

**UNIVERSIDAD DE PANAMÁ
CENTRO REGIONAL UNIVERSITARIO DE VERAGUAS**

MOMENTO CIENTÍFICO-CULTURAL REGIONAL

PONENCIA: SOFTWARE LIBRE VS. SOFTWARE PROPIETARIO:
VENTAJAS, DESVENTAJAS, DESAFÍOS Y
OPORTUNIDADES.

EXPOSITOR: RAÚL ENRIQUE DUTARI DUTARI.

FECHA: 28 DE SEPTIEMBRE DE 2009.

HORA: 10:30 A. M.

LUGAR: LABORATORIO A-05 DEL CENTRO REGIONAL
UNIVERSITARIO DE VERAGUAS.

DIRIGIDA A: PROFESORES UNIVERSITARIOS, PROFESIONALES Y
ESTUDIANTES QUE PARTICIPARON EN EL EVENTO.

DURACIÓN: 15 MINUTOS.

OBJETIVO GENERAL

1. Debatir las fortalezas y debilidades que evidencia el software libre frente al software propietario.

OBJETIVOS ESPECÍFICOS

1. Contrastar las tendencias de software propietaria y libre, en términos de: el costo de la tecnología, su dificultad de uso en el computador, así como su difusión entre el público en general.
2. Establecer el significado de la terminología usualmente empleada en la industria del software propietario y libre.
3. Contraponer los desafíos y las oportunidades más relevantes que presenta cada paradigma de software.
4. Comparar algunos tipos de herramientas, en sus versiones propietaria y libre.

RESUMEN DE LA PONENCIA

En la actualidad, se observan dos movimientos bien marcados de desarrollo de software.

Por un lado, se encuentra el “software propietario” que se ampara bajo la legislación de derecho de autor y una interpretación muy particular del concepto de “compatibilidad”, más una serie de prácticas monopólicas dirigidas a incrementar su cuota de mercado en la industria informática. Este paradigma obliga a sus usuarios a pagar sumas importantes de dinero por el derecho “restringido” de utilizar sus soluciones, bajo los términos de unos acuerdo de licencia sumamente restrictivos. Además, fomenta el uso ilegal del software.

Por otro lado, se encuentran las soluciones basadas en el software libre, que ofrecen a los usuarios la posibilidad de elegir las aplicaciones que utilizarán en sus sistemas. Por lo general las soluciones basadas en este paradigma, tienen un costo nulo o muy bajo y pueden resolver, razonablemente bien, la mayoría de los problemas que se enfrentan dentro de campo de la informática.

Es cuestión del usuario el decidir con cuál de los dos paradigmas resuelve sus problemas de procesamiento de información.

TABLA DE CONTENIDOS

Resumen De La Ponencia.....	iv
1. Observaciones Preliminares.	1
2. Algo De Terminología.	1
3. La Historia De Los Paradigmas Enfrentados.	3
4. Características Principales De Cada Paradigma.	9
4.1. Características Principales De Los Sistemas Propietarios.....	9
4.2. Características Principales De Los Sistemas Libres.....	10
5. Ventajas Y Desventajas De Cada Paradigma.....	11
5.1. Ventajas Y Desventajas Del Software Libre.....	11
5.2. Ventajas Y Desventajas Del Software Propietario.	14
6. Desafíos Y Oportunidades Que Ofrece Cada Paradigma.....	17
6.1. Software Libre.....	17
6.2. Software Propietario.....	18
7. Comparación De Algunos Tipos De Herramientas Informáticas, En Sus Versiones Propietaria Y Libre.....	20

8.	Conclusiones.....	22
9.	Referencias Bibliográficas.....	23

1. OBSERVACIONES PRELIMINARES.

En el mundo del software, se observan dos tendencias bien marcadas que se observan en los sistemas actuales. Presentan una retórica similar a la de una guerra, ya que representan paradigmas muy diferentes.

Por un lado, se observan los sistemas propietarios, por los cuales se debe pagar importantes sumas de dinero, y de no hacerlo, se incurre en delitos; además de restringir ampliamente lo que el usuario puede hacer con el software. Están ampliamente difundidos en el medio local, nacional y mundial.

Por otro lado, se encuentran los sistemas libres, que no requieren de pago por su uso; y por añadidura, ofrece una amplia libertad al usuario para hacer lo que desee con el programa; no se encuentran tan difundidos, aunque su presencia en el mundo del software se ha incrementado significativamente en los últimos años.

Esta ponencia pretende contrastar ambas corrientes, contrastando las fortalezas y debilidades de cada filosofía de desarrollo.

Se espera que el auditorio pueda evaluar, objetivamente, ambas corrientes de sistemas, para que utilicen la que más se ajuste a sus necesidades.

