



**UNIVERSIDAD INTERNACIONAL DEL ECUADOR  
FACULTAD DE JURISPRUDENCIA**

**TRABAJO DE INVESTIGACION PREVIO A LA OBTENCIÓN DEL  
TÍTULO**

**DE ABOGADA DE LOS TRIBUNALES Y JUZGADOS DEL ECUADOR**

**TEMA**

**SOFTWARE LIBRE O SOFTWARE PROPIETARIO PARA UNA  
EFECTIVA TRANSFERENCIA DE TECNOLOGÍA EN ECUADOR**

**AUTOR:**

**MARICELA SORAYA IBARRA BEDOYA**

**DIRECTOR:**

**AB. DIEGO MORALES OÑATE**

**Ecuador - Quito, 15 mayo del 2016**

## **CERTIFICACION DE DIRECTOR / A**

Por el presente dejo constancia de que el proyecto de investigación, presentado por Maricela Soraya Ibarra Bedoya, para optar por el Título de Abogada de los Tribunales y Juzgados de la República del Ecuador, Bajo el Título de SOFTWARE LIBRE O SOFTWARE PROPIETARIO PARA UNA EFECTIVA TRANSFERENCIA DE TECNOLOGIA EN EL ECUADO, ha sido supervisado.

Certifico que es fruto del autor y puede ser sometido a evaluación por el Comité de Investigación de la Facultad de Ciencias Sociales y Jurisprudencia.

Quito, 15 de Julio del 2016.



Dr. Diego Morales Oñate.  
DIRECTOR (A) DEL TRABAJO DE INVESTIGACION.

## CERTIFICACIÓN

Yo, Diego Morales Oñate certifico que conozco a la autora del presente trabajo siendo responsable exclusivo tanto en su originalidad, autenticidad, como en su contenido.

A handwritten signature in black ink, enclosed within a hand-drawn oval. The signature is stylized and appears to read 'Diego Morales Oñate'. Below the signature is a horizontal dashed line.

Firma

## DECLARACION DE AUTORIA Y HONESTIDAD ACADEMICA

Nombre: Maricela Soraya Ibarra Bedoya  
Cedula de ciudadanía: 1714265152  
Facultad: Jurisprudencia Escuela: Derecho

DECLARO QUE:

El trabajo de investigación de fin de carrera titulado "Software Libre o Software Propietario para una efectiva transferencia de tecnología en el Ecuador", para optar por el título de Abogada den los Tribunales y Juzgados de la República del Ecuador, es de mi autoría exclusiva y producto de mi esfuerzo personal; las ideas, enunciaciones, citas de todo tipo e ilustraciones diversas; obtenidas de cualquier documento, obra, artículo, memoria, entre otros versión impresa o digital, están citadas de forma clara y estricta, tanto en el cuerpo del texto como en la bibliografía.

Estoy plenamente informada de las sanciones universitarias y/o de otro orden en caso de falsedad de lo aquí declarado, en todo o en parte.

Quito, 15 de Julio del 2016

  
Maricela Soraya Ibarra Bedoya  
ESTUDIANTE

## **AUTORIZACION DE DERECHOS DE PROPIEDAD INTELECTUAL**

Yo, Maricela Soraya Ibarra Bedoya, con cedula de identidad número 1714265152, en calidad de autora del trabajo de investigación “Software Libre o Software Propietario para una efectiva transferencia de tecnología en el Ecuador”, autorizo a la Universidad Internacional del Ecuador (UIDE), a hacer uso de todos los contenidos que me pertenecen o de parte de los que contiene esta obra, con fines estrictamente académicos o de investigación.

Los derechos que como autor me corresponden, con excepción de la presente autorización seguirán vigentes a mi favor, de conformidad con lo establecido en los artículos 5,6,8,19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento en Ecuador

Quito, 15 de Julio del 2016



Maricela Soraya Ibarra Bedoya.

ESTUDIANTE

Yo, Maricela Soraya Ibarra Bedoya, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y, que se ha consultado la bibliográfica detallada.

Cedo mis derechos de propiedad intelectual a la Universidad Internacional del Ecuador, sin restricción de ningún género o especial.



Handwritten signature of Maricela Soraya Ibarra Bedoya, written in black ink over a horizontal line.

FIRMA

Yo, Diego Morales Oñate, certifico que conozco al autor (a) del presente trabajo siendo responsable exclusivo tanto en su originalidad, autenticidad, como en su contenido.



Handwritten signature of Diego Morales Oñate, written in black ink over a horizontal line.

FIRMA

CERTIFICACIÓN .....	2
DEDICATORIA .....	2
AGRADECIMIENTO.....	3
DECLARACIÓN.....	4
ÍNDICE DE CONTENIDO .....	5
LISTA DE CUADROS .....	7
RESUMEN.....	8
ABSTRACT .....	10
INTRODUCCIÓN .....	11
CAPÍTULO I.....	13
1. Software libre y Software propietario.....	13
1.1. Definición de Software .....	13
1.1.1. Software Libre y Software Propietario .....	18
1.2. Importancia del Software Libre y del Software Propietario.....	28
1.3. Características del Software Libre y del software Propietario.....	38
1.4. Diferencias entre el Software Libre y el Software Propietario .....	41
CAPÍTULO II .....	44
2. Naturaleza Jurídica del Software y su protección.....	44
2.1. Regulación jurídica de Software y su protección .....	44
2.2. Características y naturaleza del licenciamiento en el software.....	57

2.2.1. Formas o tipos de licenciamiento de Software .....	70
2.2.2. Análisis sobre las formas de licenciamiento Creative Commons y Licencia Pública General de GNU o (GPL) .....	77
CAPÍTULO III .....	84
3. La Transferencia de Tecnologías del software libre y software propietario .....	84
3.1. Presupuestos Estatales sobre adquisición de software propietario .....	84
3.2 Situación jurídica actual del software en el Ecuador .....	87
3.2.1. Análisis del decreto 1014 de 2008 sobre Software Libre .....	94
3.3. La transferencia de tecnologías en Ecuador según el Plan Nacional del Buen Vivir 2013-2017 .....	96
3.4. Transferencia efectiva de tecnología de software. ....	102
CONCLUSIONES .....	110
RECOMENDACIONES .....	112
BIBLIOGRAFÍA .....	114

## **LISTA DE CUADROS**

Cuadro N° 1 Tipo de uso, precio y funcionalidad de las licencia de software .....	765
---	-----

**RESUMEN**

**SOFTWARE LIBRE O SOFTWARE PROPIETARIO PARA UNA**

## **EFFECTIVA TRANSFERENCIA DE TECNOLOGÍA EN ECUADOR**

El trabajo investigativo tiene el objetivo de desarrollar desde el ámbito jurídico una experticia acerca de la utilización de software libre o propietario en la transferencia de tecnología en Ecuador, además de advertir cuál de los dos sistemas puede ser el más apropiado en las relaciones entre empresas privadas o públicas, la investigación también se centra en el análisis de licenciamientos y cual de este tipo de consentimientos resultarían más saludables para una mejor regulación legal del sector, con el fin de equilibrar y permitir la convivencia de derechos tan importantes en las sociedades contemporáneas como son los derechos de propiedad intelectual. De ahí que la investigación es importante e innovadora en la medida que los sectores tecnológicos en nuestro país han sido muy poco desarrollados y por ende el marco regulatorio y el debate jurídico muy escaso.

**Palabras y términos clave:** Software Propietario, Software Libre, Licencia, Transferencia de tecnología, Legislación, Decreto, Planes.

## **ABSTRACT**

### **PROPRIETARY SOFTWARE OR FREE SOFTWARE FOR AN EFFECTIVE TECHNOLOGY TRANSFER IN ECUADOR**

The research work aims to develop from the legal sphere, for expertise on the use of free software or proprietary in technology transfer in Ecuador, in addition to warning which of the two systems may be the most appropriate in relations between private companies or public, research also focuses on the analysis of licensing and what such consents would be healthier for better legal regulation of the sector, in order to balance and allow the coexistence of such important rights in contemporary societies as they are intellectual property rights. Hence research is important and innovative to the extent that technological sectors in our country have been poorly developed and therefore the regulatory framework and very little legal debate.

**Key words and terms:** proprietary software, Free Software Licensing, Technology Transfer, Legislation, Decree, Plans.

## INTRODUCCIÓN

La nueva política de desarrollo científico y tecnológico que quiere implantarse en la educación superior, industrias y comercio en Ecuador se presenta como escenario propicio para el establecimiento de grupos y empresas que desarrollan nuevas tecnologías, este escenario merece ser estudiado desde la óptica jurídica, ya que la creación de nuevos esquemas de información computarizada denominados como software van generando conflictos y debates sociales que deben ser resueltos por las normas y la legislación.

Desde este contexto el trabajo investigativo se constituye en un documento de análisis del software libre y propietario y su relación con la transferencia de tecnología en nuestro país, para lo cual es importante realizar un análisis conceptual de este término que si bien nace de un vocablo inglés en la actualidad es usado comúnmente en nuestra sociedad.

Si revisamos el aspecto histórico de este tema se encuentra que en un primer momento estos programas computarizados eran desarrollados por empresas e instituciones privadas que, con la ayuda del Derecho de Propiedad Intelectual, lograron interponer licencias de explotación para estos programas, un ejemplo claro es Microsoft o IBM que se convirtieron en verdaderas empresas transnacionales dedicadas al desarrollo y transferencia de esta clase de tecnología. Luego, aparece un movimiento tecnológico de software libre, dirigido por Richard Stallman y con ello la fundación de la Free Software Foundation en 1985; colocando así la libertad del usuario informático como un fundamento principal.

El problema de índole jurídico se encuentra en la transferencia de tecnología de software, en el término software libre que se refiere al conjunto de software que por elección

manifiesta de su autor puede ser copiado, estudiado, modificado, utilizado libremente con cualquier fin y redistribuido con o sin cambios o mejoras. Esta nueva posición en el desarrollo de software de manera libre por cualquier ingeniero informático ha dado paso a una compleja relación entre las empresas que se dedicaron en un primer momento al desarrollo de esta tecnología, con estos nuevos modos de proceder de ingenieros que ven en el software libre un campo para enriquecer de mejor forma el desarrollo computacional del mundo.

Debido a este proceder en la actualidad los procesos de investigación científica tecnológica relacionados con software que se vienen desarrollando en el país tienen un difícil campo en el cual desarrollarse, pues se ha denotado el problema en esta relación jurídica entre empresas privadas que desarrollan software propietario y pequeños conjuntos de comunidades de desarrolladores tecnológicos que amplían el conocimiento sobre software libre, lo cual sería subsanable si se realiza una investigación en la cual se deje entrever cuál de los dos procesos es el mejor para el desarrollo de la tecnología y su posterior transferencia a la sociedad.

De ahí que la investigación se centra en el análisis de esta problemática desde el ámbito jurídico y especifica lo caótico que puede terminar siendo estas posiciones entre empresas e ingenieros que desarrollan esta clase de tecnología y su posterior transferencia a la sociedad, por ello; luego de advertir el grado de la contrariedad jurídica el trabajo busca realizar las respectivas conclusiones y recomendaciones y se propondrá la respectiva solución que consiste en la creación de una normativa que regule las relaciones de transferencia de tecnología de software.

