contra el plagio. Desde entonces la historia del derecho de autor se caracteriza por la concesión de privilegios a determinados impresores, aunque además en algún momento de la época se utilizó por parte de los gobernantes como sistema para controlar y censurar las obras publicadas.

En ocasiones se maneja el término derechos de autor y en otro copyright. Conviene aclarar que los términos “**derechos de autor**” y “**copyright**”, no son completamente equivalentes. El Copyright, de orientación mercantil, es predominante en la concepción jurídica angloamericana y que por lo general, comprende la parte patrimonial de los derechos de autor. En cambio, el derecho de autor de carácter individualista nació en Francia, en la época de la Revolución Francesa, y es el fundamento de las legislaciones de tradición jurídica europea o latina. Estas dos vertientes del derecho de autor se fundamentan en la existencia de dos concepciones jurídicas diferentes.

- El término “*author’s right*” –derecho de autor en español, está basado en la idea nacida en la Europa continental, de que una obra está estrechamente relacionada con su creador y donde el acento se pone en el autor. La base de los derechos de autor es, en contraposición al copyright, el reconocimiento de unos derechos morales irrenunciables e intransmisibles a todo autor [Bain, 2004].
- El término “*copyright*” –derecho de copia en español, proviene de la tradición anglosajona y se basa en la idea de que lo más importante no es el autor, sino la obra y los derechos de propiedad que los autores tienen sobre sus creaciones, con los que se puede comerciar sobre la base de los principios económicos [Bain, 2004].

El **derecho de autor**, tiene dos componentes potencialmente significativos:

Los **derechos morales** son personalísimos, intransferibles e imprescriptibles, como la paternidad e integridad de la obra, arrepentimiento o retracto, estos indican la tutela de los intereses intelectuales del autor con respecto a su obra.

- **Derecho de divulgación:** Este derecho le permite dar a conocer su obra o mantenerla en la esfera privada.
- **Derecho al reconocimiento de la paternidad:** Se da a conocer en cada caso quién es el autor de la obra.
- **Derecho al respeto y a la integridad:** La obra no puede ser modificada, mutilada o desnaturalizada, sin el consentimiento del autor, pero éste sí puede hacerlo cuando lo entienda.
- **Derecho de retracto o arrepentimiento:** Si él autor determina que la obra debe ser retirada del ámbito público así debe hacerse aunque por ello deba pagar por el incumplimiento de determinado contrato.

Los **derechos patrimoniales** de contenido económico le permiten al titular o a sus derechohabientes, beneficiarse con la explotación, reproducción, comunicación y difusión de la obra literaria o artística, por cualquier medio o procedimiento, es decir son los relacionados con la explotación de la obra y la consiguiente obtención de beneficios económicos

- **Derecho a la reproducción:** Posibilidad de explotar la obra mediante su fijación material primero y la obtención luego de copias para la distribución. Este es uno de los derechos patrimoniales fundamentales. La reproducción no implica sólo la venta de copias o ejemplares sino también el alquiler o la entrega gratuita del producto.
- **Derecho a la comunicación pública:** Dar a conocer la obra públicamente por medios que no consisten en la distribución de ejemplares.
- **Derecho a la transformación:** El autor u otra persona previamente autorizada por el creador modifican la obra originaria haciendo surgir una obra derivada. Por ejemplo: traducciones, adaptaciones, actualizaciones, entre otras [Soler, 2006].

2.2.2 Patente

Las patentes son una autorización legal formulada por un gobierno que permite al inventor excluir a otras personas de fabricar, utilizar o vender un invento, declarado como propio, durante el plazo de vigencia con el que se le otorgue. Para recibir esta protección o titularidad, un invento debe mostrar de manera clara que se puede patentar (proceso, maquinaria, artículo manufacturado), que es original y novedoso, que no es obvio y que es útil.

Las patentes otorgan un derecho monopólico de 17 a 25 años, posibilitando también que por un determinado valor monetario se permita el acceso a la invención y a su reproducción, bajo condiciones contractuales rigurosas. Este mecanismo promueve la investigación privada, sin valor monetario para el colaborador y sin perder el resultado. El individuo que posee una patente puede decidir si permite a otros hacer uso de ella y el precio que debe pagar por la licencia. Los defensores del sistema de patentes argumentan que este suscita o estimula la innovación, pero las fuertes restricciones que promueven en ocasiones lo que hacen es dificultarla en medida importante.

2.2.3 Marca

Las marcas son nombres o símbolos protegidos por registro legal que identifica el producto o servicio de un fabricante o comerciante y lo distingue de otros productos y servicios. Los iconos, los nombres de compañías, los nombres de marcas, las envolturas, todos estos pueden tener protección de marca comercial. Los dueños de las marcas registradas o comerciales tienen el derecho a evitar que otros las utilicen, incluyendo otras que puedan resultar muy similares o que den lugar a confusiones, pero no pueden evitar que otros fabriquen o vendan los mismos productos con otra marca diferente que no se confunda.

2.2.4 Secreto comercial

Esta vía de protección está amparada por las leyes de propiedad industrial y es un mecanismo mayormente usado por las empresas para rentar sus inversiones. Esta estrategia se aplica para no tener que revelar información que la empresa no quiera exponer

por las razones que ella decida. Hay un caso en que la empresa si tiene que exponer los secretos porque necesita de una aprobación gubernamental pero para ello el estado se compromete a no hacer público el secreto de la institución.

La industria del SW propietario es un ejemplo claro de aplicación de secreto comercial, pues su fuerte es la comercialización de programas compilados que no dan acceso al código fuente, que es la característica fundamental que le permite a personas que desarrollan SW hacer mejoras o modificaciones para su uso.

La protección del secreto comercial es estricta, limitando a la humanidad el acceso a conocimientos útiles por tiempo indeterminado o excesivamente largo. Por causa de esta restricción algunas legislaciones, permiten la ingeniería inversa para desarrollar productos sustitutos, aunque la presión realizada por las empresas ha conseguido que en muchos países ésta sea una actividad prohibida y en otros sólo esté permitida en aras de la compatibilidad.

2.2 Licencias de software

La licencia de SW es el instrumento legal a través del cual el autor, titular o proveedor (licenciante) establece las condiciones y los requisitos generales bajo los que se otorga el SW al usuario (licenciataria). Las licencias de SWL y de SW propietario se diferencian entre sí en la amplitud y alcance de las libertades que brindan a los usuarios [Soler, 2006].

El SWL como movimiento progresista defiende la idea de que exista completa libertad para el usuario a la hora de utilizar el SW, y la mejor manera para lograr esto es que su autor lo libere, poniéndolo desde el principio a disposición del dominio público.

Aquí ocurre un problema muy grave, si el autor libera el SW y lo pone a disposición del dominio público, que seria lo ideal para el movimiento del SWL, cualquier persona podría entonces realizar acciones con el SW cuando este se encuentre bajo dominio público puede incluso convertirlo en SW propietario, cerrando el acceso al código fuente y liberándolo bajo una licencia restrictiva que eliminaría las libertades que proclama el SWL (recuérdese el ejemplo del BASIC y Microsoft).

Ante esa dificultad la FSF creó un ingenioso mecanismo que basado en el CR y utilizando el dominio público, añade la cláusula Copyleft (CL) mediante la que asegura que el SWL, puesto en dicho dominio, jurídicamente no sea posible convertirlo en propietario .

Por la importancia que tiene el mecanismo del CL, de manera breve se expondrá en qué consiste este, contribuyendo a esclarecer los conceptos y mecanismos relacionados.

Para proteger un programa con el CL (hacerlo CL), primero se reservan los derechos, registrándolo bajo CR o DA y luego se le agregan las cláusulas de distribución. Este instrumento legal le da a cualquiera el derecho de usar, modificar y redistribuir el código fuente del programa o de cualquier programa derivado de éste, siempre que no se alteren los términos de distribución.

La característica fundamental del CL, consiste en garantizar que todo trabajo derivado de una creación determinada, diseminada bajo CL, deba continuar su existencia bajo esa protección, de manera que las libertades que esto garantiza no puedan bloquearse por terceros.

Este concepto de distribución se ha diseminando en la comunidad . El CL en la actualidad se extiende más allá del software, apareciendo otras modalidades en dependencia del campo de la creación intelectual de que se trate, por ello es posible encontrar licencias con CL para proteger diversos tipos de obras.

Las licencias y sus pautas caracterizan y distinguen a los dos modelos generales de distribución de SW que hoy coexisten (SW propietario y el SWL). Esto las convierte en herramientas estratégicas, por lo que es de mucha importancia abordar este tema como uno de los resultados de este trabajo.

2.4.1 Licencias libres

Cuando se habla en general de licencias libres se hace referencia a las más socializadoras, las que en sus postulados brindan ciertos beneficios a los entes sociales que las utilizan , haciéndolas más de la sociedad, es decir más del público que desee utilizarlas. Las

cualidades que ofrece este tipo de licencias posibilitan su uso efectivo a mayor número de personas; tienen grandes facilidades porque le permiten a los usuarios disponer del código fuente del SW, usarlo, modificarlo y redistribuirlo, con o sin CL, generalmente sin costo y sin garantías. Son licencias que brindan un gran número de facilidades.

Este tipo de licencia tiene tres modelos fundamentales. Las diferencias de estos modelos radican en cómo los propietarios de los derechos conceden parte de ellos a los usuarios y bajo qué condiciones. A continuación se explicará en qué consisten estos modelos.

El primer grupo está integrado por las licencias del tipo CL. Esta cláusula es la que garantiza que cualquier versión que se realice derivada de un SW con CL herede las mismas condiciones que proclama el programa original. En este tipo de licencias se garantizan también las condiciones necesarias para que un SW sea libre, es decir conceden a los usuarios los derechos de copia, distribución y modificación del programa bajo las condiciones que definen al SWL, aparte de que certifica que la libertad de este persista cuando se realizan versiones derivadas del mismo.

En un segundo grupo están las licencias permisivas. Este tipo de licencias siguen las condiciones que proclama el SWL pero con una diferencia, el usuario que trabaje bajo este tipo de licenciamiento no está obligado a publicar las mejoras que realice sobre el código, por lo que entonces podría convertir este SW derivado en un producto propietario, aun cuando tuvo su génesis en el SWL. Si se miran desde un punto de vista permisivo, este tipo de licencias da más libertad a los creadores de SW ya que no imponen restricciones que impidan crear SW propietario partiendo de un proyecto libre; estas licencias abren la posibilidad de que se le quite libertad a los usuarios, al no garantizan que futuras versiones de este SW sean libres.

El tercer grupo o modelo de licencias libres depende en gran medida del uso que se le vaya a dar al programa y las libertades y obligaciones que se deseen adquirir, pues se fundamenta en que el autor cede su SW bajo dos licencias diferentes. Esto destaca dos vertientes, la que puede ofrecer una versión libre de su programa y otra con mejores condiciones para las necesidades de alguien que quiera trabajar más específicamente fuera

del modelo del SWL. Este modelo es muy usado en el entorno empresarial pues brinda la facilidad de poder ofrecer los productos de forma libre y no libre, según las necesidades de cada quien, constituyen así una de las modalidades incorporadas por este movimiento .

A continuación se enuncian y describen brevemente las licencias libres más conocidas en el entorno del SWL.

2.4.1.1 Licencia Pública General de GNU (GNU-GPL)

La GNU-GPL es la licencia más popular y conocida del mundo del SWL. Su nombre es Licencia Pública General del proyecto GNU, muy conocida por GPL, su acrónimo en inglés. En sus inicios fue creada como la licencia para la distribución y protección de todo el SW que generara la FSF, movimiento que le dio vida. Como relevante debe señalarse que ha resultado la licencia más usada en la actualidad, incluyendo su uso por el proyecto que dio origen al núcleo Linux, pero no la única en las distribuciones de sistemas que lo incluyen.

Esta licencia es atractiva desde el punto de vista legal porque hace un uso original y creativo de la legislación del CR o del DA, ya que al utilizarlas estrictamente, llega a una solución totalmente contraria a ellas, hecho por el que también se designa como una licencia CL.

La GPL facilita la redistribución de ambos códigos, el código fuente y el ejecutable , aunque en el caso de que se redistribuya el binario, obliga a que también se permita y garantice el acceso al código fuente. Permite realizar modificaciones sin ningún tipo de restricciones, teniendo en cuenta siempre que solo se puede integrar código licenciado bajo GPL con otro código que se encuentre bajo una licencia similar o compatible . Esta característica hace que a esta licencia se le atribuya una consecuencia recursiva por garantizar la libertad del SW al cual licencie, asegurando también que el código que sea publicado con este estado nunca pueda cambiar de condición.

Esta licencia asegura la libertad del código todo el tiempo, un programa publicado y licenciado bajo sus condiciones nunca podrá convertirse en propietario sin que ello implique una trasgresión jurídica de la licencia. Ni ese programa ni sus modificaciones pueden ser publicados con una licencia diferente a la propia GPL. Esto maximiza las libertades de los

usuarios; otro enfoque interesante es el que hacen muchos autores de SW, al considerar que esta licencia es muy útil por obligar a sus competidores a publicar sus modificaciones en caso que se redistribuya el SW.

2.4.1.2 La Licencia Pública General Menor de GNU (GNU LGPL)

Licencia Pública General Menor (Lesser) de GNU o GNU-LGPL. Con anterioridad se conocía como la licencia para Bibliotecas (Libraries) de las GPL; el nombre fue cambiado hace algún tiempo. Esta licencia es muy similar a la GPL en condiciones y restricciones, pero define otra alternativa, mediante la que permite al SWL vincularse con el propietario.