2. ALGO DE TERMINOLOGÍA.

A continuación, se enunciarán algunos conceptos clave que deben estar claros para la correcta comprensión de esta ponencia.

- **Software:** es un programa informático que, al ser ejecutado en un computador, realiza una tarea predeterminada. A nivel de algoritmo, es un conjunto de instrucciones que resuelve, de manera única y sin ambigüedades, un problema particular **[SMAL06]**.
- **Lenguaje de programación:** es un conjunto de instrucciones y reglas pre establecidas que permiten crear programas informáticos, y que puede comprender el ser humano con relativa facilidad. Ejemplos de lenguajes de programación son, entre otros: Java, PHP, HTML, C /C++/C#, y otros más **[SMAL06]**.
- **Código fuente:** son el conjunto de instrucciones, dadas en un lenguaje de programación específico, que constituyen a un software en particular. Estas instrucciones pueden ser comprendidas de manera relativamente simple por el ser humano. Sin tener acceso a él, no es posible modificar o mejorar el software, aunque se conozca claramente lo que se desea cambiar **[SIGA06]**.
- **Código objeto:** es un conjunto de instrucciones que ha sido analizado cuidadosamente por un conjunto de programas especial, llamado compilador, pero que no puede ser ejecutado directamente por el computador. No puede ser comprendido directamente por el ser humano **[SIGA06]**.
- **Código ejecutable:** también es conocido como programa ejecutable. Son las instrucciones que interpreta directamente el computador. Tampoco puede ser comprendido fácilmente por el ser humano **[SIGA06]**.
- **Sistema operativo:** es un conjunto de programas que permite utilizar un computador sin ser un programador experto. Adicionalmente, se encarga

de administrar los recursos de hardware y software del computador [SIGA06].

- **Licencia:** es un contrato o acuerdo legal que establece derechos y deberes que deben respetar los fabricantes de software, así como sus usuarios. Normalmente, están orientadas a la restricción de los derechos de los usuarios a utilizar el software únicamente dentro de ciertos criterios pre establecidos por quienes ofrecen la licencia [CUGO06].
- **Software Libre:** Es aquel que respeta la libertad de los usuarios sobre el producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente en [WIKI10A].
- **Software Propietario:** Por el contrario, es cualquier programa informático en el que el usuario final tiene limitaciones para usarlo, modificarlo o redistribuirlo - con o sin modificaciones -, o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido por un acuerdo de licencia, o por tecnología anticopia [WIKI10B].

3. LA HISTORIA DE LOS PARADIGMAS ENFRENTADOS.

Aunque parezca ilógico, originalmente el desarrollo de software se inició bajo el paradigma del software libre, en la década de los 50. En esta época casi todo el software era producido en masa por académicos e investigadores corporativos en colaboración y no era visto así mismo como un producto [WIKI10C].

Los sistemas operativos eran ampliamente distribuidos y mantenidos por comunidades de usuarios. El código fuente, la versión leíble para humanos del software, era distribuido con el software porque los usuarios modificaban el

software ellos mismos para arreglar errores de programación o agregar nuevas funcionalidades.

Así en esa era, el software era libre en cierto sentido, no por algún esfuerzo realizado por usuarios o desarrolladores de software, sino porque el software era desarrollado por la comunidad de usuarios.

Sin embargo, para finales de los años 1960 se empezó a producir un cambio: al evolucionar los sistemas operativos y los compiladores de lenguajes de programación, los costos del software de las compañías se incrementaron dramáticamente.

Una naciente, aunque creciente industria del software estaba compitiendo con el software incluido al hardware de los fabricantes - el costo de los programas precargados estaba incluido en el precio del hardware -, las máquinas arrendadas requerían soporte del software al tiempo que no se proveía la renta del software, y algunos clientes capaces de satisfacer de mejor manera sus propias necesidades no quería que el costo del software del fabricante fuera incluido con el costo del producto **[WIKI10C]**.

En los años 1970 AT&T distribuía las primeras versiones de UNIX sin costo al gobierno y a investigadores académicos, pero estas versiones no permitían su distribución ni la distribución de versiones modificadas, y por ende no eran software libre en la acepción moderna de la frase .

A finales de los años 1970 y principios de los años 1980, los vendedores de computadoras y compañías de software empezaron a cobrar por licencias de software de manera rutinaria, comercializándolas como “Productos Informáticos” e imponiendo restricciones legales a los nuevos desarrollos de software, ahora

vistos como activos, a través de derechos de autor, marcas registradas y contratos de arrendamiento.