## CAPÍTULO I

### 1. Software libre y Software propietario

#### 1.1. Definición de Software

Al revisar la enciclopedia Océano encontramos que básicamente un computador u ordenador se compone principalmente del software, que se refiere al “conjunto de programas, instrucciones y reglas informáticas”, y el hardware “que es el conjunto de los componentes físicos de los que está hecho el equipo”. (Grupo Océano, 2013)

En ese sentido definir el termino software tiene relación directa con los equipos computacionales pues software “es una palabra que proviene del idioma inglés, pero que, gracias a la masificación de uso, ha sido aceptada por la Real Academia Española. Según la RAE, el software es un conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora.” (Real Academia Española, 2014, pág. 950)

Para el tratadista Julián Pérez Porto: “Se considera que el software es el equipamiento lógico e intangible de un ordenador. En otras palabras, el concepto de software abarca a todas las aplicaciones informáticas, como los procesadores de textos, las planillas de cálculo y los editores de imágenes.” (Pérez, 2008)

Según el análisis realizado por el tratadista Ian Sommerville, en una obra traducida por los tratadistas españoles María Isabel, Alfonso Galipienso, Antonio Botía Martínez, Francisco Mora Lizán y José Pascual Trigueros Jover:

*“Muchas personas asocian el término software con los programas de computadora, yo prefiero una definición más amplia donde el software no son solo los programas. Sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta, por lo general, un sistema de software consiste en diversos programas independientes, archivos de configuración que se utilizan para ejecutar estos programas, un sistema de documentación que describe la estructura del sistema, la documentación para el usuario que explica cómo utilizar el sistema y sitios web que peritan a los usuarios descargar la información de productos recientes.”* (Sommerville, 2005, pág. 5)

Esto quiere decir que el software es desarrollado mediante distintos lenguajes de programación, que permiten controlar el comportamiento de una máquina. Estos lenguajes consisten en un conjunto de símbolos y reglas sintácticas y semánticas, que definen el significado de sus elementos y expresiones. Para Sommerville *“un lenguaje de programación permite a los programadores del software especificar, en forma precisa, sobre qué datos debe operar una computadora.”* (Sommerville, 2005)

Esencialmente se puede afirmar que *“el software representa toda la parte inmaterial o intangible que hace funcionar a un ordenador para que realice una serie de tareas específicas”*, (quees.info, 2015) comúnmente conocidos como programas. El software según una explicación realizada por el sitio web quees.info: *“engloba a toda la información digital que hace al conjunto de elementos físicos y materiales que componen el computador y lo hacen trabajar de manera inteligente.”*

(quees.info, 2015)

Una explicación fundamental para comprender de mejor forma lo que significa el software es advertir que todo ordenador o computadora está compuesta por dos *“partes bien diferenciadas el hardware y el software, el hardware representa a todos los materiales físicos de la computadora como la placa base, el microprocesador, el teclado o el disco duro donde se almacena la información, para que todos estos elementos físicos funcionen es necesario el software que representa a toda la parte inmaterial que no vemos del ordenador, el software es el sistema operativo que hace funcionar a tu ordenador, es el editor de textos que te permite escribir documentos, es el videojuego que ocupa tu tiempo de ocio y es el navegador que actualmente estas utilizando para leer este artículo.”* (quees.info, 2015)

Esta explicación que se encuentra en el sitio web quees.info, es muy conocida por casi todas las personas que manejan productos informáticos o computadoras, pues representa la base en la cual se fundamenta el manejo de estas máquinas, para entenderlo es preciso realizar ejemplos o analogías, así se podría decir que *“el software es como la música, cuando un compositor escribe una canción utiliza un lenguaje basado en un sistema de notación musical mediante signos en un documento llamado partitura, el cual es interpretado por los músicos cuando tocan los instrumentos produciendo música. En este símil vemos como la parte física y material son los instrumentos (hardware) los cuales son utilizados por los músicos gracias a la partitura (código del software) y a consecuencia se produce la música (software) que es un efecto inmaterial que no podemos ver ni tocar.”* (quees.info, 2015).

Desde un ámbito histórico: *“el primer software fue el conjunto de cintas perforadas que se utilizaron con el primer computador programable el Z1 en el año 1938, aunque el término fue acuñado por primera vez en el año 1958 por el matemático y estadístico John Wilder Turkey cuando denominó software a los programas que hacían funcionar a las calculadoras electrónicas en su artículo escrito en el Mensuario matemático americano”*. (quees.info, 2015)

Para el tratadista Lev Manovich crear una historia con antecedentes culturales del software es una misión que todavía los estudiosos de las ciencias sociales no han logrado elaborar (Manovich, 2013, pág. 23), porque:

*“Durante las primeras décadas de la era computacional las empresas fabricantes de los primeros ordenadores se centraban en el desarrollo del hardware abandonando el software a un conjunto de programadores los cuales desarrollaban programas complejos y costosos que solo se podían ejecutar en el modelo del ordenador que se estaba fabricando. Con el paso del tiempo y la aparición de los lenguajes de programación se empezaron a desarrollar programas que podían ejecutarse un amplio rango de ordenadores y con utilidades diferentes, durante esta época nació la ingeniería de software como el conjunto de métodos, técnicas y herramientas que se utilizan para el desarrollo y mantenimiento de programas.”*  
(quees.info, 2015)

El transcurso el tiempo no solo hizo más complejo el desarrollo del software sino también encontró variedad de programas y sistemas y por ello en la actualidad existe una clasificación y tipos de Software. Es decir que esta amplia oferta

disponible de programas desarrollados para un fin específico y el número de programas que se incrementan exponencialmente año tras año, han permitido que dentro de la clasificación de software se identifique grupos que responden a diferentes conceptos, por ejemplo, para el sitio web: [quees.info](http://quees.info), se identifican estas clases de software:

***“Ubicación donde se encuentra instalado:***

- ***Software en la red.*** - *Son aquellos programas y aplicaciones que se encuentran alojados en Internet o en un servidor propio y proveen el servicio al cliente mediante una conexión a la red, siendo su principal característica la no necesidad de instalarlo, configurarlo ni mantenerlo en el propio terminal donde se utiliza, programas como Office 365, Dropbox o Google Docs son ejemplos entre otros.*
- ***Software local.*** - *También denominados como software de escritorio son aquellos que necesitan ser instalados y almacenados en el ordenador donde se ejecuta a diferencia de los anteriores, la suite ofimática Office, el programa de diseño gráfico Photoshop o el sistema operativo Windows son ejemplos de este tipo de software.*

***Grado de libertad de uso:***

- ***Software libre.*** - *Representan al conjunto de programas en el que los usuarios disponen de plena libertad para copiarlo, compartirlo y modificarlo, para ello generalmente se tiene acceso al código fuente del propio programa. El sistema operativo Linux, el editor de imágenes Gimp o la suite ofimática Openoffice son ejemplos de este tipo de programas.*

- **Software propietario o privado.** - Representan al conjunto de programas en los que los usuarios tienen limitaciones para modificarlos, compartirlos o copiarlos salvo permiso expreso del titular del software como por ejemplo el sistema operativo Windows, el editor de imágenes Photoshop o la suite ofimática Microsoft Office”. (quees.info, 2015)

El último grupo de clasificación nos presenta la realidad en la cual se desarrolla esta investigación, pues nos muestra que según la libertad de uso de un software este se divide en libre o propietario. Desde este ámbito se puede concluir que estos aspectos de desarrollo del software como parte fundamental en el convivir de las sociedades contemporáneas han ocasionado un aspecto problemático en la regulación legal que desencadena debates y críticas de parte del espacio jurídico y de la misma doctrina legal.

En definitiva y a modo de corolario de este primer subcapítulo se debe manifestar que la industria del desarrollo de software se ha convertido en un protagonista importante dentro de la economía global, ya que mueve millones de dólares al año. Por lo tanto también es escenario de los debates jurídicos que tratan de normalizar y regular un sector que en la actualidad forma parte primordial del desarrollo social contemporáneo.

### **1.1.1. Software Libre y Software Propietario**

Una vez revisado lo que significa el software, es importante manifestar que dentro de su clasificación se encuentran este aspecto de grado de libertad de uso y que dentro de esta conceptualidad del software se divide en software libre y propietario. Para los tratadistas Joaquín Molina Caballero y Laura María Baena

Espejo, el software libre es:

*“Aquel que puede ser distribuido, modificado, copiado y usado; por lo tanto, debe venir acompañado del código fuente para hacer efectivas las libertades que lo caracterizan. Dentro de software libre hay, a su vez, matices que es necesario tener en cuenta. Por ejemplo, el software de dominio público significa que no está protegido por el copyright, por lo tanto, podrían generarse versiones no libres del mismo, en cambio el software libre protegido con copyright impide a los redistribuidores incluir algún tipo de restricción a las libertades propias del software así concebido, es decir, garantiza que las modificaciones seguirán siendo software libre”.*

(Molina & Baena, 2007, pág. 134)

Como se advierte en la cita esta clase de software toma muy en cuenta aspectos de derechos de propiedad intelectual para su uso, es precisamente en esta relación que existe entre el desarrollo de software con los derechos de propiedad intelectual donde se advierte una problemática en su transferencia de conocimiento que fundamenta la presente investigación.

Ahora como vemos dentro del software libre también existe una subdivisión entre programa protegidos por copyright y los que no, *“también es conveniente no confundir el software libre con el software gratuito, este no cuesta nada, hecho que no lo convierte en software libre, porque no es una cuestión de precio, sino de libertad.”* (Molina & Baena, 2007, pág. 134)

Para Richard Stallman el software libre es una cuestión de libertad, no de precio. Para comprender este concepto, debemos pensar en la acepción de libre como

en libertad de expresión. En términos del citado autor el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. (Stallman, Software libre para una sociedad libre. , 2004, pág. 59). Y se refiere especialmente a cuatro clases de libertad para los usuarios de software:

*“1. Libertad 0: la libertad para ejecutar el programa sea cual sea nuestro propósito.*

*2. Libertad 1: la libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades -el acceso al código fuente es condición indispensable para esto-.*

*3. Libertad 2: la libertad para redistribuir copias y ayudar así a tu vecino.*

*4. Libertad 3: 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. Software libre es cualquier programa cuyos usuarios gocen de estas libertades.”* (Stallman, Software libre para una sociedad libre. , 2004, pág. 59)

De modo que esta clase de libertad significa que se es libre de redistribuir copias con o sin modificaciones, de forma gratuita o cobrando por su distribución, a cualquiera y en cualquier lugar. Gozar de esta libertad significa, no tener que pedir permiso a alguna empresa ni pagar para ello. Asimismo, ser libre para introducir modificaciones y utilizarlas de forma privada, ya sea en el trabajo o en el tiempo libre, sin siquiera tener que mencionar su existencia. Si se decidiera publicar estos cambios, no se debería estar obligado a notificárselo a ninguna persona ni de ninguna

forma en particular. La libertad para utilizar un programa significa que cualquier individuo u organización podrán ejecutarlo desde cualquier sistema informático, con cualquier fin y sin la obligación de comunicárselo subsiguientemente ni al desarrollador ni a ninguna entidad en concreto. (Culebro, Gómez Herrera, & Torres, 2006)

La libertad para redistribuir copias supone incluir las formas binarias o ejecutables del programa y el código fuente tanto de las versiones modificadas, como de las originales, ya que debemos tener la libertad para redistribuir tales formas si se encuentra el modo de hacerlo, pues las libertades para hacer cambios y para publicar las versiones mejoradas requieren de la accesibilidad de código fuente, por supuesto de manera libre, condición necesaria del software libre. *“Cuando hablamos de software libre, debemos evitar utilizar expresiones como “regalar” o “gratis”, ya que se puede caer en el error de interpretarlo como una mera cuestión de precio y no de libertad.”* (Stallman, Software libre para una sociedad libre. , 2004, pág. 61)

Por otro lado, también es importante conceptualizar acerca del software propietario, esto con la finalidad de advertir su definición e ir entablando la importancia y diferencias entre estas dos clases de software y donde se produce el problema jurídico y su transferencia adecuada en nuestro país.

Desde ese ámbito y tomando nuevamente como fuente de cita a la investigación realizada por los tratadistas mexicanos: Monserrat Culebro Juárez, Wendy Gómez Herrera y Susana Torres Sánchez, se puede manifestar que el software no libre también es llamado software propietario, software privativo, software privado o software con propietario y se define como:

*“Cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o que su código fuente no está disponible o el acceso a este se encuentra restringido. En el software no libre una persona física o jurídica (por nombrar algunos: compañía, corporación, fundación) posee los derechos de autor sobre un software negando o no otorgando, al mismo tiempo, los derechos de usar el programa con cualquier propósito; de estudiar cómo funciona el programa y adaptarlo a las propias necesidades (donde el acceso al código fuente es una condición previa); de distribuir copias; o de mejorar el programa y hacer públicas las mejoras (para esto el acceso al código fuente es un requisito previo)”. (Culebro, Gómez Herrera, & Torres, 2006, pág. 4)*

De esta manera, según los mismos tratadista: *“un software sigue siendo no libre aun si el código fuente es hecho público, cuando se mantiene la reserva de derechos sobre el uso, modificación o distribución (por ejemplo, el programa de licencias shared source de Microsoft). No existe consenso sobre el término a utilizar para referirse al opuesto del software libre. Entre los términos más usados, en orden de frecuencia de uso (usando como fuente oficiosa una serie de consultas con el buscador Google en mayo de 2015) se encuentran:*

- *Software propietario.*
- *Software privativo.*
- *Software no libre.*
- *Software privado.*
- *Software con propietario.*

- *Software semilibre.*
- *Freeware.*
- *Abandonware.*
- *Warez.*” (Culebro, Gómez Herrera, & Torres, 2006, pág. 4)

A continuación, para una mejor comprensión de cada uno de esta clase de software pasamos a conceptualizarlos.