El uso de la LGPL solo se recomienda para casos específicos, por ejemplo cuando no hay otra alternativa y las librerías libres no resuelven el problema, entonces es cuando se recomienda su uso. Cuando la librería tiene funcionalidades extraordinarias respecto a las de su tipo y ofrece ventajas por sobre las demás, es preferible usar la GPL y no la LGPL. De esta forma se incentiva a los desarrolladores que quieran usar la librería a liberar también su código.

2.4.1.3 Licencia BSD

La licencia BSD es considerada menos restrictiva que la GPL porque no incluye la cláusula CL, dando así la posibilidad de crear versiones modificadas o derivadas de SWL que puedan convertirse con relativa facilidad en productos no libres o propietarios. Su fundamento parte de que si el SW es libre no debe imponer restricción alguna en su distribución, aunque ello implique que alguien lo use para su propio beneficio y no comparta el código fuente.

La FSF no la considera como una licencia libre porque no garantiza la continuidad de las libertades que debe caracterizar a toda distribución derivada del SWL. También existe otro inconveniente para la FSF con la BSD, relativo a su cláusula de publicidad (conocida como "4-clause"), mediante la que obligaba a hacer mención a la Universidad de Berkeley. En el año 1999 esta cláusula fue eliminada, momento a partir del que se fragmentó en dos tipos: las que incluyen la "4-clause" y las que no. Por esta razón muchas veces se encuentra SW

bajo “BSD-old/4-Clause BSD” y “BSD-new/BSD Revised”. Hoy en día muchos proyectos grandes utilizan licencias del tipo BSD o inspiradas en su filosofía.

2.4.1.4 Licencia Pública de Mozilla (MPL)

La Licencia Pública de la Fundación Mozilla (MPL) cumple plenamente la definición de SWL de código abierto de la Open Source Initiative (OSI), así como con las cuatro libertades enunciadas por la FSF. También deja abierto el paso a una posible reutilización de forma comercial no libre del SW licenciado mediante ella, si quienes lo desarrollan lo deciden así; esto último pudiera ocurrir sin limitar la reutilización del código y sin tener que usar de nuevo la misma licencia.

Sin embargo la función principal de la MPL es ser utilizada como licencia de control para el navegador Mozilla (actual Firefox) y el SW relacionado con este. También puede decirse que se usa frecuentemente por desarrolladores que quieren hacer libre su código fuente.

En su aparición marcó un hecho que resultó muy significativo y llamó mucho la atención de las comunidades del SWL y del propietario, por tratarse de la primera vez que una empresa muy conocida, la Netscape Navigator, decidió distribuir un programa bajo su propia licencia libre. Una de las interpretaciones de este hecho considera que fue una alternativa pensada para encontrar una vía de introducirse en el mercado dominado casi monopólicamente por Microsoft con su navegador Internet Explorer.

En cuanto a su compatibilidad con la GPL existen algunas limitaciones que no le permiten relacionarse de manera legal de forma directa, aunque la sección 13 de la versión 1.1 de la MPL posibilita que un programa o partes de él brinden la opción entre ella y otras licencias. Esto deja bien claro que si una parte de un programa consiente la GPL o cualquier otra licencia compatible con ella como alternativa, esa parte del programa se convierte en compatible con la GPL.

2.4.1.5 Licencia X11

Esta licencia en ocasiones es llamada X11 o licencia del MIT (*Massachussets Institute of Technology*) aunque este término no es el más recomendable, toda vez que esa institución ha utilizado diversas licencias para su SW. Es de SWL simple y permisiva, sin CL, pero compatible con la GPL.

Se usa frecuentemente en la distribución del sistema de ventanas más ampliamente difundido en el mundo UNIX, y también en entornos GNU/Linux como es el caso del X Window. Guarda cierta similitud con la BSD, permitiéndoles a los usuarios que la utilizan poder distribuir, usar y modificar SW prácticamente sin restricciones.

2.4.1.6 Licencia de la Fundación Apache

La licencia Apache es muy parecida a las BSD. No tiene compatibilidad alguna con la GPL. Permite hacer lo que se desee con el código fuente, incluyendo el uso de productos propietarios, siempre que se les reconozca su trabajo. Sus cláusulas estipulan que productos derivados de un código Apache original no pueden llevar como nombre Apache. Esta cláusula fue creada para evitar que cualquier versión modificada por otros de las distribuciones de esa organización se pudiera ofrecer a los interesados como una versión original oficial.

De esta licencia existen tres tipos, siendo la 2.0 la más utilizada a la vez que considerada como una licencia de SWL. Las otras dos versiones carecen de CL, característica clave para no mantener la garantía de que resulte realmente SWL.

2.4.1.7 Licencias de documentación libre o FDL

Si importante es liberar el SW también lo es liberar la documentación que le acompaña, trátase de manuales, documentación del código y todo lo que sobre bases razonables resulte necesario e importante para usar/modificar el programa o sus derivaciones. Es por estas razones elementales que aparecen las licencias para la documentación, en total coherencia con asegurar las libertades que proclaman los movimientos más avanzados del SWL.

La GNU FDL (*GNU Free Documentation License*) fundamentalmente permite que la obra pueda ser copiada, modificada y redistribuida. Esta licencia no hace especificaciones sobre el uso comercial y es del tipo CL. Admite precisar secciones invariantes dentro del texto, aunque esto puede ocasionar serios problemas al obligar así a que se conserven partes que pudieran resultar inconvenientes para ciertos usos.

La publicación comercial de un libro casi siempre implicaba un texto no libre. Pero, de la misma manera en que el SW comercial libre ha ido aumentando desde hace algún tiempo, la documentación comercial libre está también empezando a despegar. Algunas de las más importantes editoriales comerciales de documentación sobre el Sistema GNU/Linux y el SWL en general, están dispuestas ahora a publicar libros bajo la GNU FDL, y a pagar a los autores de la forma habitual, si los autores estuvieran interesados y decididos a ello.

Un aspecto muy importante para el uso de esta licencia es el conocimiento de su existencia, lamentablemente poco generalizado, incluyendo su funcionamiento y utilidad. Por lo general asociado al poco conocimiento sobre el SWL está o es mucho menor el relacionado con esta licencia FDL o de la documentación libre. Su divulgación por todos los medios constituye un factor importante.

También debe señalarse que junto a la FDL han ido apareciendo otras variantes de licencias sobre documentación y contenidos libres o de diversas modalidades libres, entre las que pueden mencionarse la Creative Commons como una de las más modernas, a lo que se dedica alguna atención más adelante en este trabajo.

2.4.2 Licencias propietarias

Las licencias propietarias constituyen el instrumento jurídico mediante el cual el propietario o licenciataria ejerce un control extremo sobre el producto de SW que distribuye comercialmente en diversas modalidades. En su evolución han devenido en las más restrictivas y aseguradoras del secreto y la exclusividad del derecho de la persona natural o jurídica a la que pertenece o que se le adjudicó el CR o el DA.

Hoy el caso extremo se encuentra en la muy conocida EULA, por sus siglas en inglés (End User License Agreement - Acuerdo de Licencia de Usuario Final), con su principal exponente en las versiones utilizadas por Microsoft. Sus cláusulas y términos aprovechan al máximo las posibilidades de beneficio privado que provee la base jurídica general del CR de los Estados Unidos o del país que se trate. Señala sin ambigüedad que su incumplimiento constituye delito criminal³ ante la jurisdicción norteamericana, amenazando con dar cuenta al FBI. En muchos casos sus términos contravienen normas jurídicas de algunos de los estados federados o de otros países, por lo que también suelen consignar que su alcance es pleno en tanto no contravenga las referidas legislaciones locales, en evitación de los conflictos que en no pocas ocasiones han tenido que enfrentar.

Otra característica especial que las distingue, como medida para conservar al máximo el secreto sobre el know how del programa que distribuyen, es el no suministro del código fuente (source code), entregando solo el código binario ejecutable dentro de determinado entorno o plataforma de SW, como si lo primero fuese poco. Así crean un mercado casi cautivo, obtienen ganancias enormes a partir de un producto totalmente masivo, que generalmente se vuelve obsoleto a intervalos no mayores de 3-5 años por las necesarias nuevas versiones. Junto a ello no ofrecen garantía alguna por el funcionamiento del SW y sus naturales defectos o errores, o por los perjuicios que este pudiera inducir, instando al adquirente a que lo acepte bajo esos términos, condición que de hecho acepta, con la clara precisión de que **ese SW “es como es”** y que el proveedor no ofrece garantías ni admite reclamaciones de tipo alguno, incluyendo su derecho a modificar el producto sin aviso previo.

Pudiera decirse que casi es imposible establecer mayores y más diversas restricciones que las alcanzadas hasta el momento. No solo privan al usuario del acceso al código fuente, sino también del derecho a copiar lo que recibe, a hacer modificaciones, re-distribuirlo, realizar la

³ En inglés la palabra *crime* tiene al menos los significados de delito y crimen. En el caso de las transgresiones al derecho administrativo, como es el incumplimiento de obligaciones contractuales o violación de normas jurídicas asociadas al derecho de propiedad, en ese idioma literalmente se habla de “delito criminal”. En Cuba se trataría en primera instancia de una violación dentro del marco del derecho administrativo, con independencia de que este pueda llegar al penal, en dependencia de las circunstancias.

descompilación u otro tipo de ingeniería inversa; el producto de SW bajo licencia propietaria no pertenece a quien lo compra, entre otras particularidades.

Para extremar su alcance y lograr un mayor dominio del mercado, también disponen de y usan variantes diversas, dependiendo del tipo de cliente y su potencial en el mercado, maximizando increíblemente la cantidad de copias que venden y la red de distribución. Entre ellas pueden mencionarse las llamadas licencias de máquina o computadora individual, las de uso concurrente, las corporativas y las OEM (*Original Equipment Manufacturer*), entre otras posibles.

Las licencias de máquina establecen que el programa solo puede utilizarse en una sola computadora por una persona determinada. La variante de uso individual establece que el SW solo puede utilizarse por un individuo específico. Las de uso concurrente posibilitan que varias personas puedan conectarse simultáneamente a un programa determinado, estableciendo un precio creciente, que varía en dependencia de la cantidad de usuarios para la que se licenció, con independencia de que el precio unitario disminuya en una razón mucho menos que proporcional. Las corporativas se caracterizan por autorizar el uso de un número determinado de copias dentro de los marcos de la organización que adquiere el producto en esa modalidad, partiendo de altos precios iniciales y la relativa disminución del costo unitario en la medida que la cantidad de copias contratadas aumenta.

La opción que seleccione el interesado en el producto de SW queda rigurosamente descrita en el documento que enmarca la transacción entre el cliente y el proveedor, especificando los pocos derechos y las excesivas obligaciones del usuario, dejando definidas con precisión las condiciones bajo las cuales obtuvo el producto.

Insistiendo un poquito más sobre la protección del secreto, el factor clave lo constituye la no entrega del código fuente (*source code*) y su documentación, cuyo conocimiento es privativo de su creador y del propietario del CR. El código fuente se mantiene en secreto al menos de dos maneras: la primera o más directa, consiste en no entregarlo junto con el programa; la otra, menos directa pero jurídicamente segura, es protegiéndolo como secreto industrial o empresarial, impidiendo acciones como la descompilación o la ingeniería inversa; estas

últimas prohibiciones se establecen también en el caso de no mediar la protección como secreto industrial, toda vez que en la EULA claramente se consigna la proscripción de la descompilación y la ingeniería inversa.

Cuando estas licencias consignan “el uso del programa” como derecho adquirido, dejan claro que como uso se entiende su ejecución y la obtención de los resultados que este brinda. El portador del DA o del CR no permite la distribución por constituir ello uno de sus derechos reservados, garantizando así una alternativa adicional para un mayor ingreso a su capital, controlando que la distribución solo se efectúe bajo sus reglas de participación en el beneficio monetario-financiero, junto con su consentimiento o intervención. También realizan distinciones entre el uso profesional y el doméstico, lo que influye de manera importante en el precio de la licencia y el potencial de venta de copias.

Por su casi increíble alcance restrictivo, merece la pena examinar un poquito más lo relativo al uso del SW adquirido bajo esa licencia. En la variante para una computadora, establece que este solo podrá ser usado por aquella persona que adquirió la licencia y únicamente en “una computadora”, con la sola “excepción” de permitir a esta su uso también en una computadora portátil; ante otra necesidad se tendrá que adquirir otra licencia o “cometer un delito”. En esta variante suele especificarse que para poder usar el programa en otra computadora diferente a la original en que se instaló, es obligatorio desinstalarlo previamente.

En lo relativo a los límites para transferencias a terceros, estas licencias suelen admitir otra rara excepción, mediante la que se permite al “adquirente directo” que pueda transferirlo en una sola ocasión a otro interesado, quien a su vez no tendrá ese derecho, asegurándose así jurídicamente que se pueda transferir a un tercero por una sola vez. Esto se complica un poco cuando esa licencia se adquiere como acompañamiento del hardware, acto en el que el comprador queda obligado a aceptar las condiciones de esa licencia, o quedar privado del derecho al uso del correspondiente SW.