Finalmente, en 1976, Bill Gates marcó el gran cambio de era cuándo escribió, su ahora famosa Carta abierta a los aficionados, mandando el mensaje de que lo que los hackers llaman “compartir” era, en sus palabras, “robar”, lo que provocó que una gran parte del software desarrollado a partir de ese momento, se vendiera como un producto, un empresas tales como: Microsoft, Oracle, Symantec Adobe, Macromedia, y muchas otras, bajo la forma de software propietario **[WIKI10C]**.

Sin embargo, el avance del software propietario no significó que el software libre desapareciera. Así por ejemplo, en 1979, cuando AT&T empezó a hacer cumplir sus licencias restrictivas cuando la compañía decidió que podrían generar utilidades vendiendo el sistema Unix, lo que a su vez, posteriormente originó el desarrollo de sistemas y Unix compatibles libres, entre los que destacan GNU Linux, Free-BDS, entre otros.

La llegada de Usenet al principio de los años 1980 conectó aún más a la comunidad de programadores y facilitó, a los programadores, una forma más simple de compartir su software y de contribuir al software que otros hayan escrito, lo que fortaleció a lo que ya empezaba denominarse como “movimiento de software libre”.

Mucho software libre que fue desarrollado en los años 1970 y en los años 1980 aún continúa siendo utilizado, incluyendo el simulador de circuitos integrados SPICE, TeX (desarrollado por Donald Knuth) y el sistema de ventanas X, entre otros.

En 1983, Richard Stallman lanzó el proyecto GNU para escribir un sistema operativo completo libre de restricciones sobre el uso de su código fuente. Uno de los incidentes particulares que lo motivaron a esto fue el caso de una molesta impresora que no podía ser arreglado porque el código fuente no era revelado **[WIKI10C]**.

Otro posible evento de inspiración para el proyecto GNU y su manifiesto fue el desacuerdo entre Stallman y Symbolics, Inc. sobre el acceso a las actualizaciones, por parte del MIT, que Symbolics había realizado a su máquina Lisp, la cual estaba basada en código del MIT. Poco tiempo después de su lanzamiento, acuñó el término “software libre” y para promover el concepto fundó la Free Software Foundation. Una definición de software libre fue publicada en febrero de 1986.

En 1989, fue publicada la primera versión de la Licencia Pública General de GNU. En 1991 se publicó la ligeramente actualizada versión 2 de la licencia. Vale En 1989, algunos desarrolladores de GNU crearon la compañía Cygnus Solutions, responsable de muchos desarrollos claves dentro de los proyectos de software libre **[WIKI10C]**.

El núcleo (kernel) del proyecto GNU, posteriormente llamado “GNU Hurd”, fue retrasado continuamente, pero la mayor parte de los demás componentes fueron completados para 1991. Algunos de éstos, especialmente la Colección de compiladores de GNU, se han convertido en líderes del mercado por méritos propios. El Depurador de GNU y GNU Emacs también fueron éxitos notables.

Más tarde, en la segunda mitad de los años 1990, cuándo muchas compañías basadas en Web empezaban a operar; el software libre se volvió la opción popular de los servidores web. El servidor HTTP Apache se volvió en el software

de servidor Web más utilizado, título que aún conserva a finales de la primera década del siglo XXI [WIKI10C].

Estos servidores Web, se han llegado a conocer como “sistemas LAMP”; son sistemas que vienen con un “paquete” de software inicial que incluye al núcleo **Linux** como base, a **Apache** como proveedor de servicios Web, la base de datos **MySQL** para archivo de datos y al lenguaje de programación **PHP** uniéndolos a todos. En el medio local, el WAMP es ampliamente utilizado bajo la plataforma Windows.

En 1997, Eric Raymond publicó *La catedral y el bazar*, un ensayo dónde analiza a la cultura hacker y los principios del software libre. El escrito recibió particular atención al principio de 1998 y fue un factor estimulante para que Netscape Communications Corporation liberara su popular suite de Internet Netscape Communicator como software libre. Este código, es mejor conocido en la actualidad como Firefox y Thunderbird.

Este acto por parte de Netscape, impulsó a Raymond y otros a buscar cómo llevar los principios del software libre a la industria del software comercial. Concluyeron que el activismo social de la FSF no era atractivo para las compañías como Netscape y buscaron maneras de volver a inventar el movimiento del software libre para enfatizar el potencial comercial de compartir código fuente [WIKI10C].

El nuevo nombre que escogieron fue “código abierto” (*open source* en inglés) y rápidamente Bruce Perens, el editor Tim O'Reilly, Linus Torvalds y otros firmaron a favor del nuevo nombre. La Open Source Initiative (Iniciativa por el Código Abierto) fue fundada en febrero de 1998 para fomentar el nuevo término y para difundir los principios del “código abierto”.