**“1. La expresión *software propietario*.** - proviene del término en inglés “proprietary software”. “En la lengua anglosajona, “proprietary” significa: poseído o controlado privadamente, que destaca la mantención de la reserva de derechos sobre el uso, modificación o redistribución del software. Inicialmente utilizado, pero con el inconveniente que la acepción proviene de una traducción literal del inglés, no correspondiendo su uso como adjetivo en el español, de manera que puede ser considerado como un barbarismo. Sin embargo, todavía es el término preferido por cerca del 73 % de los sitios en Internet.” (Culebro, Gómez Herrera, & Torres, 2006, pág. 4)

En la investigación realizada por los mismos tratadistas mexicanos se realiza un análisis sobre las objeciones a cada uno de los términos, en este caso “el término propietario en español resultaría inadecuado, pues significa que tiene derecho de propiedad sobre una cosa por lo que no podría calificarse de “propietario” al software, porque este no tiene propiedad sobre nada (es decir, no es dueño de nada) y, además, no podría serlo (porque es cosa y no persona). Asimismo, la expresión “software propietario” podría ser interpretada como “software sujeto a propiedad” (derechos o titularidad) y su opuesto, el software libre, está sujeto al derecho de

autor.” (Culebro, Gómez Herrera, & Torres, 2006, pág. 5)

**2. La expresión software privativo.-** fue comenzada a ser utilizada por Richard Stallman, desde el año 2003, en sus conferencias sobre software libre, pues sería más adecuada que “software propietario” para definir en español al software que no es libre (“proprietary software” en inglés).

“Se realizó este cambio idiomático, básicamente por las siguientes razones: El término “privativo” significa que causa privación o restricción de derechos o libertades, es justamente lo que se pretende describir con el (privación a los usuarios de sus libertades en relación al software). Su uso ha ido en aumento constante y actualmente se puede constatar que cerca de 20 % de los sitios en Internet lo utilizan.” (Culebro, Gómez Herrera, & Torres, 2006, pág. 5)

Sin embargo, las objeciones a la utilización de este término se enmarcan en que “haría referencia a una propiedad diferente a la de ser un software no libre, ya que la principal acepción de “privativo” es que causa privación o la significa. En este sentido, el contrario de “software privativo” sería “software no privativo”, es decir, sin limitaciones o como se le conoce: de dominio público. Así, software libre y su opuesto serían ambos conceptos diferentes a los sugeridos por “software privativo”. Asimismo, este término sería aplicable, de modo figurativo, al software cuyo coste es realmente elevado, no pudiendo ser soportado por un particular, caso muy común en el software diseñado para realizar labores muy específicas, con independencia a si este es libre o no.” (Culebro, Gómez Herrera, & Torres, 2006, pág. 5)

**“3. *Software no libre o software no-libre*** (en inglés “non-free software”), actualmente se utiliza en 5,5 % de los sitios Internet encontrados y sus objeciones a este término no libre” se encuentra en que este término es usado por la Free Software Foundation “FSF” (Fundación para el Software Libre), aunque para agrupar “todo el software que no es libre, es decir, incluye al llamado en inglés “semi-free software” (software semilibre) y al “proprietary software” (puede verse esto en Categories of Free and Non-Free Software de la FSF, en inglés).” (Culebro, Gómez Herrera, & Torres, 2006, pág. 6)

**“4. *Software privado***. - cuya expresión es usada por la relación entre los conceptos de (tener) propietario y ser privado. Su utilización es poco difundida, cerca del 1 % de los sitios Internet utilizan este término. Este término sería inadecuado debido a que, en una de sus acepciones, la palabra “privado” se entiende como antónimo de “público”, o sea, que no es de propiedad pública o estatal, sino que pertenece a particulares, provocando que esta categoría se interprete como no referente al Estado, lo que produciría la exclusión del software (no libre) generado por el aparato estatal. Además, al igual que con el término “software privativo”, el contrario “literal” de “software privado”, es decir, el “software público” se asocia generalmente con software de dominio público.” (Culebro, Gómez Herrera, & Torres, 2006, pág. 6)

**5. *software con propietario***. - pretende corregir el defecto de la expresión “software propietario”, y sería más cercana al término más utilizado (“propietario”), el cual se podría considerar una abreviación de este. Su uso sin embargo está muy poco difundido (0,1 %). “Se argumenta en contra del término “con propietario”

justamente su similitud con “propietario” en inglés, que solo haría referencia a un aspecto del software que no es libre, manteniendo una de las principales críticas a este (de “software sujeto a derechos” o “propiedad”). Adicionalmente, si “propietario” refiere al titular de los derechos de autor (y está claro que no puede referir al usuario, en tanto éste es simplemente un cesionario), no resuelve la contradicción: todo el software libre tiene también titulares de derechos de autor.” (Culebro, Gómez Herrera, & Torres, 2006, pág. 6)

En conclusión, respecto a los problemas de esta acepción y para los efectos del presente trabajo tomaremos la concepción de “software propietario”, tal como lo define la Free Software Foundation “FSF” (Fundación para el software libre), aunque como ya hemos explicado no es la definición más correcta, si es la que más se utiliza en la red y se refiere a todo software que no satisface los criterios para el software libre establecidos en la propia fundación.

Propietario significa que algún individuo o compañía tiene la titularidad de los derechos exclusivos de autor sobre una porción del software, y al mismo tiempo, niega a otras personas el acceso al código fuente del software y al derecho a copiar, modificar y estudiar el software. El término “propietario” significa “privadamente apropiado y controlado”. *“Es aquel que no es libre ni semilibre; por lo tanto, su redistribución, modificación y copia están prohibidas o, al menos, tan restringidas que es imposible hacerlas efectivas.”* (Molina & Baena, 2007, pág. 134). Ahora enunciaremos y explicaremos otros tipos de software que existen, a efecto de proporcionar mayor claridad en cuanto al tema que nos ocupa en el presente apartado.

**6. Software semilibre.** - en cambio *“es aquel que mantiene las mismas características que el software libre para los usuarios individuales, entidades educativas o sin ánimo de lucro, sin embargo, prohíbe esas libertades para su uso comercial o empresarial.”* (Carranza M. , 2006, pág. 102)

Por su parte el Freeware no tiene una definición clara y precisa, sin embargo, suele usarse para clasificar al software que *“puede redistribuirse libremente pero no modificarse, entre otras cosas, porque no está disponible su código fuente. El freeware no es software libre.”* (Carranza M. , 2006, pág. 102)

**7. El Shareware.** - en cambio *“es un software que permite su redistribución, sin embargo, no viene acompañado de su código fuente y, por tanto, no puede ser modificado. Además, pasado un periodo de tiempo, normalmente es necesario pagar una licencia para continuar usándolo, luego tampoco es software libre.”* (Carranza M. , 2006, pág. 102)

**8. El abandonware.-** es *“software cuyos derechos de autor ya no son defendidos o que ya no está siendo vendido por la compañía que lo hizo”, y por eso, se dice que ha sido “abandonado”.* (Carranza M. , 2006, pág. 103)

**9. El Warez.-** es un sufijo empleado en la jerga informática para formar términos que aluden a categorías de software, “warez” refiere a una categoría de software distinta de las anteriores. “Warez” es un término muy usado en las subculturas cracker para aludir a versiones crackeadas de software comercial, versiones en las cuales la protección de los derechos de autor ha sido quitada. Los hackers reconocen este término, pero no lo usan. “Los warez son distribuciones de

software sujetas a los derechos de autor, comercializadas en violación a la licencia de derechos de autor del software, en muchos casos la distribución de los warez es ilegal. La justificación que dan los crackers para el uso de los warez incluye la alegada posibilidad de la protección de los derechos de autor y la percibida injusticia de no compartir la información con aquellos que no podrían obtenerlo de otra manera que a través de la compra”. (Carranza M. , 2006, pág. 103)

Como se advierte de la exposición de lo que significa software libre y software propietario que es el término que en definitiva acogemos en este trabajo investigativo son antagónicos y contradictorios entre ellos, a pesar de las objeciones a cada uno de los 9 tipos de software propietario o con propietario, lo importantes en esta parte del trabajo es dejar concluido que los dos tipos de software, tanto en libre como el propietario guardan unas características propias y tienen una importancia significativa en el actual desarrollo social de la humanidad.

## **1.2. Importancia del Software Libre y del Software Propietario**

En un ensayo realizado por el tratadista Richard Stallman se puede advertir que la importancia del software libre nace con su mismo origen, porque desde 1983 el movimiento del software libre defiende la libertad de los usuarios de ordenadores para que sean ellos quienes ejerzan el control del software que utilizan, y no al revés. (Stallman, El software libre es ahora aún más importante, 2015, pág. 103)

Dentro del ensayo del tratadista Stallman se advierte que fundamenta la importancia del software libre en seis aspectos básicos que se ejercitan en: La injusticia de lo privativo, El software privativo y el SaaS, Injusticias primarias y

secundarias, El software libre y el Estado, Software libre y educación, Software libre: Mucho más que «ventajas»

En la primera parte del ensayo Stallman expone que existe una injusticia de lo privativo, porque los usuarios no controlan el programa, el programa controla a los usuarios. “En el caso del software propietario, siempre hay alguna entidad (el desarrollador o «propietario» del programa) que controla el programa y, a través del programa, ejerce su poder sobre los usuarios. Un programa que no es libre es un yugo, un instrumento de poder injusto”. (Stallman, El software libre es ahora aún más importante, 2015)

De ahí que el autor Stallman determina que la importancia del software libre se encuentra en distribuir programas con libertad, esto no supone un maltrato hacia los usuarios. Sin embargo, si un programa no se distribuye, no se estará maltratando a nadie. Si usted escribe un programa y lo usa de forma privada, esto no es malo para los demás. Estará perdiendo la oportunidad de hacer el bien, pero esto no es lo mismo que hacer el mal. Entonces, cuando decimos que todo el software debe ser libre, queremos decir que todas las copias de un programa deben conceder las cuatro libertades, no que todo el mundo tenga la obligación de ofrecer copias a los demás. (Stallman, Software libre para una sociedad libre. , 2004)

En cuanto al software privativo y el SaaS, Stallman afirma que el software privativo fue el primer medio que usaron las empresas para tomar el control de las tareas informáticas de las personas. Hoy existe otro medio, llamado «servicio sustitutivo del software» (SaaS), que significa que un servidor ajeno realiza las tareas informáticas del usuario.

El SaaS no implica que los programas en ese servidor sean privativos (aunque suelen serlo). Sin embargo, usar un SaaS provoca las mismas injusticias que usar un programa privativo: son dos caminos que conducen al mismo lugar dañino. Tomemos el ejemplo de un SaaS de traducción: el usuario envía un texto al servidor, el servidor lo traduce (del inglés al español, por ejemplo) y devuelve la traducción al usuario. De esta forma, el trabajo de traducción está bajo el control del administrador del servidor, no del usuario.

Si usted usa un SaaS, quien controla el servidor controla sus tareas informáticas. Esto implica confiar todos los datos relevantes al administrador del servidor, quien además estará obligado a mostrarlos al Estado; entonces, ¿a quién sirve realmente ese servidor?

En lo que tiene que ver con las injusticias primarias y secundarias, estas se muestran cuando usted usa programas privativos o el SaaS, en primer lugar se está haciendo mal a sí mismo, ya que le está concediendo a otra persona un poder injusto sobre usted. Por su propio bien, debería evitarlo. Si se compromete a no compartir, también estará perjudicando a otros. Respetar tal compromiso es malo, y romperlo es menos malo, pero para ser honesto de verdad, no debe comprometerse en absoluto.

Utilizar programas privativos y el SaaS conlleva otro perjuicio: premia al instigador, promueve el desarrollo de ese programa o «servicio», y conduce a que más y más personas caigan bajo el dominio de la empresa en cuestión. Todas las formas de daño indirecto adquieren una mayor dimensión cuando el usuario es un ente público o una escuela.

En la parte sobre el software libre y el Estado, Stallman expone que los entes públicos existen para los ciudadanos, no para sí mismos. Cuando realizan tareas informáticas, lo hacen para los ciudadanos. Tienen el deber de conservar el control absoluto sobre esas tareas a fin de garantizar su correcta ejecución en beneficio de los ciudadanos. En esto consiste la soberanía informática del Estado. Nunca deben permitir que el control de las tareas informáticas del Estado caiga en manos privadas.