Un caso muy singular es el correspondiente a las licencias propietarias OEM (*Original Equipment Manufacturer* - Fabricante Originales de Equipos – productores y distribuidores

del hardware con la opción de productos de SW instalados). Esta modalidad le permitió a Microsoft una de sus fuentes principales para dominar el mercado de los SO y otros paquetes y productos para PCs y redes. Aprovechando el estándar PC de la IBM y la posibilidad de sus clones, Microsoft logró una acción estratégica al producir el SO para las PC bajo licencia propietaria, de manera que cada fabricante que quiera distribuirlo con ese software, tiene que firmar contrato con MS en el que se estipula el pago que tiene que realizar por cada equipo vendido en dependencia de la configuración de su SW con el que lo realice. Por supuesto que los fabricantes y distribuidores de computadoras cargan ese gasto con MS al costo de producción de sus configuraciones, sobre la base del que forman su precio asegurando el margen de ganancia y utilidades que les resulte posible. Esta fórmula es la que en gran medida ha permitido a MS dominar el mercado de sistemas operativos, de manera que no menos del 80% de las computadoras del mundo operan bajo esa plataforma.

Las licencias de SW OEM se adquieren en el acto de la compra de una computadora, con el SW legalmente instalado con anterioridad. Quienes han adquirido una computadora nueva con Microsoft Windows o Microsoft Office ya instalado, han adquirido este tipo de licencias. Estas rigen sobre el derecho de distribuir SW Microsoft OEM e incluyen distintas formas de distribuir SW y soporte técnico. Proviene de un contrato entre el integrador OEM y Microsoft, no entre el usuario final y Microsoft o el proveedor OEM de SW propietario de que se trate.

Generalmente las licencias propietarias de SW OEM se utilizan por las empresas que producen o adquieren dispositivos del hardware al por mayor para ensamblar computadoras o equipos de forma personalizada, que presentan con su propio nombre. Los productos OEM pueden ser tanto hardware como SW. Los de hardware son aquellos que están carentes de gabinetes y chasis, accesorios como cables, tornillos o no incluyen SW. Por otra parte los de SW sólo se venden en combinación con algún tipo de hardware.

2.4.3 Compatibilidad de licencias

Entre las licencias libres y propietarias existen grandes divergencias. En muchos casos son incompatibles e inconciliables. También existen incompatibilidades y divergencias entre las licencias del grupo de las libres, así como entre las modalidades propietarias.

Las licencias propietarias tienen muchos aspectos en común, entre ellos el de adaptarse a las leyes de cada país, en lo que existe gran dispersión. En los propios Estados Unidos donde la protección al SW con el CR es muy fuerte, existe la enmienda del *Fair Use* como una conquista de la lucha contra la misma, proporcionando excepciones cuando se necesita para la enseñanza, investigación y divulgación, en cuyo caso el uso sin licencia o algunas transgresiones a estas no se consideran delito, permitiéndose legalmente dicha acción.

En el mundo del SWL existen, como se muestra en otras partes de este trabajo, diversidad de comunidades de desarrollo de SW, fundaciones e instituciones con sus principios, definiciones, lineamientos, modalidades de retribución y estimulación, modelos de negocios y comercio, así como los necesarios contratos sociales y diversidad de licencias con sus vías y formas de licenciamiento. La condición misma de la existencia de esa diversidad es necesaria y suficiente para que existan no pocas diferencias entre todos esos factores y conceptos.

Sin embargo muchas veces cuando se habla de SWL se tiende a definirlo o caracterizarlo como un fenómeno único, en el que nada o poco falta después del Manifiesto GNU y de sus ulteriores adecuaciones y precisiones, idealizándolo un tanto, donde no faltan las interpretaciones personales, a veces algo románticas, incluidos no pocos casos que supuestamente conocen profundamente este campo y sin prestar atención a toda la evolución y experiencias desarrolladas y acumuladas en los últimos 25 años, tienden a decir que el manifiesto GNU, los lineamientos GNU, las licencias GPL son siempre la mejor opción.

No obstante, los análisis y conclusiones parciales que se han expuesto muestran sin lugar a dudas la existencia de una gran variedad de incompatibilidades, contradicciones e

inconsistencias en materia de enfoques y modos de organizar e incentivar el desarrollo del SWL, su conversión en producto comercial, difusión, etc. La diversidad de concepciones y mecanismos para el aseguramiento jurídico que dimana de las diferentes licencias para asegurar y viabilizar la concesión del derecho a su uso y demás necesidades sistémicas de esta relativamente nueva rama, productora de conocimientos, que se ha situado entre las cinco más importantes de la economía mundial, cuyas reglas poco o nada tienen que ver con los modelos de la producción industrial de la sociedad capitalista, hacen pensar que no es nada sencillo ni común la comprensión, modelación y perfeccionamiento de esta rama productiva, que por demás tiene profundas implicaciones filosóficas e ideológicas.

Simplificando lo máximo posible la complejidad del problema antes planteado, es de suma importancia establecer y conocer con claridad la relación que existe, los objetivos que persiguen y las posibilidades que ofrecen las diversas licencias para el SWL, desde las menos hasta las más permisivas. Un aspecto clave a tener claro son sus incompatibilidades, así como los criterios que permiten decidir la mejor opción a seguir en dependencia de los objetivos concretos del proyecto que se pretenda desarrollar.

El no conocer con claridad los elementos anteriores por parte de los integrantes del equipo de desarrollo del proyecto es algo inadmisibles, por cuanto es equivalente y quizás mucho más grave que el desconocimiento de ciertas características y posibilidades técnicas de un programa o componente de sistema a incorporar en el producto final. En este último caso pudiera ser cuestionable la calidad del producto a obtener, pero el desconocimiento de los otros elementos es equivalente a no conocer precios y costos de los elementos a incorporar, o algo peor, armar un proyecto que jurídicamente no es viable por las incompatibilidades y violaciones que implicarían esas acciones, sin el fundamento indispensable.

La compatibilidad entre licencias debe tenerse en cuenta al menos dentro del grupo genérico al que pertenecen: el de las propietarias y el de las del SWL. Por ejemplo, dentro de las propietarias pueden aparecer dificultades en dependencia del país en que se adquieran y al que se destine el producto final. Las incompatibilidades entre las licencias del grupo del SWL también pueden llevar con facilidad a proyectos inviables desde el punto de vista jurídico, así

como en dependencia del modelo o estrategia económica y comercial al que este debe corresponder. Las incompatibilidades entre licencias de estos dos grupos contrarios son enormes, lo que en la mayoría de los casos imposibilitan la factibilidad del proyecto que inconscientemente las mezcle; tampoco quiere ello decir que no existan casos viables, pero para identificarlo y lograrlo es indispensable conocer bien las restricciones y las posibilidades.

Existen algunas licencias consideradas libres que de cierta forma presentan incompatibilidad con la GPL. Universalmente se tiende a aceptar que esta última es la que más garantiza la definición GNU de libertad del software, lo que no necesariamente significa que sea la única ni la que mejor estimula el desarrollo y distribución del mismo. La licencia Apache presenta incompatibilidad con la GPL al exigir que se mencione explícitamente su nombre en los materiales de divulgación y promoción del nuevo producto que utiliza su código, así como en cada uno de los documentos titulares de derecho. Las cláusulas de la GPL no permiten cumplir esas exigencias de la Apache. Esto no implica que no se puedan usar simultáneamente o integrar programas de ambas licencias, pero en ese caso es imposible por ilegalidad la distribución del SW derivado, al no poderse cumplir simultáneamente los requisitos de redistribución de ambas.

En este último ejemplo habría que tener en consideración el objetivo final de un desarrollo que mezcle productos GPL y Apache. Si se trata de una aplicación institucional que no requiera redistribución, el proyecto es viable desde el punto de vista jurídico; en caso contrario no lo será. En el anexo 2 se ofrece una tabla de las compatibilidades entre las diversas licencias de SWL, partiendo de considerar a la GPL como la de mayor prioridad, lo que no significa que sea el único ángulo de análisis a tener en cuenta. Es importante y útil que ese análisis se realice desde la perspectiva y requerimientos de la licencia que realmente más convenga a la estrategia y objetivos concretos a conseguir por cada proyecto específico a desarrollar.

2.2 La tecnología de la información puede sustentar una revolución social

Desde la explosión de Internet, la información está cada vez más al alcance de todos. Las nuevas tecnologías aportan instrumentos y herramientas que facilitan el trabajo en equipo, y más concretamente las técnicas de trabajo colaborativo, trayendo esto consigo un sin número de beneficios que impactan directamente en beneficio de la sociedad.

“Se trata de una revolución social basada en la colaboración online, la interactividad y la posibilidad de compartir contenido entre usuarios. Las experiencias del aprendizaje colaborativo apuntan a entenderlo como un proceso social de construcción de conocimiento en forma colaborativa” [Correa, 2003].

Debido al acceso creciente que posibilitan las computadoras personales, las personas se incorporan a un entorno en el que pueden ser gestoras de información o creadoras de cualquier tipo de obra. Internet ha revolucionado el conocimiento colaborativo permitiendo mediante las bitácoras personales o weblogs, las wiki y otras herramientas colaborativas, que las personas constituyan una fuente enorme y rica de contenidos cada vez más novedosos, críticos y confiables.

La gran liberalización de la información causada por la revolución científica y tecnológica de la informática ha generado una discusión cada vez más generalizada acerca del alcance del derecho de autor o del derecho de copia. Pero ha surgido una alternativa para enfrentar esta polémica, la que va logrando mejor concordancia entre los principios fundamentales que puede tener un autor en relación con una remuneración justa por la explotación no autorizada y de libre acceso a la creación cultural. Esa alternativa la propicia la licencia conocida como Creative Commons.

La nueva web permite interacción y múltiples formas de anotar, desarrollar, modificar y compartir. Se socializan los contenidos, se hacen colaborativos, además de reflejar un aspecto muy importante el cual le permite a todos y sin ningún tipo de restricción interactuar en ella, el aspecto al que se hace referencia es que en esta web no existen fronteras.

2.2.1 Herramientas colaborativas

Las Herramientas Colaborativas (HC) son todo aquel grupo de nuevas tecnologías que facilitan el trabajo en equipo, permiten el intercambio de información en diversos formatos ya sea mediante videoconferencias, chat conjunto, intercambio instantáneo de código, entre otras funcionalidades que pudieran mencionarse. Han sido desarrolladas para facilitar la creación y diseminación del conocimiento, provocando con esto un aprendizaje activo, colectivo y dinámico.

Su uso posibilita una manera diferente de intercambio y solidaridad entre personas que no necesitan estar geográficamente ubicadas en un mismo lugar. De manera que podrían comunicarse, entenderse, desarrollar un nuevo SW o impartir una clase a estudiantes que quizás estén en cualquier parte del mundo. Asimismo esto permite que no haya espacio para las limitaciones y sí para los puentes que permitan el desarrollo, sin que se necesiten características especiales para aventurarse en esta nueva forma de intercambio a través de la Web.

2.1.1.1 Wiki

Una Wiki es una herramienta que posibilita la creación y desarrollo de sitios web de manera colaborativa, en los que se exponen con mucha facilidad las ideas y criterios de diferentes autores. Permite a cualquier usuario de la red plasmar en ella su conocimiento o percepción de algún tema aun cuando la temática haya sido publicada por otro autor. Todo el tiempo está ligada a un ambiente colaborativo, de comunicación y compartimiento de ideales, fruto del trabajo en comunidad de diversos autores, en muy diferentes ubicaciones geográficas.

El éxito y avance de las Wiki es apreciable. Indudablemente han abierto puertas al beneficio y eficiencia en muchas empresas que han sustituido sus aplicaciones empresariales de *groupware*, apostando por las soluciones esta modalidad, que se tornan más sencillas y económicas. También constituyen un gran aporte a la superación educacional de manera colectiva, donde el principal exponente no será otro que la metodología de intercambio y contribución potenciada por esta herramienta colaborativa.

Esta herramienta permite compartir ideas, crear aplicaciones, proponer líneas de trabajo para el logro de determinados objetivos, rehacer o formar glosarios, diccionarios, libros de texto, manuales, repositorios de aula, por mencionar unos pocos. Todas esas acciones están fomentadas por la interacción, colaboración e intercambio dinámico que deviene de su uso entre por quienes se incorporan a ese espacio y modalidad. Generan estructuras de conocimiento compartido y colaborativo que potenciarán la creación de círculos de aprendizaje que incrementarán la difusión de la información y el conocimiento, a los que será muy difícil imponerles limitaciones y barreras algunas.

El ejemplo más fehaciente de lo anterior tal vez lo constituya *Wikipedia, la enciclopedia libre*, en línea e iniciada en el año 2001, escrita colaborativamente por voluntarios. Permite que la mayoría de sus artículos sean reeditados por cualquier interesado con acceso a la web. Normalmente se ofrece en 229 idiomas, 16 de los cuales tienen más de 50 000 artículos publicados en cada uno de ellos. En conjunto, tiene más de 5 000 000 de artículos en todos sus idiomas, incluyendo más de 1 500 000 en inglés. Está posicionada entre los veinte sitios más visitados y muchas de sus páginas han sido adoptadas por otros sitios, como *answers.com* por citar un ejemplo.

En el año 2005 la revista de ciencias *Nature* realizó una comparación entre secciones de la Wikipedia y la Enciclopedia Británica, encontrando que ambas estuvieron muy cerca entre sí en la precisión de sus artículos sobre ciencias naturales. No obstante, existen hechos muy serios que cuestionan la fiabilidad y precisión de la herramienta, lo que le hace objeto de críticas razonables por diversas causas, hechos sobre los que sus usuarios deben tomar conciencia y adoptar las medidas adecuadas en la publicación de artículos para así lograr eliminar estas malas impresiones de los usuarios [Rozakis, 2007].

2.1.1.2 Blog

Una blog, también conocida como *weblog*⁴ o bitácora (*Web log*), es un sitio web que recopila cronológicamente textos y artículos de uno o varios autores, actualizándose periódicamente.