Sin embargo, Richard Stallman y la FSF objetaron enérgicamente al enfoque de la nueva organización. Sintieron que, con la estrecha visión del código fuente, la OSI estaba sepultando los valores filosóficos y sociales del software libre y que estaba escondiendo el problema de la libertad de los usuarios de computadoras. Aún así, Stallman seguía manteniendo la postura de que los usuarios de ambos términos eran aliados en contra del software propietario.

En septiembre de 2000, Sun Microsystems lanzó la suite ofimática StarOffice 5.2 como software libre bajo los términos de la Licencia Pública General Reducida de GNU. La versión libre de la suite fue renombrada a OpenOffice.org (comúnmente abreviado “OOo”) y coexiste con StarOffice, la versión 1.0 de OpenOffice.org fue lanzada el 30 de abril de 2002; el producto y el código fuente abierto coexisten desde el 13 de octubre de 2000 **[WIKI10C]**.

Los inicios del siglo 21 se muestran muy controversiales, ya que empresas como Sun Microsystems, transnacionales del software y del hardware, se dedican a adquirir empresas de software libre, tales como MySQL AB, con su base de datos MySQL, Innotek GmbH, con su motor de Virtualización Virtual Box con lo que expanden su presencia en el mercado en hardware y del software. No obstante Sun Microsystems ha hablado continuidad a todos los proyectos de software libre previamente adquiridos **[WIKI10C]**.

Finalmente, a inicios del el año 2010, la Empresa Sun Microsystems fue a su vez adquirida por Oracle Corporation, lo que lleva a una posición incierta a toda la cartera de productos de software libre que manejaba dicha empresa.

4. CARACTERÍSTICAS PRINCIPALES DE CADA PARADIGMA.

A continuación, se canalizarán las principales características que presentan cada uno de los paradigmas analizados.

4.1. CARACTERÍSTICAS PRINCIPALES DE LOS SISTEMAS PROPIETARIOS.

De todo lo dialogado previamente, se pueden reducir las características primordiales del software propietario, que se plantean a continuación **[SMAL06]**:

- **Es propiedad de una persona natural o jurídica:** en consecuencia se considera un bien o activo, económicamente hablando.
- **Sea alquila su uso:** No se puede comprar, ya que la compra significaría renuncia a la propiedad, algo a lo que no están dispuestas las empresas que lucran con este tipo de producto. Su comercialización está controlada fuertemente por las leyes de derecho de autor, patentes y licencia de uso.
- **Se distribuyen únicamente en código objeto:** es decir, no es posible modificar, corregir o mejorar su funcionalidad fácilmente, ya que su código fuente es un secreto comercial de la empresa u organización que lo posee.
- **No se puede modificar a nivel de código fuente, sin cometer delitos:** ya que esta es una de las acciones que expresamente se prohíbe dentro de los acuerdos de licencia de uso y las leyes de derecho de autor.

En fin, su uso, depuración y mejora está fuertemente controlado por parte de terceros.

4.2. CARACTERÍSTICAS PRINCIPALES DE LOS SISTEMAS LIBRES.

En contraposición, software libre se caracteriza por las cuatro libertades básicas del software libre, a las que todos los proyectos de software que se autodenominan libres, deben acogerse obligatoriamente [SMAL06]:

- **Libertad de ejecutar el programa con cualquier propósito:** a diferencia del software propietario, el usuario del software libre puede utilizarlo como mejor le parezca, sin limitaciones preestablecidas por acuerdos de licencia y patentes.
- **Libertad de estudiar cómo funcionan programa y de adaptarlo a sus necesidades:** ya que no se comercializa, sino que se distribuye, todos los usuarios tienen la posibilidad de revisar sus instrucciones y mejorarlas y corregirlas, para adaptarlo a sus necesidades particulares.
- **Libertad de redistribuir copias del programa:** se debe entender en consecuencia, que cualquier usuario o sea una copia de un software denominado libre, puede transferirla, donarla, a quien mejor le parezca, sin que esta acción implique un delito.
- **Libertad de mejorar el programa y redistribuir dichas modificaciones:** es decir, las modificaciones que introduce un usuario grupo de usuarios en código fuente del software libre con la intención de

mejorarlo o corregirlo, deben ser puestas a disposición de la comunidad de software libre.

5. VENTAJAS Y DESVENTAJAS DE CADA PARADIGMA.

Cada uno de estos paradigmas de desarrollo de software, conlleva una serie de ventajas desventajas intrínsecas, que se analizan a continuación.