Para conservar el control de las tareas informáticas que realizan en nombre de los ciudadanos, los entes públicos no deben usar software privativo (software que está bajo el control de entidades que no son estatales). Tampoco deben delegar la realización de esas tareas a un servicio programado y ejecutado por un ente distinto del Estado, porque eso sería un SaaS.

También Stallman afirma que el software libre aporta a la educación, porque las escuelas influyen sobre el futuro de la sociedad a través de lo que enseñan. Para que esta influencia sea positiva, deben enseñar exclusivamente software libre. Enseñar el uso de un programa privativo equivale a imponer la dependencia, que es lo contrario de la misión educativa. Capacitando a los alumnos en el uso del software libre, las escuelas dirigirán el futuro de la sociedad hacia la libertad, y ayudarán a los programadores talentosos a dominar el oficio.

Finalmente, la importancia del software libre radica según Stallman en que este tipo de sistemas generan más ventajas que el software propietario, incluso el autor afirma que existen aspectos mucho más relevantes que ventajas dentro de la aplicación del software libre:

*“A menudo me piden que describa las «ventajas» del software libre. Pero el término «ventajas» es demasiado débil cuando se trata de la libertad. La vida sin libertad es tiranía, y eso se aplica a la informática y a cualquier otra actividad de nuestras vidas. Debemos rechazar conceder el control de nuestras tareas de computación a los desarrolladores de un programa o de un servicio informático. Es lo que hay que hacer por razones egoístas, aunque no solo por razones egoístas”.* (Stallman, Software libre para una sociedad libre. , 2004)

Al hablar de ventajas es importante dejar señalado que estas también dan un fundamento a la importancia de la aplicación de software libre y por lo tanto de su transferencia en nuestro país, así que pasemos a anotar algunas ventajas del software libre según el estudio de los tratadistas Jorge Daniel Ceballos Macías y Edith Ramírez:

***“Ventajas del Software Libre:***

***1. Económico.*** - (más de mil millones de euros en licencias de Microsoft en España anuales) *El bajo o nulo coste de los productos libres permiten proporcionar a las PYMES servicios y ampliar sus infraestructuras sin que se vean reducidos sus intentos de crecimiento por no poder hacer frente al pago de grandes cantidades en licencias.*

***2. Libertad de uso y redistribución.*** - *Las licencias de software libre existentes permiten la instalación del software tantas veces y en tantas máquinas como el usuario desee.*

***3. Independencia tecnológica.*** - *El acceso al código fuente permite el desarrollo de nuevos productos sin la necesidad de desarrollar todo el*

*proceso partiendo de cero. El secretismo tecnológico es uno de los grandes frenos y desequilibrios existentes para el desarrollo en el modelo de propiedad intelectual.*

**4. Fomento de la libre competencia al basarse en servicios y no licencias.**

*Uno de los modelos de negocio 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.*

**5. Soporte y compatibilidad a largo plazo.** - *Este punto, más que una ventaja del software libre es una desventaja del software propietario, por lo que la elección de software libre evita este problema. Al vendedor, una vez ha alcanzado el máximo de ventas que puede realizar de un producto, no le interesa que sus clientes continúen con él.*

**6. Formatos estándar.** - *Los formatos estándar permiten una interoperatividad más alta entre sistemas, evitando incompatibilidades. Estos formatos afectan a todos los niveles.*

**7. Sistemas sin puertas traseras y más seguros.** - *El acceso al código fuente permite que tanto hackers como empresas de seguridad de todo el mundo puedan auditar los programas, por lo que la existencia de puertas traseras es ilógica ya que se pondría en evidencia y contraviene el interés de la comunidad que es la que lo genera.*

**8. Corrección más rápida y eficiente de fallos.** - *El funcionamiento e interés conjunto de la comunidad ha demostrado solucionar más rápidamente los fallos de seguridad en el software libre, algo que desgraciadamente en el software propietario es más difícil y costoso.*

**9. Métodos simples y unificados de gestión de software.** - Hoy en día la mayoría de distribuciones de Linux incorporan alguno de los sistemas que unifican el método de instalación de programas, librerías, etc. por parte de los usuarios. Esto llega a simplificar hasta el grado de marcar o desmarcar una casilla para la gestión del software, y permiten el acceso a los miles de aplicaciones existentes de forma segura y gratuita a la par que evitan tener que recurrir a páginas web de dudosa ética desde las que los usuarios instalan sin saberlo spyware o virus informáticos en sus sistemas. Este sistema de acceso y gestión del software se hace prácticamente utópico si se extrapola al mercado propietario.

**10. Sistema en expansión.** - Las ventajas especialmente económicas que aportan las soluciones libres a muchas empresas y las aportaciones de la comunidad han permitido un constante crecimiento del software libre, hasta superar en ocasiones como en el de los servidores web, al mercado propietario.” (Ceballos & Ramírez, 2015, pág. 23)

“En definitiva, el software libre ya no es una promesa, es una realidad y se utiliza en sistemas de producción por algunas de las empresas tecnológicas más importantes como IBM, SUN Microsystems, Google, Hewlett-Packard, etc. Paradójicamente, incluso Microsoft, que posee sus propias herramientas, emplea GNU Linux en muchos de sus servidores. Porque la libertad incluye el ser libre de cooperar con los demás.” (Ceballos & Ramírez, 2015, pág. 24). Negar esta libertad equivale a mantener a las personas divididas, primer paso para tiranizarlas. En la comunidad del software libre somos muy conscientes de la importancia de la libertad para cooperar porque nuestro trabajo consiste en una cooperación organizada. Si un

amigo suyo viene a visitarlo y lo ve usando un programa, puede pedirle una copia. Un programa que le impide a usted que lo redistribuya, o le indica que «no debe hacerlo», es antisocial.

Por otro lado tomando como referencia el origen del software propietario, se puede evidenciar que la importancia del Software Propietario radica en que da una gran importancia a los derechos de autor, es decir:

*“No es más que un sistema operativo de manejo comercial que tiene expectativas como de actualizaciones y uso de programas reconocido en el área de la informática, es decir que se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido. Para la Fundación para el Software Libre (FSF) este concepto se aplica a cualquier software que no es libre o que sólo lo es parcialmente, sea porque su uso, redistribución o modificación está prohibida, o requiere permiso expreso del titular del software este sistema operativo posee varias actualizaciones que serían Microsoft Windows seven, xp, vista entre otros los cuales forman el manejo de los sistemas operativos privados con derecho de autor sin modificación alguna al código fuente del sistema.” (Castro, 2010)*

Y al igual que el software libre el propietario tiene unas ventajas que avalan su utilización y que las describimos a continuación:

***“Ventajas del Software Propietario:***

***1. Propiedad y decisión de uso del software por parte de la empresa.*** - El desarrollo de la mayoría de software requiere importantes inversiones para su estudio y desarrollo. Este esfuerzo, de no ser protegido se haría en balde, puesto que la competencia se podría apropiar inmediatamente del producto una vez finalizado, para sus propios fines.

***2. Soporte para todo tipo de hardware.*** - Se le toma más importancia al mercado del sistema operativo mayoritario, que es Microsoft Windows, y no al resto de sistemas operativos de tipo Unix, que es minoritario. Se da, que el actual dominio de mercado invita a los fabricantes de dispositivos para ordenadores personales a producir drivers o hardware, solo compatible con Windows. Por lo que la elección del sistema operativo de Microsoft tiene garantizado un soporte de hardware seguro.

***3. Mejor acabado de la mayoría de aplicaciones.*** - El desarrollador de software propietario, generalmente, da un mejor acabado a las aplicaciones en cuestiones, tanto de estética, como de usabilidad de la aplicación.

***4. Las aplicaciones número uno son propietarias.*** - Obviamente, el actual dominio de mercado no sólo interesa a los fabricantes de hardware, sino que también a los de software. Algunas de las aplicaciones más demandadas son, según ámbitos: Microsoft Office, Nero Burning Rom, 3DStudio, etc...

***5. El ocio para ordenadores personales está destinado al mercado propietario.*** - Los desarrolladores de juegos tienen sus miras en el mercado más lucrativo, que es el de las consolas y en última instancia el de los ordenadores personales. Además, en estos últimos la práctica totalidad de títulos benefician a Microsoft Windows.

**6. Menor necesidad de técnicos especializados.** - *El mejor acabado de la mayoría de sistemas de software propietario y los estándares de facto actuales permiten una simplificación del tratamiento de dichos sistemas, dando acceso a su uso y administración, a técnicos que requieren una menor formación, reduciendo costes de mantenimiento.*

**7. Mayor mercado laboral actual.** - *Como muchas de las ventajas restantes se deben al dominio del mercado, no hay que ignorar que cualquier trabajo relacionado con la informática pasará, en casi todos los casos, por conocer herramientas de software propietario.*

**8. Mejor protección de las obras con copyright.** - *Las obras protegidas por copyright se ven beneficiadas por mecanismos anticopia, como el DRM y otras medidas, que dificultan a la piratería. El fenómeno de la piratería, está en constante debate sobre a quién perjudica y quién beneficia realmente.*

**9. Unificación de productos.** - *Una de las ventajas más destacables del software propietario es la toma de decisiones centralizada que se hace en torno a una línea de productos, haciendo que no se desvíe de la idea principal y generando productos funcionales y altamente compatibles.” (Ceballos & Ramírez, 2015, pág. 26)*

En definitiva, el software libre tiene una clara desventaja: al ser producido y tomadas las decisiones por un exceso de grupos y organismos descentralizados que trabajan en líneas paralelas y no llegan muchas veces a acuerdos entre ellos. Esto ocasiona que haya un gran caos a programadores y usuarios finales que no saben que vías tomar. Además, genera productos cuya compatibilidad deja bastante que desear.

A modo de conclusión se puede afirmar que tanto el software libre como el

software propietario guardan una gran importancia en el desarrollo de la sociedad contemporánea y que de igual forma tienen unas características básicas que han permitido su evolución más dinámica y vertiginosa en los últimos tiempos.

### **1.3. Características del Software Libre y del software Propietario**

Como se advierte en acápites anteriores el software libre y propietario han ido desarrollándose con el lapso del tiempo, esto ha permitido que también se desplieguen en sus dinámicas una serie de características propias de cada sistema, es así que para el tratadista Carlos G. Figueroa se pueden establecer estos cinco aspectos dentro del sistema de software libre:

#### ***Características del Software Libre:***

1. *Esperan licencias abiertas y la inexistencia de métodos de activación.*
2. *Esperan poder trabajar de la manera que ellos decidan.*
3. *Manejan el código fuente.*
4. *Quieren el control de sus propios sistemas.*
5. *Los usuarios de software libre exploran.*
6. *Los usuarios de software libre no tienen miedo a la línea de comandos. (Figueroa, 2008)*

*“El uso de software libre -de código abierto- ha sido en sus orígenes una actividad de expertos, a tal punto que existe una importantísima cantidad de servidores y equipos destinados a las comunicaciones e Internet que funcionan con productos de estas características. Pero últimamente se ha popularizado gracias a la avalancha de aplicaciones destinadas al*

*usuario final no informático.*

*Los más de veinte años de desarrollo del modelo y la producción cooperativa de programas lo han ido acercando a éste, de tal manera que actualmente dispone de alternativas a los sistemas operativos y cuenta con un amplio conjunto de aplicaciones, tanto o más funcionales que sus equivalentes bajo licencia propietaria. Las versiones actuales de estos productos poseen un elevado grado de amigabilidad, permitiendo una instalación relativamente sencilla, fácil utilización y compatibilidad con otras herramientas.” (Brocca & Casamiquela, 2005, pág. 8)*

Las ventajas más importantes que presentan son la adaptabilidad y seguridad, que se desprenden de la libre disponibilidad del código fuente y el importante ahorro en costos de licencias de utilización. Como desventajas se puede señalar que la curva de aprendizaje de alguno de estos productos generalmente se extiende algo más en el tiempo que su contraparte propietaria y que a la fecha algunos dispositivos físicos no funcionan adecuadamente bajo software de este modelo.

El software libre -de código abierto- produjo una verdadera revolución en el mundo de la informática. Las grandes empresas desarrolladoras de software propietario se están sumando a esa corriente: en algunos casos entregan de forma gratuita sus productos, en otros ponen a disposición de la comunidad el código fuente de sus aplicaciones, compatibilizando a veces sus licencias con las de la comunidad FLOSS.