⁴ La palabra inglesa log en español significa bitácora o diario. La unión de web y log conduce a weblog y de esta

Muchos blogs se enfocan en un tema particular, entre los que se pueden encontrar periodismo, tecnología, educación, deportes o astronomía por mencionar algunos. Otros, son simplemente blogs personales que presentan el diario del autor, sus ideas, características e intereses personales.

En esencia, un blog es una manera libre, de expresión, creación y conocimiento compartido, de fácil creación y uso, sin necesidad de especialización en informática. Por ello se han convertido en el servicio de Internet más famoso para la publicación personal de actualidades en la Web. Esta característica ha propiciado que millones de personas puedan escribir y compartir vivencias, aficiones personales e intereses profesionales. Asimismo son un medio de comunicación colectivo que promueve la creación y consumo de información original y veraz, que provoca con mucha eficiencia, la reflexión personal y social sobre los temas de los individuos, de los grupos y de la humanidad. También se ha demostrado su buena eficiencia en los procesos de enseñanza-aprendizaje, favoreciéndolos significativamente.

Los lectores pueden enviar sus comentarios, incluso de forma anónima, pero el autor es el que tiene la libertad de responder y dejar publicado lo que crea conveniente. Los mensajes que se escriben en los blogs se conocen como *posts*, suelen ser bastante intuitivos y no requieren que el usuario tenga conocimientos técnicos avanzados.

Los blogs como las Wiki son herramientas colaborativas de gran importancia, pues ayudan a las personas que las usan a tener una guía de manera lógica de cualquier contenido que se busque, organizado cronológicamente y con la ventaja de que se están actualizando periódicamente. Su esencia se fundamenta en que mantiene una forma libre de expresión, creación y compartimiento de la información y el conocimiento.

a su contracción como blog, que en español pudiera identificarse como diario o bitácora web o en la web.

2.4.2 Creative Commons

La corporación Creative Commons (CC) fue creada en el 2001 en Massachusetts con sede en la universidad de Stanford. Es una alternativa sin ánimos de lucro que propone formar un mecanismo viable del *Copyright* diferente a la tendencia del clásico. Esta alternativa facilita la construcción de obras a partir de otras ya existentes, tratando de proporcionarles a los creadores las libertades necesarias que pueden definir sobre sus obras, las que luego otros necesitarán para construir sobre ellas. Las libertades son definidas mediante etiquetas sencillas, sujetas a descripciones que las personas pueden leer. Las licencias CC se proponen conseguir un equilibrio entre autores y usuarios.

Las etiquetas que se utilizan para definir las libertades de las CC funcionan sin intermediarios ni abogados. Al desarrollar una serie de licencias libres que lo usuarios pueden añadir a sus contenidos, CC pretende determinar un espectro de los mismos de manera que puedan ser empleados fácil y seguramente como base para otros de estos. Las etiquetas se enlazan después a versiones legibles por máquinas, las que permiten a las computadoras identificar automáticamente los contenidos que se pueden compartir fácilmente.

Para que se constituya una licencia CC debe presentarse como una licencia legal, una descripción legible para seres humanos, una etiqueta legible para máquinas. Este tipo de licencias garantiza la libertad y es una expresión de las ideas de la persona asociada a la licencia. Los contenidos se identifican con la marca de CC, lo que no significa que se renuncie al *Copyright*, sino que se conceden ciertas libertades.

Estas libertades van más allá de las prometidas por el ***fair use***⁵, al ser competencia del autor o creador la decisión de definir el uso de la obra como estime conveniente, lo que puede manifestarse de diversas formas: usos no comerciales, concederle a otros usuarios las libertades que él decida, definir que no sea un uso derivado de la obra, que se permita su

⁵ En el segundo párrafo del epígrafe 2.4.3 se ofrece información sobre el significado y alcance de la enmienda *Fair Use* a la legislación norteamericana del *Copyright* respecto al SW.

uso en países en vías de desarrollo, que no se hagan copias completas y por último que se pueda usar en cualquier entorno educativo.

Estas alternativas le dan al usuario una serie de posibilidades con un mayor alcance del que establecen por defecto las leyes de *Copyright* o del Derecho de Autor. Asimismo le brindan ciertas oportunidades de usar las obras y confiarse de lo que hacen sin que haga falta contratar a un abogado. Con esas variantes CC pretende construir una capa de contenidos, gobernados por una capa de leyes razonables de *Copyright*, sobre la que otros puedan construir. Las decisiones voluntarias de individuos y creadores harán que los contenidos estén disponibles, los que a su vez permitirán reconstruir un dominio público.

Debe destacarse que la CC es una de las tantas organizaciones en busca de esas libertades. Lo que la hace especial y la diferencia de las demás es que su interés no es exclusivamente por el dominio público o porque los legisladores ayuden a construirlo. Se desempeñan especialmente con el objetivo de forjar un movimiento de consumidores y productores de contenidos que ayuden en su formación, y demostrar así la importancia que tiene para otras formas de creatividad. Su interés tampoco radica en luchar contra las personas que defienden el escenario de todos los derechos reservados, sino el de complementarlo.

La problemática creada por las leyes sobre la propiedad intelectual desde hace siglos puede que tuviera sentido para la tecnología de aquella época y durante buen tiempo posterior, pero en la actualidad con la aparición y alcance de las tecnologías de la información ello ha ido perdiendo sentido crecientemente. Se hace necesario la creación de nuevas reglas que permitan libertades diferentes, y que puedan ser usadas por las personas sin tener abogados de por medio. CC les brinda una manera efectiva de empezar a construir esas reglas.

Esta ha sido una manera de incentivar la construcción del dominio público o quizás representen pasos de avance hacia él. Este recurso es de gran importancia por el acceso abierto que permite al conocimiento; su existencia contribuye a tomar mayor conciencia sobre su necesidad de dinamizar mucho más la creatividad y la innovación y el acceso a sus productos.

Lo expuesto ejemplifica parte de los esfuerzos realizados por un grupo de personas para cambiar la estructura de derechos de autor o de derechos de copia imperante en la actualidad. El objetivo de este proyecto es que los autores y creadores tengan una mayor flexibilidad, y así conseguir que la creatividad se propague fácilmente [Lessig, 2004].

Estas modalidades no están en manera alguna separadas o aisladas del movimiento colaborativo que ha caracterizado a la creación, producción, mejora y distribución del SW, que sin lugar a dudas ocupó el centro de esas luchas, tomando forma explícita y ampliamente pública con el movimiento de SWL de FSF, diversificándose grandemente en los años subsiguientes, a pesar de sus controversias, constituyendo hoy una diversidad mucho más rica en oferta y posibilidades. No pudiera decirse que ese movimiento no esté directamente relacionado con los avances que contribuyeron a motivar al CC como algo de mucho mayor alcance respecto a las fuentes abiertas de la cultura.

Diagnóstico del Nivel de Conocimiento sobre SWL en la UCI

3.1 Introducción

Recuérdese que al exponer los métodos utilizados en esta investigación se reseñó de manera general la herramienta empírica diseñada y aplicada para diagnosticar el estado o nivel de conocimientos existente en los colectivos de la UCI sobre el SW como producto y como industria, con énfasis en el SWL.

Entre los objetivos específicos de esta parte del trabajo pueden reseñarse los siguientes:

1. Comprobar o refutar aspectos principales descritos en la situación problemática referentes a la falta de unicidad sobre conceptos fundamentales del SW como producto e industria y en especial sobre el SWL, lo que da lugar a la existencia de preconcepciones y mitos que no favorecen el trazado de una estrategia integral con mayores posibilidades de éxito y eficiencia.
2. Teniendo en cuenta que la comunidad de la UCI debiera ser una de las líderes del país en conocimientos claros y avanzados sobre estas temáticas, la realización de este diagnóstico en la misma, además de constituir una confirmación de una mayor certeza y vigencia de los problemas planteados respecto al escenario nacional, proporcionaría una base y guía de utilidad para formular y realizar las acciones que conduzcan a transformar la situación existente hacia planos superiores.

3.2 Sistema de referencia para la evaluación e interpretación de los resultados

Como sistema de referencia para la evaluación e interpretación de los resultados obtenidos con la aplicación del instrumento utilizado se utilizaron los criterios y escalas que a continuación se describen.

Las respuestas se calificaron mediante una métrica preestablecida desde el diseño del cuestionario-encuesta, mediante la que es posible clasificar los resultados en dos categorías generales y complementarias o recíprocas, aplicables por separado a cada uno de los tres

grupos en que se sub-clasificó la muestra tomada. Ello a su vez permite diagnosticar al grupo respecto al nivel de conocimiento que debiera tener según los requisitos asignados a sus roles y objetivos.

La primera de las dos categorías incluye las respuestas correctas y se le denominó “Conocimiento”; la que al mismo tiempo se clasificó en dos subcategorías, la del “Conocimiento Preciso o Consciente” y la del “Conocimiento Impreciso o Inseguro”.

La segunda corresponde al estado de “Desconocimiento”. En ella se incluyeron las respuestas correspondientes a “No sé”, lo que posibilita la subclasificación en “Desconocimiento Consciente”, así como aquellas erróneas por las que optó el encuestado en opción contraria a “No sé”, lo que permite la subclasificación como “Desconocimiento Inconsciente”, que puede tomarse también como un estado de preconcepción del sujeto sobre el tema o cuestión concreta. Este último desconocimiento es mucho más nocivo que el otro, porque pudiera dar lugar a actuaciones erróneas.

Sobre esta base es posible diagnosticar de manera sencilla ese estado de conocimiento en la UCI, mediante la expresión porcentual del conocimiento o del desconocimiento sobre el tema en cada uno de los grupos, refiriéndolo a los requisitos por roles.

No obstante, la calidad y fiabilidad de este sistema de evaluación y diagnóstico pueden estar afectadas o desviadas a causa de la incorrecta interpretación de algunas de las preguntas por una determinada cantidad de los encuestados y por otros factores de naturaleza parecida. Sin embargo, la intencionalidad de la muestra y demás controles aplicados a las variables que así lo requerían, junto con la compensación estadística propia de los errores, deben minimizar esas desviaciones, por lo que los resultados alcanzados ofrecen un diagnóstico con la calidad mínima requerida en relación con los objetivos buscados.

3.3 Resultados y análisis del diagnóstico

3.3.1 Categoría del Nivel de Conocimiento

Los resultados de esta categoría se ofrecen en la Tabla 2, representados mediante el

por ciento de conocimiento total de la cantidad de personas que integran a cada grupo y por pregunta de la encuesta. En la Figura 1 se ofrece la representación gráfica de esos resultados.

Nótese que los requisitos de conocimientos definidos a los grupos 1, 2 y 3 fueron equivalentes a Bajo, Medio y Alto respectivamente. Para diagnosticar al grupo en términos cuantitativos, se partirá de que Bajo equivale al rango de 0 a 30%, Medio de 35 a 65% y Alto de 70 a 100%.

En el caso del Grupo 1 es interesante que los resultados que obtuvo en las preguntas 5, 6, 7^a y 8 iguallen o superen su requisito superior. En el Grupo 2 ocurre algo similar con las mismas preguntas. Sin embargo, la situación es diferente en el caso del 3, por demás el grupo con roles de mayor influencia o impacto sobre los resultados a alcanzar en los objetivos del SW y especialmente del SWL. La connotación de esto último es mayor aún si se toma en cuenta que son las preguntas sobre aspectos mucho más sencillos y generalizados de estos conocimientos, lo que hace más importante el relativamente bajo nivel del grupo.

Tabla 2 - Nivel del Conocimiento por Grupos - en % por pregunta

Grupos \ Preguntas	1	2	3	4	5	6	7 ^A	7 ^B	7 ^C	8	9	10	11	12
Grupo 1	17	17	13	9	29	33	33	17	8	41	13	4	0	12
Grupo 2	34	22	12	53	73	83	76	34	32	81	56	20	42	14
Grupo 3	39	51	25	63	84	75	86	57	51	81	65	12	53	31

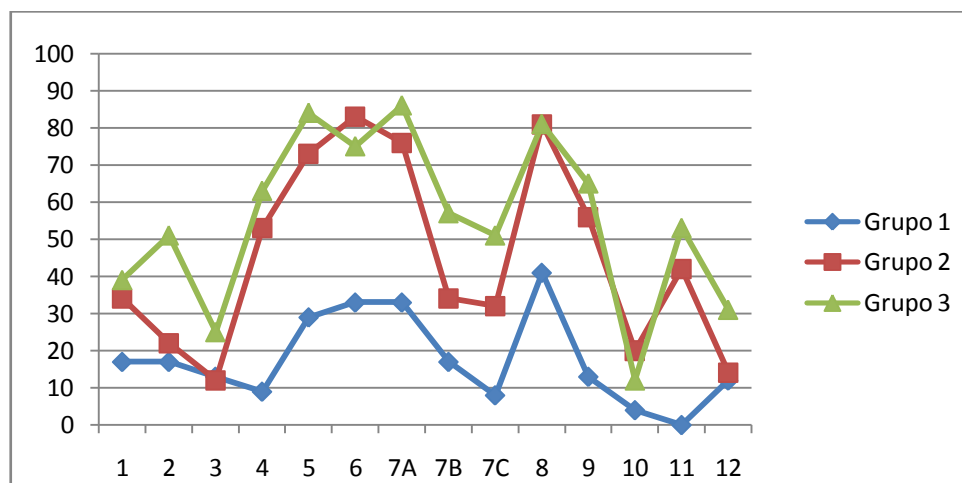


Figura 1 - Representación gráfica del Nivel de Conocimiento

Las preguntas del 1 al 4 se refieren a temas conceptualmente importantes respecto al SW como producto e industria comercial. En este segmento del conocimiento los resultados del Grupo 1 son satisfactorios. Sin embargo en los otros dos su nivel está por debajo de sus requisitos mínimos, siendo mucho más bajo en el caso del 3.