5.1. VENTAJAS Y DESVENTAJAS DEL SOFTWARE LIBRE.

El software libre ofrece una serie de ventajas y desventajas que se consideran a continuación **[CUGO06]**:

- **Libre de costo:** El bajo o nulo coste de los productos libres permiten proporcionar, a las pequeñas y medianas empresas, nuevos servicios y ampliar sus infraestructuras sin que se vean mermados sus intentos de crecimiento por no poder hacer frente al pago de grandes cantidades en licencias. Esta misma situación, se puede extrapolar hacia el usuario doméstico.
- **Soporte brindado por la comunidad libre:** a diferencia de la mayoría de las empresas que se dedican a desarrollar software - que cobran por el servicio de soporte -, muchos sistema de software libre reciben el soporte por parte de grupos de usuarios que, desinteresadamente, ofrecen sus conocimientos a la comunidad de usuarios en general, por lo general un costo. Esto no impide, que se formen empresas especializadas en ofrecer soporte técnico para este tipo de producto.

- **Falta de garantías:** usualmente este es un argumento muy utilizado por las compañías de software propietario, que alegan la falta de garantía del funcionamiento de los sistemas libres. Sin embargo, al leer cuidadosamente los acuerdos de licencia de los sistemas propietarios, se observa que tampoco las ofrecen. En general las empresas productoras de Software Propietario no ofrecen ningún tipo de garantía por sus productos. Además en materia de soporte técnico, presentan muchas deficiencias pues este es brindado muchas veces por quienes no tienen la posibilidad real de solucionar problemas. En el caso del Software Libre, la situación es radicalmente distinta. Aunque los creadores originales en general tampoco ofrecen ninguna garantía por el producto, la misma (junto con el soporte) puede ser asumida por una empresa que elija brindar dicho servicio a sus clientes. Dicha situación sería posible ya que se dispondría del código fuente del programa en cuestión, junto con el derecho de efectuar las modificaciones que el cliente requiriera para la satisfacción de sus necesidades, con lo cual estaría en posición de ofrecer un servicio garantía y asistencia técnica de gran calidad, como efectivamente se presenta en muchas situaciones.
- **Libre modificación del código fuente:** al estar disponible el código fuente de estos sistemas, significa que, los usuarios con los conocimientos técnicos suficientes pueden modificar, a su libre conveniencia, los sistemas. De esta manera, es sumamente difícil que en un sistema de este tipo se encuentren situaciones tales como: puertas traseras, contraseñas maestras, o cualquier otro tipo de implementación de código que permita violar la seguridad del sistema por parte de terceros.

- **Protegido por las leyes de derecho de copia (Copyleft):** en este tipo de sistema, por las cuatro libertades antes señaladas, se garantiza de manera explícita a los usuarios que pueden realizar todas las copias y redistribución que consideren necesarias de los medios de distribución del software.
- **Ofrecen una amplia cantidad de herramientas para todo tipo de necesidades:** al contrario de lo que mucha gente piensa, en el mundo del software libre existen soluciones informáticas, equivalentes a la soluciones propietarias, para prácticamente todos los problemas que se requieren resolver. Son muy contadas las áreas de especialidad donde el sobre libre no se ha podido desarrollar a nivel de poder competir con los productos propietarios.
- **Corrección más rápida y eficiente de fallos:** El funcionamiento e interés conjunto de la comunidad ha demostrado solucionar mas rápidamente los fallos de seguridad en el software libre, algo que desgraciadamente en el software propietario es mas difícil y costoso. Cuando se notifica a las empresas propietarias del software, éstas niegan inicialmente la existencia de dichos fallos por cuestiones de imagen y cuando finalmente admiten la existencia de esos errores de programación tardan meses en resolverse completamente, para proporcionar los parches de seguridad.
- **Fomento de la libre competencia al basarse en servicios y no licencias:** Uno de los modelos de negocio alternos que genera el software libre es la contratación de servicios de atención al cliente. Este sistema permite que las compañías que den el servicio compitan en igualdad de condiciones al no poseer la propiedad del producto del cual dan el servicio. Esto, además, produce un cambio que redundará en una

mayor atención al cliente y contratación de empleados, en contraposición a sistemas mayoritariamente sostenidos por la venta de licencias y desatención del cliente.

- **Soporte limitado para el hardware y los periféricos modernos:** esta situación se presenta ya que la mayoría de los controladores de dispositivos desarrollados en el mundo del software libre, se implementan después que el hardware es puesto a la venta pública. Sin embargo, con algo de paciencia, se observa que: a la larga, el soporte de hardware y periféricos se desarrolla eventualmente para la mayoría de los productos existentes en el mercado.
- **Aplicaciones menos acabadas:** dado que son productos de bajo un nulo costo a sus usuarios, los desarrolladores de ellos generalmente no se preocupan tanto por los refinamientos estéticos de estos productos. En su lugar, se concentran más en ampliar su rendimiento y funcionalidad.