Por otro lado, tomando en cuenta la definición de software propietario se

puede determinar que si es considerado como “*cualquier programa informático en el que el usuario tiene limitaciones para usarlo, modificarlo o redistribuirlo (esto último con o sin modificaciones). (También llamado código cerrado o software no libre, privado o privativo)*” (Castro, 2010) y también fundamentándonos en que para la Fundación para el Software Libre (FSF) este concepto se aplica a cualquier software que no es libre o que sólo lo es parcialmente (semi-libre), sea porque su uso, redistribución o modificación está prohibida, o requiere permiso expreso del titular del software, se puede establecer las siguientes características:

***“Características del Software Propietario:***

- 1. Este software no te pertenece no puedes hacerle ningún tipo de modificación al código fuente.*
- 2. No puedes distribuirlo sin el permiso del propietario.*
- 3. El usuario debe realizar cursos para el manejo del sistema como tal debido a su alta capacidad de uso.*
- 4. Este posee accesos para que el usuario implemente otro tipo de sistema en él.*
- 5. Cualquier ayuda en cuanto a los antivirus.” (Figuerola, 2008, pág. 10)*

El software propietario se encuentra altamente difundido y cubre todas las posibilidades técnicas existentes al momento, tan es así que, en algunos casos muy específicos, solamente existen soluciones bajo este paradigma.

Muchas de las piezas de software propietarias constituyen un estándar de facto, a tal extremo que los usuarios finales se encuentran familiarizados con ellas y utilizan el nombre comercial de un producto para referenciar un sistema operativo o

un determinado tipo de aplicación.

Como el software propietario responde fundamentalmente a los intereses comerciales de quien lo desarrolla, es necesario adquirir las correspondientes licencias para usarlo dentro de la legalidad. Además, y por el mismo motivo, periódicamente se liberan al mercado nuevas versiones de los productos, que corrigen los defectos encontrados e incorporan nuevas funcionalidades.

Son justamente estas nuevas prestaciones las que, tarde o temprano, obligan al usuario a migrar a las versiones actualizadas, con el consiguiente costo asociado. También debe tenerse en cuenta que pasado cierto tiempo, la empresa comercializadora deja de dar soporte técnico al producto. Esta es también una frecuente causa de migración.

Debe tenerse en cuenta, en aquellos casos donde no se dispone del código fuente de estos programas, que existen situaciones de utilización potencialmente riesgosas en cuanto a errores en el diseño o vulnerabilidades.

Podemos resaltar otras características de otros autores y estudios sobre software libre y propietario, pero en definitiva son en cada uno de estos dos sistemas las características aquí anotadas las que fundamentan su actual comportamiento en el mercado económico y en el desarrollo cotidiano de nuestra sociedad.

#### **1.4. Diferencias entre el Software Libre y el Software Propietario**

Para finalizar el primer capítulo del trabajo de titulación es importante dejar

manifestado las diferencias entre software propietario y libre, con la finalidad de advertir su naturaleza y aplicación en el mercado económico y en la sociedad y de ese modo poder interpretar la forma más favorable en la que se debería desarrollar su transferencia en nuestro país.

Desde ese ámbito se puede manifestar que, no obstante, el software libre está en constante crecimiento y evolución, logrando día a día mejorar y eliminar los aspectos en que flojea. Por otra banda las empresas de software propietario empiezan a invertir en el software libre, en vistas de que este modelo es mucho más eficiente que el modelo tradicional.

El Software propietario consta de estas características que lo diferencia y contraponen con el software libre.

***“Diferencias del Software Propietario:***

- 1. Dependencia de un proveedor; la licencia deja claro que el software sigue siendo propiedad de la empresa productora del mismo y que el usuario no está facultado a realizar ningún cambio en él, ni tampoco analizarlo para determinar cómo realiza sus funciones.*
- 2. Es desarrollado por una empresa y no difunde sus especificidades.*
- 3. El código es cerrado, nadie puede ver cómo está elaborado el programa.*
- 4. La licencia nos limita a usar ese software por una sola persona (quien lo adquiere). No se puede redistribuir, a no ser que la licencia lo permita.*
- 5. Al no tener acceso al código fuente, no podemos corregir los posibles errores que tenga el programa, ni adaptarlo a nuestras necesidades.*
- 6. Dependemos de las variaciones que la empresa propietaria realice con*

*las nuevas versiones.*

7. *Lo habitual es tener que pagar por adquirir estos programas, y muchos de ellos ya sabemos que no son precisamente baratos y asequibles.*

***Diferencias del Software Libre:***

1. *Libertad de ejecutar el programa, con cualquier propósito.*
2. *Libertad de estudiar cómo funciona el programa y de adaptarlo a sus necesidades.*
3. *Libertad de redistribuir copias del programa.*
4. *Libertad de mejorar el programa y redistribuir dichas modificaciones. ”*

(Figuerola, 2008, pág. 22)

Con estas diferenciaciones se pretende establecer que para el análisis jurídico posterior al que se va arribar en este trabajo investigativo es necesario buscar la esencia y naturaleza de estos aspectos novedosos que forman parte del desarrollo social económico y cultural de la sociedad global contemporánea, de ahí su importancia y fundamental investigación desde el ámbito del derecho y de la ley.

## CAPÍTULO II

### 2. Naturaleza Jurídica del Software y su protección

#### 2.1. Regulación jurídica de Software y su protección

La problemática que origina el trabajo investigativo surge en la cuestión de buscar una protección jurídica al software y se va desarrollando en advertir cuál de los dos tipos de software: propietario o libre es el mejor para la transferencia de tecnología informática en nuestro país. Desde ese punto básico, se puede afirmar que en este segundo apartado del trabajo se expone lo referente a la regulación jurídica y a la naturaleza de esta reglamentación para con el software.

Entrando ya en tema de análisis se debe primero tener en cuenta que, desde los orígenes de la informática, el software era considerado un componente más de la máquina y los fabricantes de computadoras lo desarrollaban y distribuían, incluso poniendo a disposición de los usuarios el código fuente, casi sin restricciones en cuanto a su utilización o destino. *“Pero, para evitar inconvenientes por acciones antimonopólicas, a partir de 1970 la firma IBM independiza la comercialización del software de la venta de sus equipos y ejerce derechos legales sobre el mismo.”* (Brocca & Casamiquela, 2006)

*“Esto quiere decir, que en un primer momento mientras la industria del hardware estuvo ligada con la del software, no se previó la necesidad de proveer a éste último de una protección jurídica especial. El ordenador era el bien jurídico que merecía una tutela legal y su programa estaba protegido por el régimen legal de aquél. Sobre el punto, no hubo mayores inconvenientes para el ámbito jurídico.”*

(Moisset & Hiruela, 2010, pág. 1)

Esto obedeció también a que *“el hardware pudo insertarse fácilmente en las categorías jurídicas existentes, siendo clasificable como cosa mueble y su tutela se instrumentó”* (Moisset & Hiruela, 2010, pág. 1) a través de instrumentos del derecho intelectual como las patentes.

Según el estudio realizado por los tratadistas argentinos Luis Moisset de Espanés y María del Pilar Hiruela de Fernández; *“desde la década del 60, y en virtud de la espectacular baja del costo de los componentes materiales de los sistemas informáticos registrada a partir de la tercera generación de computadoras gracias a los procesos de miniaturización, el elemento intelectual de esta industria (el software) comenzó a ocupar el primer plano de la escena.”* (Moisset & Hiruela, 2010, pág. 1)

*“En otras palabras, como resultado de la creciente demanda de los programas de computación, su vulnerabilidad y gran costo económico, el software comenzó a adquirir -para la ciencia del derecho- un valor independiente del hardware, generando un intenso debate en orden a su protección jurídica”.* (Moisset & Hiruela, 2010, pág. 2)

Se inicia así un cambio de modelo económico donde el software se desvincula del fabricante del equipo y se desarrollan aplicaciones que pueden correr sobre diversas plataformas. En este nuevo escenario, la mayoría de los diseñadores de software comienza a utilizar todos los medios técnicos y legales disponibles para proteger su desarrollo y mantener su posición en el mercado.

La cuestión de cómo proteger jurídicamente al software, en tanto creación del intelecto que indudablemente es, se ha ido convirtiendo en uno de los temas más candentes y de mayor desarrollo del derecho informático actual, en efecto el tratadista Carlos Correa afirma que: *“La transformación del software en un objeto separado de comercialización, y su vulnerabilidad –dada la facilidad de copiarlo a bajo costo- son las dos causas más importantes que han estimulado el debate sobre la protección jurídica”* (Correa, 1994)

Sin embargo, las características inmateriales del software no han facilitado su encuadre en las categorías jurídicas existentes y, por lo tanto, ha resultado difícil de tutelar. Las opiniones al respecto no son unánimes: *“en algunos casos, se propone regularlo a través de marcas o patentes -en el marco del régimen jurídico de la propiedad industrial- o utilizar los mecanismos de protección de la propiedad intelectual, mientras que, en otros, se pretende desarrollar un sistema especial.”* (Brocca & Casamiquela, 2006)

Esto hace concluir que *“a diferencia de lo que había ocurrido para la regulación del hardware, este nuevo objeto de derecho no es tan sencillo de tutelar, desde que apareciera como refractario a las categorías jurídicas existentes, y presentara como característica propia la de ser un objeto inmaterial o incorporeal. En efecto, dada su particular naturaleza (intelectual, moral y económica), el software fue reacio a dejarse encuadrar en las clásicas categorías jurídicas existentes.”* (Moisset & Hiruela, 2010, pág. 2)

*“Por ello, la búsqueda de un régimen de derecho idóneo no ha sido fácil y aún hoy en día”* (Moisset & Hiruela, 2010, pág. 2) las opiniones doctrinarias y

jurisprudenciales no son unánimes al respecto a nivel mundial, algunos tratadistas han *“sostenido que el programa debe ser regulado a través del derecho de patentes; otros, aseveran que el régimen idóneo para tutelar a los programas de computación es el propio del Derecho de autor. Finalmente, un sector -hoy minoritario- considera que es menester crear un nuevo ordenamiento jurídico, que atienda y prevea las particularidades que caracterizan al software.”* (Moisset & Hiruela, 2010, pág. 2)

*“Sin intentar un análisis exhaustivo de las modalidades de protección propuestas, mencionaremos los sistemas mayormente propugnados, puntualizando los beneficios que cada de ellos uno ofrece y las diversas críticas u objeciones que se han formulado a su respecto”*  
(Moisset & Hiruela, 2010, pág. 2)

En primer lugar, el derecho de patentes, que desde un inicio no ha sido visto con buenos ojos para proteger inventos relacionados con el software. Esta posición la demuestra el estudio realizado por los tratadistas argentinos: Luis Moisset de Espanés y María del Pilar Hiruela de Fernández, que afirman:

*“Los programas de computación no observan -al menos en principio las condiciones que el derecho de patentes exige como recaudos insoslayables para la protección de un bien. Nos referimos a las tres clásicas condiciones exigidas para el otorgamiento de una patente de invención, a saber: a) Novedad; b) Nivel inventivo y c) Aplicabilidad industrial”.* (Moisset & Hiruela, 2010)

Es decir, que un programa de computación no cumple de un modo cabal con los requisitos mencionados. La novedad en materia de patentes significa una

solución a un problema específico que no haya sido anteriormente conocida, e - indudablemente- no puede garantizarse que el método o esquema mental que caracteriza al software no haya sido antes imaginado por otro sujeto.

Por otro lado, el nivel inventivo impone que la solución otorgada a un problema específico no implique sólo la unión de ideas o métodos ya contenidas en el estado de la técnica, y el software se caracteriza por ser justamente eso, la combinación de operaciones mentales ya conocidas y manejadas en el mundo de la técnica. Finalmente, en la mayoría de los casos, los programas de computación no cumplen con la condición de aplicabilidad industrial ya que -generalmente- no aportan una solución a un problema de naturaleza técnica.

Resta indicar que ningún ordenamiento jurídico puede - válida y legítimamente- proteger y otorgar un título de propiedad con relación a ideas o conceptos abstractos, porque, tanto las ideas como los conceptos abstractos, son patrimonio común de la humanidad y resulta inimaginable la propiedad o exclusividad sobre los mismos que restringiese de cualquier modo su utilización. Por esto la patente de invención tiende a tutelar productos y procesos, más nunca teorías. (Moisset & Hiruela, 2010, pág. 2)

Desde esta perspectiva también se presenta una importante objeción en orden a tutelar al software por el régimen de patentes. Y se origina en que el programa de computación se elabora basándose en un algoritmo, el cual en esencia es una mera elaboración intelectual matemática y que -por tanto- pertenece al ámbito de las ideas de la humanidad.