Las restantes preguntas se refieren básicamente al SWL y de complejidad creciente según aumenta su número de orden, sin dejar de ser generales. Los resultados son similares al caso anterior, siendo también significativamente bajo el nivel alcanzado por el Grupo 3, generalmente por debajo del nivel mínimo de sus requisitos.

3.3.1.1 Subcategoría del Nivel de Conocimiento Preciso

Los resultados de esta subcategoría se ofrecen en la Tabla 3. Este enfoque del diagnóstico permite apreciar otra arista de este problema. Según los requisitos el nivel del Grupo 1 aunque está dentro de su intervalo, es mucho más bajo que lo reflejado en el conocimiento total, lo que quiere decir que una parte de este es impreciso, con la connotación de que en muchos casos se pudiera tratar de preconcepciones o algo por el estilo. Una situación similar se encontró en los otros dos grupos, hecho que requiere más atención por la mayor connotación de sus roles.

Tabla 3– Nivel del Conocimiento Preciso por Grupos – en % por pregunta

Grupos \ Preguntas	1	2	3	4	5	6	7A	7B	7C	8	9	10	11	12
Grupo 1	13	13	13	9	29	33	33	17	8	0	13	4	0	12
Grupo 2	7	30	0	47	73	83	76	34	32	7	56	20	42	14
Grupo 3	10	37	23	47	84	75	86	57	51	6	65	12	53	31

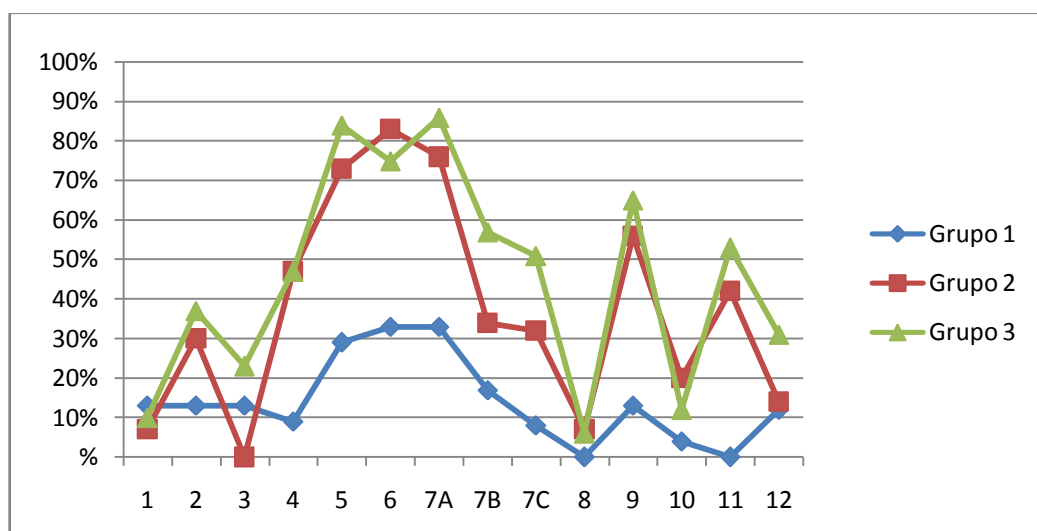


Figura 2 – Representación gráfica del nivel de Conocimiento Preciso

Un breve examen de este resultado en el gráfico de la Figura 2 permite apreciar cómo en esta subcategoría es significativo el bajo nivel del Grupo 2 respecto a sus requisitos mínimos en los temas de mayor connotación; esa situación es mucho más pronunciada en el caso del Grupo 3.

3.3.1.2 Subcategoría del Nivel de Conocimiento Impreciso

Para ampliar un poquito más el alcance del análisis realizado en el apartado anterior, en la Tabla 4 y Figura 3 se recogen los resultados porcentuales por grupos sobre la parte del conocimiento que resultó imprecisa. Estos permiten apreciar que en el segmento de las preguntas 1 a la 4, relativas a ideas básicas del origen y naturaleza del SW como producto comercial y su industria los grupos 2 y 3 reflejan una parte significativa de lo que se le

atribuye como conocimiento impreciso, lo que no es deseable.

Resulta interesante el caso de la pregunta 8. Esta solicita seleccionar una opción de a qué institución dirigirse a solicitar asesoría cuando usted tiene dudas sobre aspectos legales relativos al SWL, ofreciendo a la IP de la universidad entre las opciones de respuesta. En la evaluación de este resultado la opción de la IP se calificó como conocimiento impreciso, teniendo en cuenta que esta forma parte del Grupo 3, el que al mismo tiempo alcanzó muy bajo nivel en el diagnóstico, razón por la que esta opción se clasificó en esa subcategoría. Dentro de este caso la muestra de la IP es un tanto recursiva al responder que en ella misma buscarían la asesoría, hecho que tiene dos lecturas, la primera refleja el prestigio y confianza que se tiene en esa área respecto a un tema que así debe ser; la segunda es que, al formar parte del Grupo 3, es presumible que el nivel integral que este obtuvo por debajo de sus requisitos mínimos, aconseja que esta situación se modifique lo antes posible en el área.

Tabla 4– Nivel del Conocimiento Impreciso por Grupos – en % por pregunta

Grupos \ Preguntas	1	2	3	4	5	6	7 ^A	7 ^B	7 ^C	8	9	10	11	12
Grupo 1	4	4	0	0	-	-	-	-	-	41	-	-	-	-
Grupo 2	27	22	12	6	-	-	-	-	-	74	-	-	-	-
Grupo 3	29	14	2	16	-	-	-	-	-	75	-	-	-	-

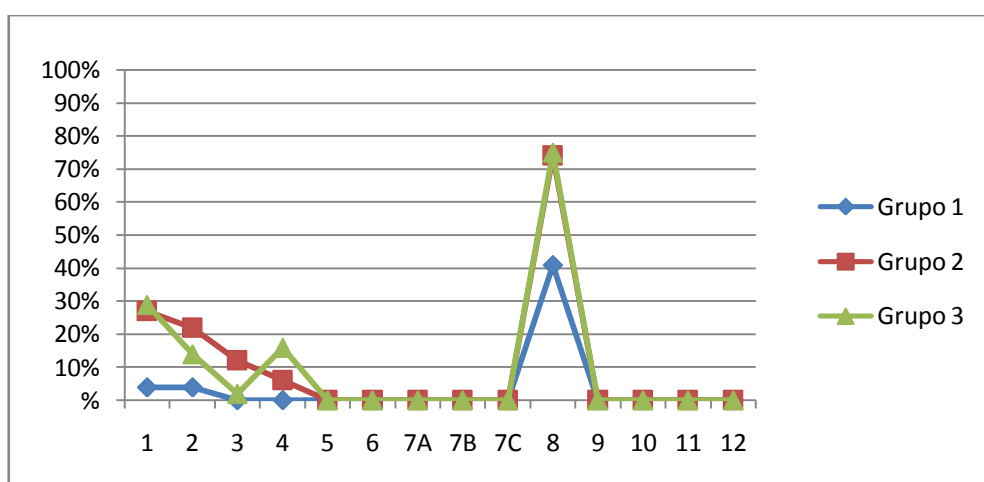


Figura 3– Representación gráfica del nivel de Conocimiento Impreciso

3.3.2 Categoría del Nivel de Desconocimiento

Esta categoría es complementaria a la 3.3.1 o nivel de conocimiento total. Cuantitativamente dice poco a nivel de la categoría, sin embargo, al tener en cuenta su desglose en dos subcategorías, la de los que tienen conciencia de que no saben y la inconsciente, estos elementos ameritan un análisis adicional de esta parte del diagnóstico. Si se observa la Tabla 7 y el gráfico de la Figura 6 se podrá apreciar que la mayor parte del desconocimiento es del tipo impreciso o inconsciente. Eso es especialmente cierto para los grupos 2 y 3, contrario a lo deseable, toda vez que la posibilidad de que ello refleje preconcepciones, pensar o creer que se conoce un tema, es algo más nocivo que tener conciencia de que no se sabe. Esos resultados recomiendan fuertemente que se adopten medidas urgentes de superación postgraduada y capacitación.

Tabla 5 - Nivel de Desconocimiento por Grupos – en % por pregunta

Grupos \ Preguntas	1	2	3	4	5	6	7 ^A	7 ^B	7 ^C	8	9	10	11	12
Grupo 1	83	83	87	91	71	67	67	83	92	59	87	96	100	88
Grupo 2	66	46	81	47	27	17	24	66	68	19	44	80	58	88
Grupo 3	61	49	75	37	16	25	14	43	49	19	35	88	47	69

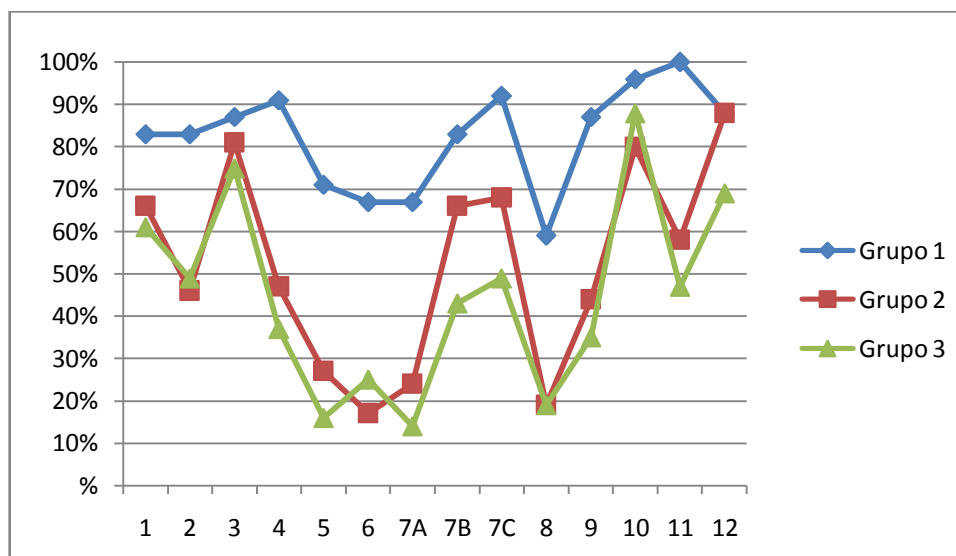


Figura 4– Representación gráfica del Nivel de Desconocimiento

Tabla 6– Nivel de Desconocimiento Consciente por Grupos – en % por pregunta

Grupos \ Preguntas	1	2	3	4	5	6	7 ^A	7 ^B	7 ^C	8	9	10	11	12
Grupo 1	66	58	50	58	46	63	59	75	84	59	74	88	88	67
Grupo 2	42	22	13	3	3	3	9	22	32	19	3	47	37	22
Grupo 3	37	12	12	8	6	2	10	18	33	19	12	62	39	30

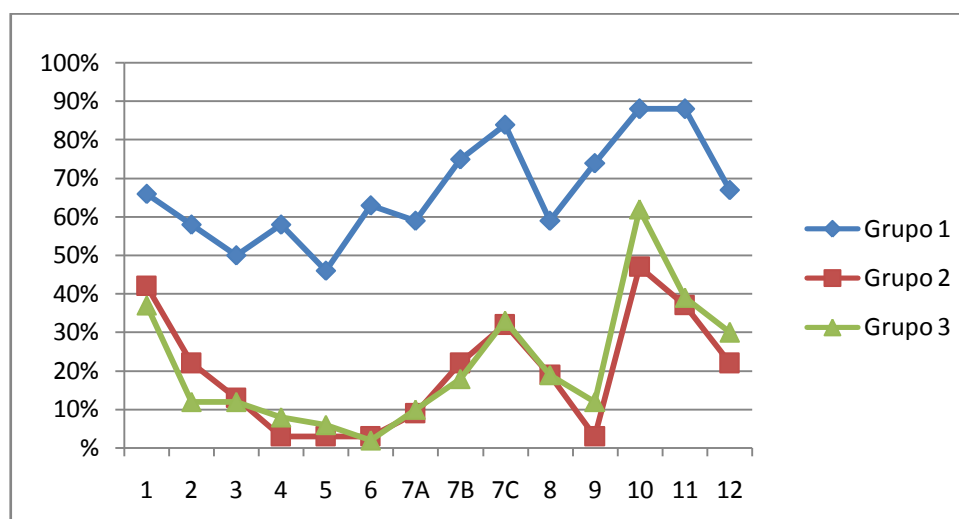


Figura 5 – Representación gráfica del Nivel de Desconocimiento Consciente

Tabla 7 – Nivel de Desconocimiento Inconsciente o Impreciso por grupos – en % por pregunta

Grupos \ Preguntas	1	2	3	4	5	6	7 ^A	7 ^B	7 ^C	8	9	10	11	12
Grupo 1	17	25	37	33	25	4	8	8	8	-	13	8	12	21
Grupo 2	24	26	68	44	24	14	15	44	36	-	41	33	21	64
Grupo 3	24	37	63	29	10	23	4	25	16	-	23	26	8	39

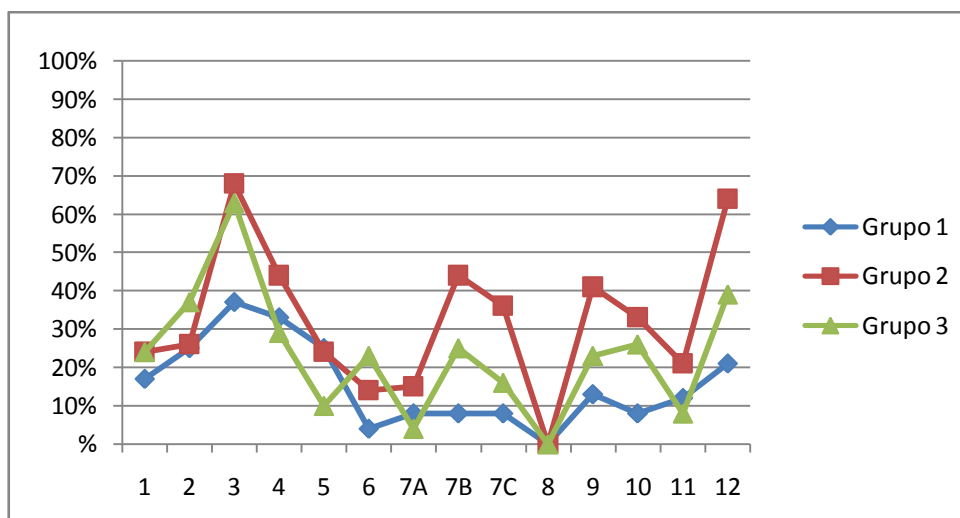


Figura 6– Representación gráfica del Nivel de Desconocimiento Inconsciente

Conclusiones

La investigación realizada corrobora que la situación problemática existente en torno al conocimiento claro sobre la naturaleza verdadera del SW en general y a la gran controversia entre éste, el propietario y el SWL en particular, supera significativamente a la formulación realizada inicialmente. En el informe presentado se sustenta esta conclusión general y a las se ofrecen a continuación.