5.2. VENTAJAS Y DESVENTAJAS DEL SOFTWARE PROPIETARIO.

A su vez, el software propietario evidencia una serie de ventajas y desventajas, generalmente contrapuestas, frente a los sistemas libres. Dicha ventajas Y desventajas se analizan a continuación [CUGO06]:

- **Se debe pagar por usarlo:** es quizás, la principal de ventajas de este tipo de producto. Son relativamente pocos los casos donde se encuentran productos gratuitos desarrollados bajo sistemas propietarios. La mayoría de las veces, los productos de software tienen un costo significativamente elevado.

- **Soporte para todo tipo de hardware:** tomando como referencia en el mercado a los productos como Microsoft Windows, y no al resto de sistemas operativos de tipo Unix, que es minoritario: este soporte se da. Sin embargo, esta situación tiene sus matices. Por un lado, Microsoft "invita" a los fabricantes de dispositivos para ordenadores personales a producir drivers o hardware solo compatible con Windows. En consecuencia, la elección del sistema operativo de Microsoft tiene garantizado un soporte de hardware seguro. Aunque siendo justos, se debe señalar que dicho soporte por parte de los fabricantes se limita generalmente al sistema operativo a la venta. Los sistemas operativos que su propio fabricantes define como "descontinuados", no reciben el mismo tratamiento.
- **Soporte brindado por los desarrolladores:** los desarrolladores de productos propietarios, normalmente tienen personal dedicado al brindar soporte técnico a los usuarios registrados de sus productos. Esto servicios, en muchos casos, se facturan al cliente independientemente de la compra del sobre, previa comprobación de que el - el cliente - adquirió legalmente el software. Sin embargo, en la práctica suelen ser más efectivos los apoyos logrados vía foros de discusión, blogs, así como otros recursos gratuitos disponibles en Internet.
- **Falta de garantías:** con tanta insistencia como en el mundo del software libre, las empresas que comercializan software propietario, se liberan de toda responsabilidad ante cualquier fallo del producto, que represente para el usuario, pérdidas materiales.
- **Dependencia del proveedor:** Como se ha mencionado anteriormente, la corrección de errores o el agregado de nuevas funciones en un programa solamente puede hacerse si se dispone del código fuente. Es claro que, al

ser la empresa proveedora la única que dispone de dicho código, sólo esta puede atender a los requerimientos de un cliente insatisfecho con el producto del cual ha adquirido una licencia de uso. Esto pone al usuario en una clara situación de dependencia del proveedor. Si el programa presenta algún defecto, éste debe aceptar las condiciones de la empresa productora del programa (en el supuesto caso de que dicha empresa reconozca el error y acceda a repararlo).

- **Protegido por las leyes de derecho de autor (Copyright):** las leyes de derecho de autor y copyright constituyen dos concepciones sobre la propiedad literaria y artística. El derecho de autor se basa en la idea de un derecho personal del autor, fundado en una forma de identidad entre el autor y su creación. El derecho moral está constituido como emanación de la persona del autor; reconoce que la obra es expresión de la persona del autor y así se le protege. Este tipo de legislación ha sido impuesto por las compañías que explotan financieramente este recurso. En pocas palabras, quienes utilizan el software, deben no sólo pagar la compra del producto, también deben por las regalías a las que tiene derecho el un los autores del producto, por efecto de las leyes de derecho de autor, lo que claramente represente una desventaja económica para sus usuarios.
- **Sujeto a prácticas monopólicas:** el hecho que una única empresa sea la que posee el código fuente del producto software, condiciona a que los usuarios deben aceptar forzosamente las condiciones de uso que imponga dicha empresa; situación que claramente caracteriza las prácticas monopólicas.

6. DESAFÍOS Y OPORTUNIDADES QUE OFRECE CADA PARADIGMA.

Ambos paradigmas - el libre y el propietario -, se enfrentan a desafíos que deben superar en la medida en que evolucionan; así como se les presentan oportunidades potenciales que les pueden permitir evolucionar con ventaja frente a la competencia. Algunos de estos desafíos Y oportunidades se analizan a continuación [CUGO06]:

6.1. SOFTWARE LIBRE.

El software libre se encuentra ante toda una serie de oportunidades reales que permiten que se incremente en presencia en el mercado informático. Entre ellas destacan:

- **Se presta para ser distribuido de manera gratuita y cobrar por el soporte:** lo que representa un modelo de negocios bastante justo.
- **Puede ser adaptado a múltiples escenarios, originalmente no previstos por sus autores:** lo que le brinda una gran versatilidad.
- **Permite que múltiples empresas ofrezcan los mismos servicios de soporte:** de esta manera, se fomenta la competencia real entre las empresas en función a los servicios que ofrecen.
- **Dificulta significativamente la aparición de puertas traseras y similares; y facilita enormemente su eliminación:** ya que cualquier intento alterarlo en este sentido, se ha detectado y corregido rápidamente.