En este sentido, al revisar la investigación de los tratadistas argentinos Luis Moisset de Espanés y María del Pilar Hiruela de Fernández, se advierte que la Corte Suprema de Justicia de los EE.UU. en el caso: *AGott Schalk v. Benson* ha sostenido que no se puede patentar una idea “(...) *la fórmula matemática implicada en el caso no tiene ninguna aplicación material, salvo en relación con un ordenador digital, lo que significa que si se admitiese su patentamiento, la patente cubriría la fórmula matemática y constituiría en la práctica una patente sobre el propio algoritmo.*” (Moisset & Hiruela, 2010, pág. 3)

Esto quiere decir que, el Alto Tribunal americano, rechazó -sin matices- la posibilidad de patentar el software. De igual forma y sobre la base de estas cuestiones, la Convención de Munich sobre la Patente Europea en 1973 excluyó expresamente a los programas de computación del campo de las invenciones patentables.

“Por otro costado, la exclusión de los programas de computación del significado de invención, y por ende de la protección dispensada por el derecho de patentes, la encontramos de manera expresa en las legislaciones de Francia, Alemania, Inglaterra, Brasil, Ecuador, Honduras, Guatemala, Méjico, Uruguay, entre otras”. (Moisset & Hiruela, 2010, pág. 3)

No obstante, lo expuesto, corresponde señalar que actualmente asistimos a una tendencia mundial que propugna la posibilidad de admitir -al menos en algunos casos- el patentizar software.

Quizás por la presión ejercida por las grandes multinacionales dedicadas a la

programación y comercialización de software se ha entendido que ésta es la modalidad idónea para la protección jurídica a los programas de computación ya que confiere un monopolio de explotación y habilita al inventor a oponerse a que otros sujetos no autorizados exploten su invención. En este orden de ideas, tanto la doctrina como la jurisprudencia han ido - progresivamente- perfilando argumentos para utilizar el régimen de patentes en la protección de programas de computación.

Sin embargo, no resulta ocioso señalar que, en todos los casos en que se ha recurrido al régimen de patentes se ha coincidido con que lo protegible por el título de patente es el conjunto inventivo que incluye al software como uno de sus elementos y no al programa de manera aislada. En otras palabras, lo protegido en estos supuestos excepcionales ha sido el invento que presupone la existencia de un software y no el programa de computación en sí mismo.

En cuanto a la protección jurídica que los Derechos de Autor le pueden dar al software se debe primero tener en cuenta que este régimen jurídico ha sido adoptado de modo ampliamente mayoritario por el derecho comparado y prácticamente no encuentra resistencias en el panorama mundial. Ha sido también la vía de tutela jurídica adoptada por los países signatarios del Acuerdo Sobre los Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio (ADIPC), conocidos como ATRIPS por sus siglas en inglés, celebrados en el marco de la Ronda Uruguay sobre el Acuerdo General sobre Aranceles Aduaneros y Comercio o GATT por sus siglas en inglés.

La preferencia por este sistema de protección se justifica en diversas razones y los tratadistas Luis Moisset de Espanés y María del Pilar Hiruela de Fernández lo

detallan en la siguiente exposición:

*“Por un lado, se evita la dispersión legal por la creación de nuevos instrumentos jurídicos, eludiéndose la inseguridad jurídica que provoca la introducción de nuevas normas hasta tanto se defina el sentido y alcance de las mismas. Igualmente, se garantiza la aplicación de un ordenamiento legal conocido. Las leyes y principios del copyright son bien conocidas a escala mundial. Por otro costado, se asegura la tutela internacional del programa ya que estará amparado tanto por la Convención de Berna, como por el ADIPC. En efecto, conforme al Convenio de Berna toda obra creada en un país es susceptible de recibir automáticamente idéntica protección en casi todos los países del mundo. Finalmente, se da acabada protección a la forma o manifestación de la idea expresada en el programa tutelado, y no se monopoliza el fondo de la misma, el cual -como se adelantará- es patrimonio de toda la humanidad y por tanto su titularidad no puede ser otorgada con exclusividad a una persona.”* (Moisset & Hiruela, 2010, pág. 4)

No obstante, este consenso existente a nivel internacional, alguna doctrina ha criticado la protección del software por el Derecho de Autor y a este respecto las principales objeciones se centran en:

*“Sostener que los programas de computación carecen de estética o belleza, que no son perceptibles de manera directa por los sentidos, que la protección jurídica otorgada por el copyright se prolonga demasiado en el tiempo, que el Derecho de Autor no da suficiente protección a las diversas etapas para la elaboración del software y que sus normas no prohíben el uso del software.”* (Moisset & Hiruela, 2010, pág. 4)

Empero, tales críticas han sido suficientemente respondidas. Así se ha dicho que la estética o belleza de la obra no es un elemento del Derecho de autor ya que es principio de éste que se protege la obra en sí misma sin entrar en la consideración de su valor estético o finalidad.

Igualmente, carece de toda trascendencia que la obra sea susceptible de ser directamente percibida por los sentidos. Un CD o un disquette sólo pueden ser percibidos con la previa lectura de una máquina y no por ello se niega protección a la obra que contienen. La cuestión de la duración de la protección, aun cuando atendible, no importa la exclusión de este régimen como inválido. Se trata de una cuestión de política legislativa y basta con que se reduzca el plazo de tutela, sin necesidad de recurrir a un régimen jurídico diverso. En relación a la crítica referida a ausencia de protección de los pasos o procesos de elaboración del software se ha dicho que éstos pueden válidamente ser tutelados por el Derecho de Autor siempre que cada una de las etapas constituya en sí misma una obra. Finalmente, la última de las censuras ha sido descartada por no ser cierta, afirmándose que el uso ilícito o no autorizado de la obra es absolutamente punible y reprochable por las normas del Derecho de Autor. (Moisset & Hiruela, 2010, pág. 4)

Otro mecanismo de protección que también puede efectivizar una eficiente tutela al software es el secreto industrial, porque este tipo de protección cubre de manera muy amplia todo ataque dirigido a apoderarse de las conquistas que la empresa ha realizado para la elaboración de un software, sea que el ataque provenga de la deslealtad de empleados o funcionarios, sea que provenga de un adquirente del programa vinculado mediante un contrato con la empresa, sea que provenga de

maniobras de espionaje o de cualquier otro tipo de conducta semejante que haya conducido a la apropiación indebida del secreto industrial.

En el common law esta modalidad de protección tiene una importante aplicación. Sobre el punto, resta indicar que la protección de los programas de computación por el secreto industrial actúa en un terreno diverso al ámbito propio del Derecho de Patentes y del Derecho de Autor.

Por el secreto industrial se protege la confiabilidad de los datos del funcionamiento del programa; por el Derecho de Patentes -en cambio- el software se hace público, pero se prohíbe su reproducción. Aun cuando en el Derecho de Autor presenta otros matices, ya que se protegen incluso las obras inéditas, lo cierto es que este ordenamiento no tiene por objeto la tutela del secreto sino la protección de la titularidad de la obra.

En definitiva, el secreto industrial protege la confidencialidad y reprocha jurídicamente todo ataque que se pueda realizar contra la buena fe. En el Derecho de Patentes, así como en el Derecho de Autor, lo protegido es la propiedad del titular de la obra, impidiendo su reproducción no autorizada por un tercero.

Los Diseños y modelos también son otros mecanismos que otorga la legislación de propiedad intelectual, con esta vía de protección se procura tutelar las representaciones visuales de algunos programas de computación. Aun cuando podría objetarse que las figuras o representaciones se fijan en la pantalla temporariamente y se modifican en el curso de la ejecución del programa, lo cierto es que se admite válidamente tutelar por este régimen algunas imágenes características o algunos

gráficos de control de stocks o de contabilidad.

Finalmente, la Marca o Nombre y el derecho especial son nuevas formas de tutelar al software, porque el primero da nombre a los programas y así puede ser protegido mediante este sistema, asegurando la exclusividad y propiedad de la marca o de la denominación del software. Mientras que en el Derecho Especial algunas posiciones aisladas han sostenido que los programas de computación deben ser protegidos mediante un sistema jurídico *sui generis* en razón de la complejidad de los programas.

Esta corriente, propugna la elaboración de un régimen legal especial y original que, tomando los elementos más significativos del Derecho de Autor, del Derecho de Patentes y de los demás instrumentos jurídicos de tutela existentes, atienda a las singularidades del software y constituya una estructura normativa particular y distinta.

La tesis bajo la lupa, mantenida a nivel doctrinario, aunque cuestiona la protección dada por el Derecho de Autor y de Patentes, se ve obligada a reconocer que la adopción de un derecho especial no podría -a nivel internacional- articularse sobre un tratado vigente, lo que en definitiva la torna poco aconsejable.

En Ecuador la protección del software o programas de ordenador no se la considera desde la protección de derechos de autor y de patentes, pues al software se lo conceptualiza como *“una obra intelectual sui generis que requiere una protección específica, ya que constituye el resultado de un esfuerzo creativo, de inversión de tiempo y dinero”*. (Sánchez, 2005). La Ley de Propiedad Intelectual

define al programa de ordenador (software) como:

*“Toda secuencia de instrucciones o indicaciones destinadas a ser utilizadas, directa o indirectamente, en un dispositivo de lectura automatizada, ordenador o aparato electrónico o similar con capacidad de procesar información, para la realización de una función o tarea, u obtención de un resultado determinado, cualquiera que fuere su forma de expresión o fijación. El programa de ordenador comprende también la documentación preparatoria, planes y diseños, la documentación técnica y los manuales de uso”.* (Ley de Propiedad Intelectual, 2016)

Así mismo la propia Ley, en su artículo 28 señala que:

*“Los programas de ordenador se consideran obras literarias y se protegen como tales. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresado, ya sea en forma legible por el hombre (código fuente) o en forma legible por la máquina (código objeto), ya sean programas operativos y programas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia y organización del programa.”* (Ley de Propiedad Intelectual, 2016)

Desde esta perspectiva, y tomando como referencia el análisis realizado por la Dra. Verónica Sánchez González para la Alianza antipiratería se puede decir que con la adquisición de un programa de ordenador al usuario o propietario se le autoriza a realizar única y exclusivamente:

*“a) Una copia del programa con fines de seguridad*

b) *Fijar el programa en la memoria interna del aparato, para su utilización*

c) *El uso normal previsto en la licencia.*” (Sánchez, 2005)

Según la Ley de propiedad Intelectual *“Para cualquier otra utilización inclusive la reproducción para fines de uso personal o el aprovechamiento del programa por varias personas a través de redes u otros sistemas análogos”*, (Ley de Propiedad Intelectual, 2016) se requiere de la autorización del titular de los derechos, autorización que se traduce por lo general en una licencia de uso. De no respetar esto dentro de nuestra legislación se considera como infracciones a los derechos de propiedad intelectual.

La infracción a cualquiera de los derechos de propiedad intelectual, da lugar al ejercicio de las acciones legales previstas en la Ley de Propiedad Intelectual y pueden ser civiles, administrativas y penales.

Entre las sanciones en vía civil de conformidad con el Art. 304 de la Ley de Propiedad Intelectual se contempla que, al infractor de estos derechos se le puede imponer una multa que va de *“tres a cinco veces el valor total de los ejemplares de obras ...o de las regalías que hubiere percibido el titular de los derechos por su explotación legítima...”*, (Ley de Propiedad Intelectual, 2016) más la indemnización por daños y perjuicios causados.

Entre las sanciones en la vía penal se puede establecer que estas infracciones son perseguibles de oficio y son reprimidas con una pena pecuniaria.

Y finalmente en las sanciones administrativas se procede una multa y la

adopción de cualquiera de las medidas cautelares previstas en la Ley y que consisten en:

- a) El cese inmediato de la actividad ilícita,*
- b) La suspensión de la actividad, la utilización, exportación, venta, oferta en venta, exportación, importación, reproducción, comunicación, distribución, según proceda, e incluso*
- c) Cualquier otra que evite la continuación de la violación de los derechos.*

(Ley de Propiedad Intelectual, 2016)

Como queda determinado la tutela del software por su naturaleza es muy compleja, en la actualidad todavía existe una discusión sobre qué mecanismos de derecho intelectual sería el más correcto para emplearlo, o si es necesario emprender en nuevas formas de legislación para su efectiva protección y tutela, lo que si queda claro es que actualmente las relaciones de explotación comercial de software en la mayoría de países y particularmente en el nuestro el software se lo maneja mediante licenciamientos que individualizan su programación y ejecución, haciendo problemático su desarrollo y su transferencia entre empresas que necesitan de su empleo para su normal desenvolvimiento.