1. Se encontraron y ofrecen las evidencias que contribuyen a explicar cómo la naturaleza de producto intelectual del SW, que no obedece al paradigma industrial clásico, y su rol de universalizar la aplicación de la máquina real, incentivaron desde el inicio a su desarrollo en colaboración por comunidades académicas y corporativas, independientemente de la naturaleza e intereses materiales que predominaran en cada momento y vertientes de su historia.
2. También se muestra cómo la incompatibilidad inicial y el desarrollo del HW, junto con el avance de la ciencia y tecnología del SW, indujeron la necesidad creciente de los sistemas operativos y de su complejidad, convirtiéndolos en uno de los componentes principales del incremento y diversificación de comunidades, intereses y modalidades de desarrollo e intercambio en esta industria.
3. La historia del UNIX y del lenguaje C, desde su génesis y su evolución hasta el presente, ofrece el hilo rojo o guía para facilitar la comprensión y explicación de una gran parte la problemática estudiada.
4. Se exponen con alcance sistémico las principales modalidades del llamado SWL desde su origen cómo fenómeno explícito con el movimiento GNU, hasta la expresión actual de F/OSS como identificación de un escenario de software libre y de código abierto como expresión de un fenómeno variado y enriquecido en sus casi veinticinco años de existencia.
5. Las concepciones, modalidades con sus aspectos jurídicos y el inevitable influjo del

desarrollo de la Ingeniería y la Industria del SW sobre ellas, explican la justeza del enfoque F/OOS como integrador y actualizador, cuya comprensión se dificulta por cierto romanticismo acerca de la originalidad del Manifiesto GNU, la falta de información sistematizada y fiable y el consiguiente escaso conocimiento preciso necesario. Estos problemas se complican porque en su esencia tienen una profunda connotación filosófica e ideológica y consiguientemente política

6. Se ofrecen elementos que modestamente contribuirán a la solución de estos problemas en un nivel básico y que argumentan la necesidad de su conocimiento por prácticamente todos los actores de los procesos del SWL, desde la etapa inicial de los proyectos hasta su distribución.
7. El diagnóstico realizado a la muestra intencionada de la comunidad de la UCI, dividida en grupos representativos de sus actividades fundamentales y complementarias, permitió comprobar que su nivel de conocimientos sobre SW en general y el SWL en particular, como producto e industria, está por debajo de los requerimientos mínimos para los dos grupos fundamentales de su actividad.
8. También permitió confirmar que el conocimiento y desconocimiento constatados, reflejan la presencia de preconcepciones o creencias sobre aspectos básicos, claves para un mejor desempeño y aseguramiento de los objetivos de la UCI.
9. Al partir del supuesto que el nivel de conocimientos sobre estos temas en la universidad debe estar muy por encima de la media nacional, se puede entonces inferir que esa situación en el país debe ser necesariamente más compleja y de urgente atención.
10. Finalmente debe señalarse que, aunque en un nivel básico, se alcanzaron los objetivos de la investigación. Este informe, junto con parte de la bibliografía encontrada y revisada, constituyen una base útil para apoyar las acciones a realizar desde el corto plazo, constituyendo un modesto resultado de esta investigación.

Recomendaciones

1. Continuar las investigaciones sobre la situación problemática que motivó a este trabajo, profundizando en los en los problemas identificados en este trabajo y en especial en las conclusiones.
2. Tomar en cuenta e incorporar estos resultados en los cursos de postgrado que sobre SWL se imparten en la universidad. Realizar una evaluación comparativa entre sus contenidos y las precisiones que resultaron de este trabajo.
3. Sobre la base del diagnóstico realizado en la universidad, organizar y llevar a cabo cursos y seminarios de postgrado y capacitación que contribuyan a elevar el nivel de conocimientos de los grupos dos y tres en el más breve plazo de tiempo. Debatir aquellos aspectos principales planteados.
4. Diseñar e introducir curso optativo que vincule este tema con la Historia de la Informática, cuyo eje central lo constituya la historia del SW desde la perspectiva de este proyecto, incluyendo su dimensión filosófica, política e ideológica. En este deben conjugarse intereses de las disciplinas afines de la especialidad, ciencias empresariales y marxismo.
5. Seleccionar los contenidos y aspectos de mayor interés de los presentados en este informe para su divulgación el país por la vía de la Oficina Nacional para la Informatización, de las universidades y otras de interés.
6. Sistematizar el estudio de los temas relacionados con la Sociedad de la Información, las redes y la economía del conocimiento con el objetivo de establecer vínculos y mostrar su influencia en el marcado proceso evolutivo de la Tecnología de la Información y su naturaleza colaborativa.
7. Revisar este informe con el objetivo de su publicación. Utilizar sus contenidos para profundizar en temas a presentar a eventos científicos nacionales e internacionales.

Bibliografía**Bibliografía Consultada**

ALTISEN, C. J. *Epistemología*. Argentina, 2001. [Disponible en: <http://www.librosencred.com/autores/claudioaltisen.html>] Consultado en noviembre del 2008.

ARMENDÁRIZ, L. M. *Sobre el 'código abierto' (open source)*, 2006. [Disponible en: http://guimi.net/descarga/tec-docs/Sobre_el_OS.pdf] Consultado en marzo del 2008.

BAIN, M.; M. G. RODRÍGUEZ, *et al. Aspectos legales y de explotación del software libre Parte I*. 1ra ed. Barcelona: Eureka Media, 2004.

BAIN, M.; M. G. RODRÍGUEZ, *et al. Aspectos legales y de explotación del software libre Parte II*. 1ra ed. Barcelona: Eureka Media, 2004.

BENÍTEZ, L. L. B. *El marco conceptual del derecho de autor*, 2003. [Disponible en: <http://www.cinfo.cu/Userfiles/file/Cinfo/cinfo2003/v34n1a2003/MARCO.PDF>] Consultado en noviembre del 2008.

BENKLER, Y. *The wealth of networks*. 1ra ed. New Haven and London: Yale University, 2006. [Disponible en: http://www.benkler.org/Benkler_Wealth_Of_Networks.pdf]

BURKE, M. *Pirates Of Silicon Valley*, (Documental) 1999. Disponible en: http://internos.uci.cu/Teleclases/Teleclases.asp?id_as=29 (este recurso sólo está disponible dentro de la UCI).

CAMPBELL-KELLY, M. *From Airline Reservations to Sonic the Hedgehog. A History of the Software Industry*. 1ra ed. London: The MIT Press, 2003. .

CORREA, L. M. Z. *Aprendizaje colaborativo: una nueva forma de Diálogo Interpersonal y en Red*, 2003. [Disponible en: http://www.educrea.cl/documentacion/articulos/aprendizaje/09_aprendizaje_colaborativo.html]

Consultado en mayo del 2008.

GALLARDO, C. R. *Copyright, Copyleft y Creative Commons*, 2007. [Disponible en: <http://www.lawebera.es/comunidad/articulos/legal/copyright-copyleft-creative-commons.php>

Consultado en febrero del 2008.

GONZÁLEZ, F. S. *Herramientas colaborativas para la enseñanza usando tecnologías web: weblogs, wikis, redes sociales y web 2.0*, 2005. [Disponible en: http://gabinetedeinformatica.net/descargas/herramientas_colaborativas2.pdf Consultado en diciembre del 2008.

HARDINGS, J. and A. FUENTES. *Software Libre: introducción histórica*, 2003. [Disponible en: <http://www.hardings.cl/publications/hardings2003intro.pdf> Consultado en abril del 2008.

LEÓN, R. A. H. and S. C. GONZÁLEZ. *El paradigma cuantitativo de la investigación científica*. La Habana: EDUNIV, 2002.

LEÓN, R. C. *Aprendizaje colaborativo vía Internet. Aplicación a la teleoperación de un observatorio astronómico de libre acceso*. Madrid, 2007. [Disponible en: <http://www.ciclope.info/docs/SufInv-rcedazo-trabajoDoctorado.pdf> Consultado en mayo del 2008.

LESSIG, L. *Free Culture*. 1ra ed. New York: Penguin, 2004.

MARTÍNEZ, E. M. y O. J. F. MURILLO. *Principios y filosofía del software libre*, 2007. [Disponible en:

http://eveliux.com/mx/index.php?option=com_content&task=view&id=187&Itemid=26

Consultado en diciembre del 2008.

MOORE, J. T. S. *Revolution OS*, (Documental) 2001. Disponible en: http://inter-nos.uci.cu/Teleclases/Teleclases.asp?id_as=29 (este recurso sólo está disponible dentro de la UCI).

- MURILLO, O. J. F. and E. M. MARTÍNEZ. *El origen del Software Libre*, 2007. [Disponible en: http://www.labrechadigital.org/labrecha/index2.php?option=com_content&do_pdf=1&id=115 Consultado en octubre del 2008.
- NONIUS, J. *Introducción a las licencias de software libre*, 2002. [Disponible en: <http://www.laespiral.org/articulos/licencias/licencias.html> Consultado en abril del 2008.
- PASCUAL, J. S.; J. M. G. BARAHONA, *et al. Introducción al software libre*. 2.0.1 ed. España, 2007. [Disponible en: <http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.pdf>
- PLANTADA, R. X. Las licencias Creative Commons: ¿una alternativa al copyright?. *Vocopers*, Marzo de 2006, nº2. [Disponible en: <http://www.universia.net.co/tesis-de-grado/view-document/documento-360.html> Consultado en abril del 2008.
- RAYMOND, E. S. *La Catedral y el Bazar*, 1998. [Disponible en: <http://www.softlibre.salta.org.ar/slw/pdf/catedralbazar.pdf> Consultado en enero del 2008.
- RAYMOND, E.; M. K. MCKUSICK, *et al. Open Sources: Voices from the Open Source Revolution* .1 ed. O'Reilly, 1999. Disponible en: <http://oreilly.com/catalog/opensources/book/toc.html>
- SAMÓN, R. P. *Metodología para la migración a Software Libre de la Universidad de las Ciencias Informáticas*. Universidad de las Ciencias Informáticas. La Habana, 2007.
- SARAVIA, D. *Diccionario conceptual del conocimiento libre*, 2005. [Disponible en: <http://docs.hipatia.info/diccionario/> Consultado en junio del 2008.
- SOLER, O. B. *Análisis jurídico del software libre. Sus particularidades en Cuba.*, 2006.
- STALLMAN, R. *CONFERENCIA: "Copyright y Comunidad"*. Argentina, 2004.
- STALLMAN, R. *Software libre para una sociedad libre*.1ra ed. Madrid: Traficantes de Sueños,2004. [Disponible en: <http://www.worcel.com/archivos/6/Software%20libre%20para%20una%20sociedad%20libre.%20Richard%20Stallman.pdf>

TANENBAUM, A. *Operating systems design and implementation*. 3ra ed. New Jersey: Pearson Prentice Hall, 2006.

VERCELLI, A. H. *Creative Commons y la profundidad del copyright*, 2003. [Disponible en: http://www.sindominio.net/afe/dos_copyleft/cc.pdf Consultado en junio del 2008.

MARÍN, J. E. T. *Uso de Software Libre para la integración de servicios sobre LDAP en la EFTS Frank País García*. Universidad de las Ciencias Informáticas. La Habana, 2006.

Software Magazine. 2008. [Disponible en: http://www.freesoftwaremagazine.com/columns/2008_google_summer_code_21_projects_im_excited_about Consultado en mayo del 2008.

FELLER, J.; B. FITZGERALD, *et al.* *Perspectives on Free and Open Source Software*. 1er ed. Massachusetts: The MIT, 2005. [Disponible en: <http://mitpress.mit.edu/books/chapters/0262562278.pdf>

ROZAKIS, L. *Writing great research papers*. 2da ed. Mc Graw- Hill, 2007

Bibliografía Referenciada

BAIN, M.; M. G. RODRÍGUEZ, *et al.* *Aspectos legales y de explotación del software libre Parte I*. 1ra ed. Barcelona: Eureka Media, 2004.