Por otro lado, el software libre se enfrenta a un gran desafío:

- **Aún se encuentran demasiadas empresas que desconfían de este modelo de desarrollo informático:** deberán pasar muchos años de desarrollo sostenido de los proyectos de sobre libre para que muchas empresas importantes, sobre todo de los sectores industriales, se vuelcan plenamente hacia los productos de software libre.

6.2. SOFTWARE PROPIETARIO.

El software libre se encuentra ante toda una serie de desafíos reales que deben superar si se desea que a largo plazo se mantenga como modelo viable de comercialización, ya que [CUGO06]:

- **Se cobra por la venta y por el soporte:** esta es una situación que, a todas luces, resulta molesta a los clientes, sobre todo, a los de recursos limitados y en donde el software libre actúa con ventaja.
- **Se debe utilizar, estrictamente, en el entorno para el que fue diseñado:** muchos paquetes propietarios, se están autoexcluyendo de explotar mercados alternos, ya que no implementan versiones de sus sistemas en otras plataformas que no sean las más populares - es decir, Microsoft -. Esta una tendencia que ya se observa que está alterando significativamente, a favor de la diversificación de plataformas operativas.
- **Facilita el establecimiento de monopolios comerciales cerrados:** ya que son productos desarrollados sin que el código fuente se encuentre disponible al público, imposibilita que otras empresas implementen a la misma solución, de manera alterna - es decir, versiones alternativas del

mismo paquete implementadas por empresas completamente diferentes -. Adicionalmente, esta práctica violaría abiertamente todos los acuerdos de licencia que generalmente establecen las empresas que originalmente desarrollan sus productos como soluciones propietarios.

- **Posibilita la implantación de puertas traseras, gusanos, troyanos y otros tipos de software maligno:** aunque muchas empresas que desarrollan sobre propietario lo niegan, es una realidad que ellas, con cierta regularidad, realizan esta práctica. Hace algunos años, la empresa Sony, de manera encubierta implementó un sistema de rootkit ese instalaba de manera automática en todos los computadores donde se introducía un CD de audio de esta compañía de determinados artistas. Por un error de programación, este software permitía la inclusión en el sistema de software maligno de muy diversas fuentes [DIT109].
- **Muchas empresas aún confían en este modelo, por encima de sus defectos:** actuando en base aún muy malentendidos sentido de "compatibilidad", una gran cantidad empresas e instituciones públicas y privadas establecen sus políticas de intercambio de información basándose en estándares cerrados, fundamentados en sistema propietarios, y despreciando abiertamente las soluciones que representan los estándares libres. Esta situación, refuerza la posición monopólica de estas empresas, creando un círculo vicioso entre ellas los usuarios de dichas organizaciones, obligando a estos últimos a utilizar las soluciones propietarios, e incurriendo en datos significativos de compra de licencia de software.

7. COMPARACIÓN DE ALGUNOS TIPOS DE HERRAMIENTAS INFORMÁTICAS, EN SUS VERSIONES PROPIETARIA Y LIBRE.

Un argumento muy esgrimido por los defensores del software propietario, consiste en que este paradigma de desarrollo no ofrece suficientes soluciones para los problemas que se deben resolver en la actualidad.

A continuación, se muestra un cuadro en el que aparecen, tipificados por categoría, los sistemas propietarios más importantes de la actualidad, conjuntamente con su equivalente respectivo en software libre **[OFSO10]**:

<i>Tipo</i>	<i>Programa</i>	<i>Equivalente Propietario</i>
<i>Ofimática</i>	OpenOffice	Microsoft Office
<i>Programación</i>	Mono	.NET
<i>Gráficos Vectoriales</i>	Inkscape	Corel Draw / Adobe Freehand
<i>Video</i>	VLC	Windows Media Player
<i>Sonido</i>	XMMS	Windows Media Player
<i>E-Mail</i>	Mozilla ThunderBird	Microsoft Outlook
<i>Navegador Web</i>	Mozilla Firefox	Internet Explorer
<i>Navegador Web</i>	Google Chrome	Internet Explorer
<i>Retoque Fotográfico</i>	The Gimp	Adobe Photoshop
<i>Visión y Edición de Imágenes</i>	Picasa	ACDSee
<i>Mensajería Instantánea</i>	Pidgin	MSN Messenger
<i>Mensajería Instantánea</i>	aMSN	MSN Messenger
<i>Diseño Electrónico</i>	Oregano	PSpice
<i>Lector PDF</i>	Foxit Reader	Adobe Acrobat Reader
<i>Impresora PDF</i>	OpenOffice	Adobe Acrobat Writer
<i>Impresora PDF</i>	Cute PDF Writer	Adobe Acrobat Writer
<i>Grabación CD/DVD</i>	CD Burning	Ahead Nero
<i>Antivirus</i>	Avast!	Panda/Norton
<i>Antivirus</i>	AVG	Panda/Norton
<i>AntiSpyware</i>	Dr.Web	Ad-Aware
<i>AntiSpyware</i>	Spybot Search&Destroy	Ad-Aware
<i>Servidor</i>	Apache	Windows Server
<i>Gestos de Bases de Datos</i>	MySQL	Oracle
<i>Gestor de Bases de Datos</i>	PostgreSQL	Oracle
<i>Lenguaje de Programación Web</i>	PHP	ASP