## **2.2. Características y naturaleza del licenciamiento en el software**

En la gran mayoría de los casos, el software se entrega con una licencia de uso. Dicha licencia es un contrato entre el productor y el usuario que establece cuáles son los derechos y obligaciones de cada una de las partes. Según el análisis realizado por los tratadistas Juan Carlos Brocca y René Casamiquela para la Universidad

Nacional del Comahue se debe tener en cuenta que:

*“Al instalar, utilizar o copiar un producto de software bajo licencia, el usuario está aceptando las condiciones estipuladas en la misma y queda obligado por los términos de dicho contrato. Es importante, entonces, leer detenidamente la licencia de un programa para conocer las condiciones para su utilización. Debe tenerse en cuenta que algunas licencias son tan restrictivas que incluso el producido con la herramienta en cuestión no es de libre disponibilidad, mientras que otras invitan a su modificación o duplicación”.* (Brocca & Casamiquela, 2006)

En otras palabras, *“la licencia de software es una especie de contrato, en donde se especifican todas las normas y cláusulas que rigen el uso de un determinado programa, principalmente se estipulan los alcances de uso, instalación, reproducción y copia de estos productos.”* (Muro, 2007, pág. 12)

El tema de las licencias de software puede ser muy complejo. El negocio del software se basa en licencias binarias. La propiedad intelectual de los distribuidores de software comercial nace del código fuente. *“Las licencias de software se crean con diversos fines empresariales y para afrontar diversos tipos de relaciones (como distribuidor/cliente). Los desarrolladores de software tanto comercial como no comercial utilizan decenas de licencias que abarcan una gran variedad de términos y condiciones.”* (Muro, 2007, pág. 12)

Dado este panorama, es común que las grandes empresas dispongan de sistemas que poseen altos costos de mantenimiento, actualización, capacitación o soporte; que muchas veces superan el costo de obtención de la licencia. Por otra

parte, han surgido cada vez con mayor fuerza programas de código libre amigables para el usuario del hogar que le permiten abaratar costos en desmedro de otro software comerciales con altos costos en licencias.

Es importante conocer cómo afectan estas licencias al trabajo de otras personas, además las licencias de uso de software generalmente caen en alguno de estos tipos:

- ***Licencia propietaria.*** *Uso en una computadora por el pago de un precio.*
- ***Shareware.*** *Uso limitado en tiempo o capacidades, después pagar un precio.*
- ***Freeware.*** *Usar y copiar ilimitado, precio es cero.*
- ***Software libre.*** *Usar, copiar, estudiar, modificar, redistribuir. Código fuente incluido. (Muro, 2007)*

Es posible dividir las licencias de software libre en dos grandes familias. Una de ellas está compuesta por las licencias que no imponen condiciones especiales, sólo especifican que el software se puede redistribuir o modificar. Estas son las llamadas licencias permisivas. La otra familia, denominadas licencias robustas o licencias copyleft, imponen condiciones en caso de que se quiera redistribuir el software, condiciones que van en la línea de forzar a que se sigan cumpliendo las condiciones de la licencia después de la primera redistribución. (Muro, 2007)

Mientras que el primer grupo hace énfasis en la libertad de quien recibe un programa, ya que le permite hacer casi lo que quiera con él (en términos de las

sucesivas redistribuciones), el segundo obliga a que las modificaciones y redistribuciones respeten los términos de la licencia original.

Entrando en tema de análisis acerca de las características y naturaleza de las licencias de software se debe manifestar que básicamente su naturaleza se encuentra en que una licencia rige la utilización del software, *“es decir, en ningún momento un usuario compra un programa o se convierte en propietario de él, tan sólo adquiere el derecho de uso, incluso así haya pagado por él.”* (Hispavista, 2011)

*“Las condiciones bajo las cuales se permite el uso del software, o sea las licencias, son contratos suscritos entre los productores de software y los usuarios. En general, las licencias corresponden a derechos que se conceden a los usuarios, principalmente en el caso del software libre, y a restricciones de uso en el caso del software propietario. Las licencias son de gran importancia tanto para el software propietario como para el software libre, igual que cualquier contrato.”* (Hispavista, 2011)

*“Un caso especial, en lo que concierne a la propiedad sobre el software, lo constituyen los programas denominados de dominio público, porque sus creadores renuncian a los derechos de autor.”* (Peña, 2008) Para percibir de mejor forma vamos hacer referencia al estudio realizado por el tratadista Colombiano Daniel Peña Valenzuela en su obra *“Software Libre y Software Propietario: Impacto Jurídico, Económico y Cultural en Colombia”* quien para advertir las características del licenciamiento de software primero determina los elementos que se licencian.

*“De ahí que normalmente se concede permiso de utilización de los archivos*

*binarios junto con sus manuales, sea que estos últimos se entreguen en papel o en formato digital, pero ahora que el software libre está en auge cada vez es más frecuente recibir los archivos fuentes de los programas y la autorización para analizarlos y modificarlos, si se desea. Es decir, los elementos que se pueden licenciar son:*

- *Archivos binarios.*
- *Archivos fuentes.*
- *Manuales y documentación relacionada.” (Peña, 2008)*

*“Normalmente las licencias del software se aplican al software considerado como un todo, es decir, la documentación que acompaña al software es considerada como parte del software. La mayor parte del software, incluso se podría decir, todo el software, incluye alguna documentación.” (Peña, 2008)*

*“Como se mencionó, los productores de software propietario, shareware, de demostración y freeware, normalmente sólo entregan los archivos binarios. En cambio, quienes desarrollan software libre y semi-libre distribuyen tanto los archivos binarios como los archivos fuentes, aunque existe toda una variedad en los términos bajo los cuales se entregan los archivos fuentes.” (Peña, 2008)*

También dentro de la caracterización y naturaleza de las licencias de software se debe analizar los usos permitidos y no permitidos del software, porque *“los usos que se le puede dar al software son diversos, como se ha descrito. El uso natural es la ejecución de los archivos binarios, pero a este se le suma la copia, la distribución e incluso el análisis y modificación de los archivos binarios o fuentes”.* (Peña, 2008)

Lo primero en advertirse es el Permiso de ejecución que se caracteriza por ser de uso personal:

*“El permiso de ejecución en ocasiones se limita a una persona. Se dice entonces que la licencia es para uso personal. Esta situación es típica de productores de software propietario que quieren difundir un producto nuevo o rescatar un producto que está siendo desplazado por la competencia. Un ejemplo del primer caso es el paquete StarOffice v.5.0 de la empresa Star Division. StarOffice es uno de los denominados paquetes integrados de oficina, equivalente a Microsoft Office, que se está abriendo camino en el mercado. En cambio, Word Perfect v.8.0 para el sistema operativo Linux, distribuido por Corel, es una versión de este popular procesador de palabra, que busca nuevos horizontes. Tanto StarOffice como Word Perfect se pueden adquirir de Internet con licencia para uso personal, pero también existen versiones sin esta limitación.” (Peña, 2008)*

Según el estudio de este blog de internet, *“el uso personal implica que sólo una persona puede ser utilizadora, es decir, si se instala una versión de estos programas en un hogar, de manera estricta, los restantes integrantes de la familia no podrán hacer uso de ellos.” (Peña, 2008)*

*“Como se ve esta restricción al menos en teoría tiene sus consecuencias. Normalmente esta limitación se manifiesta en forma explícita, cuando de entrada el nombre del producto así lo indica, por ejemplo: "Word Perfect Personal Edición", en cambio, no faltan las excepciones y sorpresas en software que se piensa puede ser utilizado por cualquier persona de manera no simultánea en un determinado*

computador.” (Peña, 2008)

*“Por ejemplo, una licencia de Microsoft Visual FoxPro Edición Profesional dice: "Microsoft le otorga a usted como individuo, una licencia personal, no exclusiva, para que haga y use copias del SOFTWARE"... "Si usted es una entidad, Microsoft le otorga el derecho de designar a un individuo dentro de su organización para que tenga el derecho a usar el SOFTWARE en la forma arriba señalada. Las licencias de uso personal que se conceden sin que los usuarios tengan que pagar pueden ser aceptables, pero, ¿qué decir de software propietario para uso personal que se adquiere pagando?; sin duda, restricciones de este tipo son muestras de los abusos que se cometen en contra de los usuarios.”*

(Peña, 2008)

El segundo segmento que advierte que en el uso comercial; no existe una definición precisa o universal de lo que entiende por comercial o no comercial. Así, por ejemplo, de acuerdo a lo que se lee en los términos de algunas licencias, por comercial, se puede entender como:

*“Toda situación en la cual se obtiene un lucro o beneficio económico ya sea en forma directa o indirecta. Por ejemplo, alguien que se dedica a transcribir documentos se está beneficiando directamente del uso del software. Toda actividad productiva, aunque no necesariamente genere beneficio económico inmediato para el utilizador del software. Por ejemplo, un docente puede utilizar software para ser más eficiente en su trabajo, pero continúa recibiendo un mismo salario. El problema de las licencias que prohíben el uso*

*comercial es que normalmente no se define el término, limitándose a consignar en forma escueta, por ejemplo, "libre para uso no comercial". En estos casos, la única alternativa es preguntar, preferiblemente por escrito, a los autores. Normalmente el uso de software en establecimientos educativos es considerado como comercial. El mismo análisis que se hizo para el término comercial es válido para el término ánimo de lucro." (Hispanista, 2011)*

*El tercer segmento es el uso simultáneo, "ya que algunos programas se prestan para utilizarse simultáneamente haciendo uso de una red local de cómputo, la mayoría de productores de software propietario prohíben este tipo de uso, a menos que esa sea la forma normal de utilización del software, caso en el cual es necesario adquirir varias licencias. En ciertas situaciones es posible instalar el software en varios computadores siempre y cuando no se utilice simultáneamente en mayor número que el permitido." (Peña, 2008)*

*La cuarta característica se da con el "uso para determinados grupos sociales; en ocasiones se permite o se prohíbe el uso del software a determinados grupos sociales. Por ejemplo, en ocasiones el software propietario puede utilizarse libremente en entornos educativos. Por el contrario, en otros casos poco comunes, se prohíbe el uso de un programa, por ejemplo, a personas vinculadas a las fuerzas militares. El uso libre para el sector educativo, al igual que el uso personal y el uso no comercial, ya comentados, generalmente son meramente estrategias comerciales. En cambio, el caso de las fuerzas militares puede obedecer más a razones filosóficas que a otra cosa." (Peña, 2008)*

La quinta característica es el uso privado, pues algunos programas permiten libertad de uso privado, es decir, uso exclusivo en casa en un ambiente familiar. *“El uso privado en ciertos casos es equivalente al uso personal.”* (Peña, 2008)

La sexta característica se da en el *“uso en un sólo computador, en la cual los productores de software propietario normalmente permiten instalar el software en un sólo computador (obviamente sin considerar el software diseñado para trabajar en red), más aún, prohíben tener más de una copia instalada en un mismo computador.”* (Peña, 2008)

La séptima característica posible en el licenciamiento de software es cuando se otorga permiso de copia, porque *“como ya se mencionó normalmente los productores de software propietario únicamente permiten al usuario realizar una copia como respaldo del software, sea que se entregue discos de instalación o sea que venga preinstalado.”* (Peña, 2008)

*“No todo el software que viene preinstalado viene con licencia de uso, incluso, lo que normalmente ocurre es lo contrario, es decir, es costumbre de los vendedores entregar software preinstalado sin licencia en computadores que no son "de marca". Cuando el software propietario se entrega con licencia, pero preinstalado en los discos duros, si ocurre algún daño en un disco, el usuario se encontrará en problemas para volver a instalar los programas, por carecer de la copia de respaldo. Desde que los discos tipo CD-ROM entraron en vigencia, cada vez es más común poseer software en estos medios. En teoría estos discos tienen una vida útil de muchos años si se tratan con un mínimo de cuidado, sin*

*embargo, cuando se trata de software importante es preferible disponer de una copia de respaldo a la mano. Retomando el tema, al contrario del software propietario, como se mostró en la tabla anterior, el software libre y el software shareware se pueden copiar libremente, e incluso generalmente se anima a ello. Normalmente el permiso de copia se aplica de igual manera a la documentación del software, ya sea en beneficio de los usuarios cuando es permitida, o como limitante cuando no lo es.”* (Peña, 2008)