BURKE, M. *Pirates Of Silicon Valley*, (Documental) 1999. Disponible en: http://internos.uci.cu/Teleclases/Teleclases.asp?id_as=29 (este recurso sólo está disponible dentro de la UCI).

CAMPBELL-KELLY, M. *From Airline Reservations to Sonic the Hedgehog. A History of the Software Industry*. 1ra ed. London: The MIT Press, 2003. .

HARDINGS, J. and A. FUENTES. *Software Libre: introducción histórica*, 2003. [Disponible en: <http://www.hardings.cl/publications/hardings2003intro.pdf> Consultado en abril del 2008.

LESSIG, L. *Free Culture*. 1ra ed. New York: Penguin, 2004.

MOORE, J. T. S. *Revolution OS*, (Documental) 2001. Disponible en: http://internos.uci.cu/Teleclases/Teleclases.asp?id_as=29 (este recurso sólo está disponible dentro de la UCI).

PASCUAL, J. S.; J. M. G. BARAHONA, *et al.* *Introducción al software libre*. 2.0.1 ed. España, 2007. [Disponible en: <http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.pdf>

RAYMOND, E. S. *La Catedral y el Bazar*, 1998. [Disponible en: <http://www.softlibre.salta.org.ar/slsw/pdf/catedralbazar.pdf> Consultado en enero del 2008.

RAYMOND, E.; M. K. MCKUSICK, *et al.* *Open Sources: Voices from the Open Source Revolution* .1 ed. O'Reilly, 1999. Disponible en: <http://oreilly.com/catalog/opensources/book/toc.html>

SARAVIA, D. *Diccionario conceptual del conocimiento libre*, 2005. [Disponible en: <http://docs.hipatia.info/diccionario/> Consultado en junio del 2008.

SOLER, O. B. *Análisis jurídico del software libre. Sus particularidades en Cuba*, 2006.

TANENBAUM, A. *Operating systems design and implementation*. 3 ed. New Jersey: Pearson Prentice Hall, 2006.

MARÍN, J. E. T. *Uso de Software Libre para la integración de servicios sobre LDAP en la EFTS Frank País García*. Universidad de las Ciencias Informáticas. La Habana, 2006.

Software Magazine. 2008. [Disponible en:

http://www.freesoftwaremagazine.com/columns/2008_google_summer_code_21_projects_im_excited_about Consultado en mayo del 2008.

FELLER, J.; B. FITZGERALD, *et al. Perspectives on Free and Open Source Software*. 1er ed. Massachusetts: The MIT, 2005. [Disponible en:

<http://mitpress.mit.edu/books/chapters/0262562278.pdf>

ROZAKIS, L. *Writing great research papers*. 2da ed. Mc Graw- Hill, 2007

CORREA, L. M. Z. *Aprendizaje colaborativo: una nueva forma de Diálogo Interpersonal y en Red*, 2003. [Disponible en:

http://www.educrea.cl/documentacion/articulos/aprendizaje/09_aprendizaje_colaborativo.html

Consultado en mayo del 2008.

Glosario de términos

Hacker: término de difícil traducción, mitad genio y mitad pirata, entre virtuoso de la computadora y pirata tecnológico. Steven Levy, les dedicó su libro Hackers. *Heroes of the Computer Revolutions* (1984), donde explica que el término deriva originalmente de la expresión hack, que se usaba para indicar bromas refinadas que los estudiantes se hacían periódicamente en la época del surgimiento de las computadoras, después se fue extendiendo a toda la comunidad científica que demostraba innovación, virtuosismo y estilo técnico.

DECUS: Sociedad de Usuarios de las computadoras de *Digital Equipment Corporation*; la mayoría de los miembros eran los programadores que escribieron el código para las máquinas de la DEC de manera cooperativa. El actual ICID utilizó infinidad de documentos y programas de DECUS.

Mainframe: computadora central o *mainframe* era una máquina potente y costosa usada principalmente por una gran compañía para el procesamiento de una gran cantidad de datos; por ejemplo, para el procesamiento de transacciones bancarias. Hoy se habla de los dinosaurios para referirse a ellas. Modernamente ese concepto se mantiene sobre la base de arquitecturas con potentes microcomputadoras y suelen costar algunos cientos de miles de dólares.

Driver: programa cuya función es controlar el funcionamiento de un dispositivo o periférico de entrada salida de la computadora bajo un determinado sistema operativo.

Núcleo: En informática, el **núcleo**, también conocido en español con el anglicismo *kernel*, de raíces germánicas como *kern*, es la parte fundamental de un sistema operativo. Es el software responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma más básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Es la parte del sistema operativo responsable de mantener la fluidez de datos entre los discos duros, la memoria, las impresoras, la pantalla de video y todas las partes que se hallan unidas al mismo.

GCC: *GNU Compiler Collection*, es un conjunto de compiladores creados por el proyecto GNU.

Feedback: Equivale a retroalimentación o retroacción. Consiste en introducir o realimentar como datos o señal de entrada una muestra de los resultados obtenidos o señal de salida, para considerar al inicio del nuevo proceso, una mejora o rectificación del mismo si resultara necesario.

Fair Use: es una doctrina en la ley norteamericana de derecho de autor, introducida como enmienda especialmente en la lucha contra el reforzamiento del *Copyright* en el software, que permite usos limitados de materiales protegidos por el copyright sin tener el requerimiento del permiso de los dueños o de los poseedores de los derechos y sin que constituya delito. Se justifica su uso cuando es para la enseñanza, la investigación o fines similares.

Groupware: se refiere al conjunto de programas informáticos que integran el trabajo en un sólo proyecto con muchos usuarios concurrentes que se encuentran en diversas estaciones de trabajo, conectadas a través de una red (internet o intranet).

ISA: *Instruction Set Architecture*, conjunto o repertorio de instrucciones de máquina de la arquitectura de una computadora. Antiguamente cada fabricante tenía su propio ISA lo que hacía muy difícil el desarrollo y la inter-cambiabilidad del SW. Un resultado del movimiento colaborativo de producción del SW, con ejemplo cumbre en el UNIX y C originales inició la solución de ese gran problema, lo que finalmente convergió gradualmente a la unificación del hardware sobretodo con la evolución del Intel 8086 y la previa influencia de la arquitectura de la familia DEC PDP-11.

Apéndices del documento

Apéndice A - Cronología

1955: Se crea la asociación de usuarios de IBM llamada SHARE, que significa compartir, que se dedicaban a escribir y distribuir software de manera colaborativa entre los clientes de las computadoras de esa compañía. Fue la sucesora de una similar, para incrementar las ventas de un nuevo modelo de *mainframe* IBM.

1957: Kenneth H. Olsen y Harlan Anderson crean una empresa de fabricación de equipos de pruebas a la que dan el nombre de DEC (Digital Equipment Corporation). Fueron los creadores de las minicomputadoras PDP – Esas siglas corresponden a *Programmable Data Processor*. Ese nombre, que en la práctica era una computadora, fue una táctica de mercadeo de los creadores de DEC para introducir su innovador concepto de las minicomputadoras, para obviar el nada buen prestigio de las *mainframe*, caracterizadas como muy caras, poco seguras y muy difíciles de utilizar.

1961: Fue fundada DECUS, Sociedad de Usuarios de las minicomputadoras de *Digital Equipment Corporation –DEC*. La mayoría de los miembros eran los programadores que escribieron el código para las máquinas de la DEC de manera cooperativa. El actual ICID utilizó infinidad de documentos y programas de DECUS

1964: Los profesores John Kemeny y Thomas Kurtz del Darmouth College, diseñan el lenguaje BASIC (Beginners All purpose Symbolic Instruction Code), que incorpora características del FORTRAN y ALGOL, pero mucho más fácil de aprender y poco costoso de implantar y utilizar.

1965: Aparece públicamente el lenguaje BASIC.

1965: MULTICS: MIT, General Electric, Bell Labs de AT&T.

1968: Bell Labs se retira del proyecto del MULTICS.

1969: Ken Thompson, Dennis Ritchie, Rudd Canaday y otros colaboran y desarrollan el feto de UNICS en lenguaje ensamblador.

1970: Brian Kernighan lo bautiza UNIX.

1971: El bebé UNIX en Lenguaje B en una DEC PDP-7.

1972: UNIX comienza a distribuirse a universidades y centros de investigación.

1972: UNIX en Lenguaje C en una minicomputadora DEC PDP-11 con innovadoras herramientas de procesamiento de texto.

1973: Bell Labs y AT&T usan UNIX internamente en unas doce computadoras.

1974: Bell Labs publica la descripción de UNIX: un sistema operativo portable escrito en un lenguaje avanzado de 3ª generación, el Lenguaje C, multiusuario, multitarea, con capacidad de procesamiento en línea, sistema de archivos jerárquico, editor, compilador de C y herramientas de procesamiento de texto.

1973: UNIX llega a la Universidad de California en Berkeley. Comienza la historia de UNIX BSD.

1974: La revista Popular Electronics (EE.UU.) publica un anuncio de la empresa MITS (Micro Instruments & Telemetry Systems Inc) con la oferta de un pequeño ordenador, el ALTAIR 8800, con sus periféricos correspondientes, en dos modalidades: kit (desmontado) o ensamblado. Había dado comienzo la historia de las computadoras personales.

1975: Bell Labs distribuye UNIX con su código fuente de manera abierta y libre a las personas del ambiente académico y científico, con la condición de reportar toda investigación y mejora que se desarrolle con el Sistema. Así es como llega UNIX a la universidad de Berkeley.

1975: William H. Gates, funda la empresa Microsoft líder indiscutible del software para computadoras personales.

1983: Richard Stallman escribe el Manifiesto GNU, en el que solicita y proclama el regreso a la compartición pública de software.

1984: Comienza el proyecto GNU. Los desarrolladores que colaboran en él, inicialmente coordinados por Richard Stallman, comienzan a crear un gran número de herramientas similares a las que había en UNIX, incluyendo un editor (Emacs) y un compilador (GCC). La meta es construir un sistema operativo completamente libre.

1985: Richard Stallman funda la Free Software Foundation. Entre otros fines, funcionará como centro receptor de fondos y recursos que ayuden al desarrollo del proyecto GNU y como dueño de la propiedad intelectual generada por el proyecto.

1987: Desarrollo de Minix por el profesor Andrew S. Tanenbaum. Constituye un clon del sistema operativo UNIX, distribuido junto con su código fuente y desarrollado con fines educativos.

1989: Se funda Cygnus, la primera empresa dedicada fundamentalmente a proporcionar servicios comerciales para el software libre (entre ellos, soporte, desarrollo y adaptación de programas libres).

1990: La Free Software Foundation anuncia su intento de construir un núcleo que se llamará GNU Hurd. La meta de este proyecto es completar el mayor hueco que queda en la estrategia del proyecto GNU de construir un sistema operativo completo.

1991 Agosto: Linus Torvalds, anuncia que ha empezado a trabajar en un núcleo tipo UNIX libre, usando herramientas de GNU, como GCC. Su meta en esa época es construir un Minix Libre.

1991 Octubre: Linus Torvalds libera la primera versión de su núcleo, aún muy primitiva, que es llamada Linux.

1997 Enero: Eric S. Raymond presenta su artículo *The Cathedral and the Bazaar* (La Catedral y el Bazar), con sus opiniones sobre por qué funcionan ciertos modelos de desarrollo del software libre.

1998 Enero, 22: Netscape declara su intención de distribuir como software libre el código de su Navigator, que había sido el líder en el mercado de navegadores de la web.

1998 Febrero, 3: Chris Peterson, Todd Anderson, John Hall, Larry Augustin, Sam Ockman y Eric S. Raymond se reúnen para estudiar las consecuencias del anuncio de Netscape de liberar su navegador y deciden promover el término *Open Source software* (software de fuente abierta), usándolo como una marca que garantice que los productos que la lleven sean software libre. Los promotores de este término entienden que es más adecuado para el mundo empresarial que el habitual hasta ese momento, free software (software libre). Para gestionar el término, se crea la Open Source Initiative.

1998 Marzo, 31: Netscape publica en Internet gran parte del código fuente de su Navigator.

2001 Enero, 15: Comienza la Wikipedia. La idea de hacer una enciclopedia usando un wiki como soporte informático, donde cualquiera puede en principio colaborar, aplicando métodos de trabajo muy similares a los habituales en el software libre, se hace realidad.

2002 Diciembre, 16: Se publican las primeras licencias Creative Commons (aunque el proyecto fue lanzado en 2001).

2003 Enero, 22: La versión en inglés de la Wikipedia llega a los 100.000 artículos. Poco después la versión en alemán llega a los 10.000.

2004 Septiembre, 20: La Wikipedia llega al millón de artículos en 105 idiomas.

2006 Enero, 16: Se publica el primer borrador de la GPLv3, un intento de actualizar la GPL, que es en ese momento (y con diferencia) la licencia más utilizada por los proyectos de software libre. Con esto, comienza un proceso abierto de debate sobre los cambios.

2006 Marzo, 1: La Wikipedia en inglés llega al millón de artículos.

2007 Febrero, 23: Se publica la versión 3.0 de las licencias Creative Commons.

Apéndice B - Elementos del escenario internacional del desarrollo de la industria de software.