Cuadro 1: Aplicaciones Libres Y Sus Equivalentes Propietarios

Los renglones más relevantes donde, a la fecha, el software libre no puede competir, con productos relevantes se pueden señalar a:

- Las aplicaciones para desarrollo multimedia.
- Las aplicaciones contables.
- Los juegos para Computadoras.

Es de esperar que a mediano plazo, esta situación se altere, a favor de las soluciones libres.

8. CONCLUSIONES.

Al finalizar esta ponencia, se pueden llegar a las siguientes conclusiones:

- Los sistemas propietarios se presentan como aplicaciones informáticas cerradas, de alto costo y sujetas a las leyes de derecho de autor (copyright).
- Los sistemas libres se plantean como alternativas serias y bien fundadas a los sistemas operativos propietarios, ya que son aplicaciones informáticas abiertas, de costo razonable, así como de libre distribución (copyleft).
- Como toda industria, el software propietario ha establecido una serie de normas y prácticas que obligan a la gran mayoría de los usuarios de computadoras a utilizar estas soluciones, basándose en un mal interpretado principio de "compatibilidad", así como de prácticas monopólicas dirigidas a acentuar su presencia en el mercado informático.

- Para el usuario común del medio local, el uso de los sistemas libres como plataforma de trabajo debería estar limitado, únicamente, por el deseo de utilizarlos, ya que para la mayoría de sus necesidades, existen programas libres que la pueden resolver razonablemente bien.

9. REFERENCIAS BIBLIOGRÁFICAS.

- [CUGO06] CULEBRO JUÁREZ, Monserrat; GÓMEZ HERRERA, Wendy Guadalupe; TORRES SÁNCHEZ, Susana. Software Libre Vs. Software Propietario: Ventajas Y Desventajas. Fecha De Actualización: 2006-05. Fecha de Consulta: 2010-09-28. Disponible en:
http://gnutransfer.info/descargas/software_libre_vs_propietario.pdf.
- [DITI09] DIARIO TECNOLOGÍAS DE LA INFORMACIÓN. Sony Nuevamente Causa Problemas Con Su Rootkit Ilegal. MPA Consulting Group. Fecha De Actualización: 2009-09-15. Fecha de Consulta: 2010-09-28. Disponible en:
<http://www.diarioti.com/gate/n.php?id=24011>.
- [KAPE07] KAPELL G., Abdiel E. Transición De Software Propietario A Software Libre En El C.R.U.V. Trabajo De Graduación Para Optar Por El Título De Licenciatura En Ingeniería En Informática. Universidad De Panamá. 2007.

- [OFSO10] **OFICINA DEL SOFTWARE LIBRE.** Software Libre Vs. Software Propietario. Universidad De Las Palmas De Gran Canaria. Fecha de Actualización: Sin Fecha De Actualización. Fecha de Consulta: 2010-09-28. Disponible en: <http://www.softwarelibre.ulpgc.es/node/56>.
- [SIGA06] **SILBERSHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg.** Fundamentos De Sistemas Operativos. McGraw-Hill, 2006.
- [SMAL06] **SMALDONE, Javier.** Software Libre Vs. Software Propietario. Javier Smaldone. Fecha De Actualización: 2006. Fecha de Consulta: 2010-09-28. Disponible en: <http://www.smaldone.com.ar/opinion/docs/slvssp.pdf>.
- [WIKI10A] **WIKIPEDIA.** Software Libre. Wikimedia Foundation, Inc. Fecha de Actualización: 2010-09-20. Fecha de Consulta: 2010-09-28. Disponible en: http://es.wikipedia.org/wiki/Software_libre.
- [WIKI10B] **WIKIPEDIA.** Software Propietario. Wikimedia Foundation, Inc. Fecha de Actualización: 2010-09-18. Fecha de Consulta: 2010-09-28. Disponible en: http://es.wikipedia.org/wiki/Software_propietario.
- [WIKI10C] **WIKIPEDIA.** Historia Del Software Libre Y De Código Abierto. Wikimedia Foundation, Inc. Fecha de Actualización: 2010-09-16. Fecha de Consulta: 2010-09-28. Disponible en: http://es.wikipedia.org/wiki/Historia_del_software_libre_y_de_c%C3%B3digo_abierto.