A continuación se ofrecen algunos elementos tomados del libro “From AIRLINE RESERVATIONS to SONIC THE HEDGEHOG – A History of the SOFTWARE INDUSTRY” del Dr. Martin Campbell-Kelly, profesor de Ciencia de la Computación de la universidad de Warwick, publicado en el 2004 como parte de la serie de Historia de la Computación por la editorial MIT Press en su primera edición. En esta obra se realiza una profunda y cuidadosa investigación del desarrollo de la industria del software en los EE.UU. como uno de los pioneros de la misma. Además de ofrecer elementos básicos de la génesis y evolución de este rama de la industria, ofrece su evolución desde sus primeros elementos en la década de los años de 1950 hasta convertirse en el cuarto mayor sector industrial de la economía norteamericana a principios del presente siglo. También contribuye a corregir la popular concepción de que Microsoft es la firma que está en el centro del universo de esa industria, mostrando que su peso no es mayor del 10% entre otros aspectos de mucho interés.

Como aspectos de especial interés para la tesis “Contribución a la Epistemología del Software Libre”, para la que es fundamental la génesis del software en general, a continuación se ofrecen selecciones resumidas tomadas de dicha obra [Campbell-Kelly, 2004].

En enero de 1952 la industria de software era relativamente nueva, no existían en ese entonces los términos de “programación” o “software”. En ese año un artículo publicado por la revista Fortune abordó de manera básica algo relacionado con el tema haciendo alusión a los términos de programación o software de la siguiente forma: “A la conmutación de algunos interruptores”, para hacer referencia como la máquina (la UNIVAC) podía cambiar de la

manipulación de masas de datos estadísticos a la solución de ecuaciones diferenciales para los científicos o a la manipulación de listas de nóminas, o a cálculos y a la elaboración de los cheques necesarios para los procesos de negocios. No es difícil percatarse de que el concepto de programación o software en ese entonces era algo desconocido, que quizás el autor conociera o entendiera o probablemente considerara que el público no estaba preparado para el mismo, prefiriendo utilizar la metáfora citada.

No fue hasta 1966 que apareció otra publicación importante donde se hiciera referencia al software. El Bussines Week presenta un artículo donde hacía referencia a la falta de programadores que existía en esa época y enfatizaba sobre las gloriosas oportunidades a las que apuntaba la **“naciente industria del software”**.

En 1980, pasados 14 años regresa el Bussines Week con otra publicación, era lo más importante que realizaba relacionado con la industria del software desde su artículo en 1966. Aunque se puso mucho empeño por parte del Comité de Imagen de la ADAPSO (siglas en inglés de la Asociación norteamericana de Organizaciones de Procesamiento de Datos) para que el Bussines Week le hiciera publicidad a la naciente industria del software; el artículo era bien parecido al de los años 1966 donde solo se especificaba la falta de software y de programadores, aunque abordaba los importantes avances recientes de la industria.

El siguiente reportaje significativo relacionado con la industria de software de Business Week apareció en 1984, resultando un tanto diferente al de 1980, pues ya en ese entonces no era necesaria tanta explicación para que se comprendiera lo relacionado con la industria de software, la que había logrado tal avance que el mundo conocía crecientemente sobre esa dinámica e increíble línea de productos.

Se trató de hacer un breve recuento demostrativo donde no es difícil percatarse que la industria del software fue evolucionando crecientemente y que en sus inicios no se conocían siquiera los términos que se manejan hoy día de programación o software. Como puede apreciarse también, es a mediados de la década de 1960 cuando publicaciones influyentes comienzan a hablar de **“la naciente industria del software”**.

Otro aspecto de mucho interés evidenciado en esta obra es el relativo a la evolución y modalidades del desarrollo de esta industria que fueron apareciendo a intervalos de unos diez años unas de otras, entre 1955, 1965 y 1975. Una breve reseña es la siguiente:

- **Empresas Contratistas de Software.**

Este fue el primer tipo de empresa y forma de comercialización y entrega de software. Se desarrolló desde mediados de los 1950, orientadas al uso y aplicaciones de las computadoras centrales de primera y segunda generación –mainframe computers. Esas empresas desarrollaban programas para clientes de las grandes corporaciones y del gobierno. Esos programas orientados a clientes específicos eran muy caros, sin que se partiera de que se vendían los mismos y mucho menos conceptualizados como software.

- **Productos de software corporativo.**

A mediados de los años 1960 aparece esta forma de comercialización a partir del lanzamiento de las computadoras centrales del Sistema IBM 360 –de tercera generación; en este caso un producto de software era un programa que podía ser utilizado por un gran número de usuarios o clientes corporativos sin necesidad de ser modificado, o con relativas pocas modificaciones o adecuaciones.

- **Comercialización del software orientado al mercado en masa o masivo.**

Este tipo de comercialización de software comenzó a aparecer a mediados de los 1970 con el surgimiento de las microcomputadoras. Esta manera de comercialización o entrega es conocida como software empaquetado, vendido en el comercio minorista o por solicitudes por correspondencia y entrega a domicilio o mediante comercio electrónico.

Esta manera de denominar a los sectores que se mencionan anteriormente es un poco polémica pues a lo largo del periodo de desarrollo de la industria del software existe un poco de irregularidad en la terminología porque los tres sectores han continuado floreciendo desde su aparición o inserción en el mercado y han adoptado la terminología preferida o dominante en cada momento o época.

Anexos del documento

Anexo 1

An Open Letter to Hobbyists

By William Henry Gates III

February 3, 1976

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 on 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 to 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

General Partner, Micro-Soft

Anexo 2

Tabla de compatibilidad entre licencias de SWL.

Name	Versions	Is Copyleft?	GPL compatible?	Used by	Notes
Academic Free License	1.1, 2.1	No	No		
Affero General Public License	1	Yes	No		Version 3 of the AGPL is compatible with the GPL version 3.
GNU Affero General Public License	3	Yes	Yes - GPLv3		Compatible with GPL version 3 only.
Apache License	1.0, 1.1, 2.0	No	Yes - GPLv3	Apache	Versions 1.0 and 1.1 of the Apache License are not GPL compatible
Apple Public Source License	2	No	No	Apple's Darwin operating system	
Arphic Public License			No		
Artistic License	2.0		Yes	Perl	According to the FSF only version 2 of the Artistic License is compatible with the GPL
Berkeley Database License			Yes	Berkeley DB	Also known as the Sleepycat Software Product License.
Boost Software		No	Yes		

License					
CeCILL	2		Yes		GPL compatible since version 2
Clarified Artistic License			Yes		
Common Development and Distribution License		Yes	No	Sun's OpenSolaris	
Common Public License	1.0		No		
Cryptix General License		No	Yes		
EU DataGrid Software License		No	Yes		
Eclipse Public License	1.0		No		
eCos license	2.0		Yes		GPL compatible since Version 2
Eiffel Forum License	2	No	Yes		GPL compatible since Version 2
Expat License		No	Yes	Expat	Also known as the MIT License
GNU General Public	1, 2, 3	Yes	Yes	GNU, Linux kernel,	

License				MySQL	
GNU Lesser General Public License	2, 2.1, 3	Yes	Yes	glibc and many other libraries, themes and icons	
IBM Public License	1.0		No		
Intel Open Source License (OSI)			Yes		As published by OSI.
Interbase Public License	1.0		No		
Jabber Open Source License	1.0		No		This license has been voluntarily retired from OSI.
LaTeX Project Public License	1.2, 1.3a		No	LaTeX	
GNAT Modified General Public License			Yes	GNAT runtime toolkit	
License of Netscape Javascript			Yes		
Vim	6.1 and later		Yes	Vim	
zlib license			Yes	zlib	Also known as the libpng

					license.
License of the iMatix Standard Function Library			Yes		
License of xinetd		Yes	No	xinetd	
Lucent Public License	1.02		No	Plan 9	Also known as the Plan 9 License .
MIT License		No	Yes	X.org	Also known as the X11 License .
Modified BSD license		No	Yes	FreeBSD and OpenBSD	
Mozilla Public License (MPL)			No	Mozilla and Mozilla Firefox	
Netizen Open Source License	1.0		No		
Netscape Public License			No		
Nokia Open Source License			No		
Open Software License	1.0	?	No		
OpenLDAP	2.3, 2.7		Yes	OpenLDAP	GPL compatible since

License					version 2.7
OpenSSL license			No	OpenSSL	
Original BSD license		No	No	NetBSD	
PHP License	3.0		No	PHP	
Phorum License	2.0		No		
Public Domain	-	No	Yes	-	Public Domain isn't truly a license, per se. The definition of what is in the public domain varies from country to country, as is whether or not the author of a work can place his/her work voluntarily in the public domain. If a work is in the public domain, no-one owns the copyright to the work, and/or (usually) anyone can use, copy, modify, redistribute, or otherwise exploit the work for any purpose.
Python Software Foundation License	1.6a2, 1.6b1 - 2.0/2.1, 2.0.1/2.1.1		Yes	Python	This license replaces the original Python License which was not GPL compatible
Q Public License	1.0	No	No	Qt	It was used until Qt 3.0, as Trolltech toolkit version 4.0 was released under GPL version 2.
Standard ML of New Jersey		No	Yes		

Copyright License					
Sun Industry Standards Source License	1.0		No		Sun has voluntarily retired this license.
Sun Public License			No		This license seems now to have been "dropped" by Sun in favor of the new CDDL license, which is also derived from the MPL.
Condor Public License			No		
Ruby License			Yes	Ruby	
Vita Nuova Liberal Source License		Yes	No		
W3C Software Notice and License			Yes	libwww	
X11 License		No	Yes	X.org	This is the FSF's preferred name for the MIT License.
XFree86 1.1 License	1.1	No	No	XFree86	The XFree86 Project contests GPL incompatibility.
Zend License	2.0	No	No	Parts of PHP	
Zope Public License	1, 2.0	No	Yes	Zope	GPL compatible since version 2.0

Anexo 3

Encuesta

Trabajo de Diploma: Contribución a la Epistemología del Software Libre

El presente cuestionario tiene por objetivo contribuir a conocer el grado de conocimiento sobre algunos aspectos generales sobre el Software Libre y el Código Abierto en los diferentes grupos de personas de la universidad. Se le ruega que lea bien todas las preguntas y que responda no por lógica, sino sencillamente el concepto que tiene sobre lo preguntado. En caso de dudas es preferible que seleccione No sé X como respuesta. Sus respuestas serán de gran ayuda para la investigación que se realiza. Los resultados se mantendrán en anonimato, pero si lo desea puede ofrecer los datos siguientes:

Datos del Encuestado (Opcional)

Profesión: _____

Cargo que desempeña: _____

Años de experiencia: _____

PREGUNTAS

1.- ¿En qué década apareció el concepto de software como producto?

(Seleccione con una X):

1950___ 1960___ 1970___ 1980___ Posterior a 1980___ No sé___

2.- La primera forma de producir y distribuir software fue: (Seleccione con una X)

Propietaria___ Colaborativa___ Libre___ No sé___

3.- A su juicio, que fenómeno incentivó con más fuerza el software propietario y la no entrega del código fuente: (Seleccione con una X)

___ El uso por el gobierno y las grandes empresas.

___ El uso por las universidades y muchas empresas.

___ El uso generalizado por profesionales de diversos sectores.

___ No sé.

4.- El software propietario, además de venderse mediante licencias muy restrictivas, las empresas dueñas del mismo lo protegen mediante: (Seleccione con una X):

Patentes___ Marcas___ Secreto comercial___ Derecho de autor___ Derecho de copia___
Una combinación de las anteriores___ No sé___

Añada cualquier aclaración que desee hacer respecto a esta pregunta: _____

5.- ¿El Software Libre es gratis? (Seleccione con una X):

Sí ___ No ___ No sé ___

6.- Todo tipo de software debe protegerse mediante derecho de autor o derecho de copia, independientemente de que sea libre, código abierto o propietario (Seleccione con una X):

Sí ___ No ___ No sé ___

7.- Se habla de Software Libre y también de Código Abierto (Open Source). En inglés últimamente se usan las siglas FOSS para identificar "Free and Open Source Software".

¿Para usted Free Software y Open Source es lo mismo? (Seleccione con una X):

Sí ___ No ___ No sé ___

¿Los dos se venden?

Sí ___ No ___ No sé ___

¿Cuál de los dos es más Copyleft?

El Software Libre ___ El Open Source ___ No sé ___

8.- Si usted tiene dudas sobre como proteger legalmente su proyecto de software, ¿a quien consultaría? (Seleccione con una X o especifique):

A la FSF ___ Al SFLC ___ A la IP de la UCI ___ No sé ___ Otro (Especificar) _____

Significado de las siglas anteriores:

FSF (Free Software Foundation)

SFLC (Software Freedom Law Center)

IP (Infraestructura Productiva)

9.- ¿Disponer del código fuente te autoriza a hacer libremente lo que desees con ese software?

Sí ___ No ___ No sé ___

10.- Se conocen diversas licencias para el **software no propietario**; para usted, ¿Cuál es la más restrictiva? (Seleccione con una X o especifique):

La BSD ___ La AGPL ___ La GPL ___ La Apache ___ La ISC ___ La EULA ___

Otra, menciónela: _____ No sé ___

11.- De las anteriores, ¿cuál es la más socializadora? (Seleccione con una X o especifique):

La ISC ___ La EULA ___ La AGPL ___ La BSD ___ La Apache ___ La GPL ___

No sé ___ Otras, menciónelas: _____

12.- Cuando usted está organizando su proyecto de software y su objetivo final es que resulte Software Libre, ¿Puede utilizar productos de diversas licencias no propietarias?

Sí ___ No ___ No sé ___

13.- En el caso que seleccionó **Sí** o **No** como respuesta a la pregunta anterior, explique brevemente su elección: _____