El permiso de distribución también es una característica de las licencias de software, porque “la redistribución de software es una actividad que, por ejemplo, llevan a cabo empresas que ofrecen libros o revistas que incluyen CD-ROMs. La redistribución es una de las formas más comunes de entrega de software.” (Peña, 2008)

*“La redistribución de software propietario también es permitida en algunos casos para ciertos archivos necesarios para el funcionamiento de programas elaborados por los usuarios. Por ejemplo, cuando se desarrollan aplicaciones bajo Microsoft Visual Basic, estas aplicaciones requieren de ciertas bibliotecas de funciones. En este caso, para que el software desarrollado por un usuario pueda ser distribuido y ejecutado en otros computadores, las empresas que licencian el software de desarrollo, permiten la libre distribución de las "bibliotecas" mencionadas.”* (Peña, 2008)

Finalmente, el último carácter que se puede mostrar en las licencias de

software es el *“permiso de modificación de archivos fuentes, los permisos de modificación de archivos fuentes son propios del software libre o semi-libre, por ejemplo, aquellos que se distribuyen bajo la Licencia Pública General. La entrega de los archivos fuentes junto con los archivos ejecutables es una acción que últimamente se está presentando en software propietario de varias empresas, como una estrategia de desarrollo de sus productos. La modificación de archivos fuentes es de poca o ninguna importancia para un usuario común y corriente.”* (Peña, 2008)

Una característica que se da generalmente en las licencias de software propietario es las *“restricciones generales, pues los contratos de licencia de software, en especial del software propietario, son más restricciones que otra cosa. Adicionalmente a las restricciones de copia, distribución y ejecución mencionadas, es común encontrar prohibiciones de:*

- *Alquilar, vender, o prestar el software*
- *Realizar ingeniería inversa al software*
- *Alterar o desensamblar el software*
- *Exportar - el software”* (Peña, 2008)

*“El software no se puede revender o sub licenciar. En este caso la única alternativa es la cesión o transferencia de la licencia. Por otro lado, normalmente se sabe que la ingeniería inversa está prohibida, sin embargo, la ley la permite en contados casos y en Europa está permitida cuando se hace para lograr mayor compatibilidad o interoperabilidad.”* (Peña, 2008)

*“Así, por ejemplo, Samba es software que se ha desarrollado de esa forma.*

*De todas las restricciones indicadas, causa extrañeza aquella que prohíbe el préstamo, porque, por ejemplo, cuando se compra un libro lo más natural es disponer de él como a bien se tenga. ¿Por qué se ha dejado que el uso del software se restrinja hasta este extremo?” (Peña, 2008)*

Otra características en las licencias de uso del software es que la garantía limitada o ausencia de garantía de los productos, porque *“normalmente no se ofrece ninguna garantía del correcto funcionamiento del software que se licencia, o se ofrece una garantía muy limitada. Esta característica es comprensible cuando se trata de software que se adquiere sin pagar, pero en este caso también la ley está a favor de los productores de software propietario.” (Peña, 2008)*

También en la forma de presentación de las licencias se puede ver otra característica propia de estos sistemas de regulación contractuales, pues igual que la *“documentación, físicamente las licencias pueden corresponder a material impreso o archivos en medios digitales, es decir, pueden venir en papel o en disco, o en los dos medios.” (Peña, 2008)*

*“De cualquier forma, en algunos casos los términos de las licencias no mencionan explícitamente a los productos en particular, utilizando en su lugar términos genéricos como "EL PRODUCTO" o "EL SOFTWARE". Por esta razón, cuando las licencias vienen en papel, es bueno escribir sobre ellas los nombres de los productos adquiridos y guardarlas en un lugar seguro, para evitar contratiempos futuros. Respecto al software que se obtiene de Internet, en ciertas ocasiones algunos programas pequeños no traen una licencia de uso formalmente escrita sino alguna*

*descripción del tipo de uso y distribución permitidos.” (Peña, 2008)*

Finalmente, las características que por su esencia misma trae una licencia está en su validez y en esta misma validez para su traducción, *“las licencias permanecen vigentes en el tiempo, sin importar los virajes que puedan sufrir las licencias posteriores correspondientes a nuevas versiones de los programas. Si no se adquieren las actualizaciones, los términos de las licencias continúan siendo válidos.” (Peña, 2008)*

*“Puede ocurrir que las licencias de las nuevas versiones del software reemplacen a las anteriores, pero sólo si son más favorables para los usuarios. Así, por ejemplo, la licencia de Netscape Communicator 4.5 Edición Estándar, permite el uso del programa para fines comerciales o no comerciales, lo que no ocurría en las primeras versiones. Para poder utilizar las versiones anteriores sin problema es necesario entonces obtener una copia de la nueva licencia, más por precaución, ya que esto es de conocimiento público.” (Peña, 2008)*

La validez de las traducciones de las licencias en cambio se refiere al uso de estos productos para *“ser traducidas a otros idiomas diferentes al país de origen, sin embargo, ninguna de ellas tendrá valor legal, siendo necesario remitirse siempre a la original Incluso, si se llegase a presentar un reclamo, en algunos casos se debe presentar la demanda exclusivamente en el idioma nativo y bajo las leyes locales a las cuales se acoja el productor.” (Peña, 2008)*

Todas estas características inherentes a los licenciamientos y la misma

naturaleza de estos instrumentos dentro de contratos han ocasionado *“dificultades en la interpretación de las licencias, la principal dificultad para la interpretación de las licencias suele ser el idioma. Incluso, si la licencia viene en idioma español, aún puede haber inconvenientes en su interpretación, debido a la terminología propia de abogados que muchas veces se emplea.”* (Peña, 2008)

*“Pese a que no son palabras propias de letrados, particularmente los términos "comercial" y "sin ánimo de lucro", se prestan para malas interpretaciones. Como se consignó en estos casos lo mejor es contactar y preguntar directamente al desarrollador del software. Finalmente, otro gran obstáculo en la interpretación de las licencias se debe a la irresponsabilidad de ciertos distribuidores de software, que se valen de la palabra "free" para promocionar software tipo shareware o de demostración, lo que causa confusión.”* (Peña, 2008)

Sin embargo, en la actualidad se muestra una gran variedad de licencias de software que tratan de regular el sector de su desarrollo y transferencia a nivel planetario, o cual hace necesario pasar a analizar los tipos y forma de licenciamiento de software.

### **2.2.1. Formas o tipos de licenciamiento de Software**

Las licencias presentan algunos aspectos que permiten realizar una clasificación del software en relación con éstas. Este agrupamiento no es único, dado que es frecuente encontrar la utilización tendenciosa de los distintos términos, como también la asignación de significados diferentes.

De acuerdo con la normativa vigente, la utilización del software puede ser legal, si cumple con los requerimientos establecidos para la obtención de la licencia y se emplea de acuerdo con lo establecido en ella; o ilegal, cuando es adquirido en desarreglo con las condiciones estipuladas o se utiliza para fines no contemplados en el contrato.

Con relación al costo que representa la obtención del producto, el software puede ser gratuito: no requiere erogación monetaria o a lo sumo una equivalente al valor del soporte que lo contiene; u oneroso: donde el usuario debe adquirirlo efectuando un pago al proveedor.

En cuanto a la comercialización, si bien no existe una única interpretación al respecto, las licencias pueden establecer que el software sea comercial o no comercial. En el primer caso, quienes lo producen lo hacen como parte de una actividad económica y por lo tanto obtienen un beneficio por el producto, su distribución o soporte. En el segundo, lo hacen sin fin de lucro. Corresponde aclarar que las empresas desarrolladoras de software más importantes a nivel mundial -que obtienen sus ingresos a partir de la venta de licencias de programas- establecen claramente que sus productos son comerciales e interpretan que todo software que pueda obtenerse de una manera diferente es no comercial.

Algunas licencias establecen condicionamientos al usuario en relación con la utilización que puede dar al producto. Por eso, es frecuente encontrar el término de software de uso comercial, que no establece limitaciones en cuanto a las actividades productivas realizadas con el programa. Los de uso no comercial son aquellos en los que la licencia exige una utilización predeterminada, por ejemplo, la versión

educativa de un producto, impide el empleo del software en otra actividad.

Cuando las licencias permiten la distribución del código fuente de la aplicación, se clasifica al software en abierto. Es cerrado, si la licencia contempla solamente la entrega del código ejecutable.

Pero tal vez sea la característica más importante la que considera los derechos concedidos a los usuarios con relación al uso, modificación, copia y distribución. En este caso el software puede ser privativo, cuando las licencias plantean restricciones al respecto o libre, en aquellos casos en que otorgan plenas libertades a los usuarios. Nótese que para que un software pueda considerarse libre, necesariamente debe ser abierto, dado que para ejercer derechos de modificación es necesario contar con el código fuente del programa.

Dentro de esta clasificación es donde vamos a analizar el tipo de licenciamiento, porque *“si bien cada programa viene acompañado de una licencia de uso particular, existen diversos aspectos en común entre las licencias que hacen posible su clasificación. De acuerdo a ello, es común encontrar términos tales como software shareware, freeware, de dominio público, o de demostración. A estos nombres hay que agregar software libre y software propietario, términos un tanto desconocidos pero que se usan en medios informáticos. Incluso, es posible hablar de software semi-libre.”* (Hispanista, 2011)

Según un análisis realizado por el blog realizado por hispana el uso y la relación que un software tiene con los usuarios se puede clasificar en: software propietario y libre, y dentro de estos segmentos hay unos tipos de software que

también pueden ser clasificables. A continuación, se deja expuesto en base al análisis del Doctor Amaya este tipo de clase de software:

**“Software propietario:** *En términos generales, el software propietario es software cerrado, donde el dueño del software controla su desarrollo y no divulga sus especificaciones. El software propietario es el producido principalmente por las grandes empresas, tales como Microsoft y muchas otras. Antes de poder utilizar este tipo de software se debe pagar por su licencia. Cuando se adquiere una licencia de uso de software propietario, normalmente se tiene derecho a utilizarlo en un solo computador y a realizar una copia de respaldo. En este caso la redistribución o copia para otros propósitos no es permitida.*

**Software shareware o de evaluación:** *El software tipo shareware es un tipo particular de software propietario, sin embargo, por la diferencia en su forma de distribución y por los efectos que su uso ocasiona, puede considerarse como una clase aparte.*

*El software shareware se caracteriza porque es de libre distribución o copia, de tal forma que se puede usar, contando con el permiso del autor, durante un periodo limitado de tiempo, después de esto se debe pagar para continuar utilizándolo, aunque la obligación es únicamente de tipo moral ya que los autores entregan los programas confiando en la honestidad de los usuarios. Este tipo de software es distribuido por autores individuales y pequeñas empresas que quieren dar a conocer sus productos. Muchas veces por ignorancia los programas de esta clase se utilizan ilegalmente. A menudo el software shareware es denominado como software de evaluación.*

**Software de demostración:** *No hay que confundir el software shareware con el software de demostración, que son programas que de entrada no son 100%*

*funcionales o dejan de trabajar al cabo de cierto tiempo. También estos programas son los que se consiguen en los quioscos de periódicos y revistas. El software de demostración o como se acostumbra a decir "software demo", es similar al software shareware por la forma en que se distribuye, pero en esencia es sólo software propietario limitado que se distribuye con fines netamente comerciales.” (Amaya, 2010)*

Por otro lado, el mismo blog de internet divide al software libre en pequeños tipos y clase de productos que también pasamos a exponerlos para mejor comprensión del trabajo investigativo:

***“Software libre:** El software libre es software que, para cualquier propósito, se puede usar, copiar, distribuir y modificar libremente, es decir, es software que incluye archivos fuentes. La denominación de software libre se debe a la Free Software Foundation (FSF), entidad que promueve el uso y desarrollo de software de este tipo. Cuando la FSF habla de software libre se refiere a una nueva filosofía respecto al software, donde priman aspectos como especificación abierta y bien común, sobre software cerrado y ánimo de lucro esto no impide que el software libre se preste para que realicen negocios en su entorno.*

***Software de dominio público:** El software de dominio público (public domain software), es software libre que tiene como particularidad la ausencia de Copyright, es decir, es software libre sin derechos de autor. En este caso los autores renuncian a todos los derechos que les puedan corresponder.*

***Software semi-libre:** Para la FSF el software semi-libre es software que posee*

*las libertades del software libre pero sólo se puede usar para fines sin ánimo de lucro, por lo cual lo cataloga como software no libre.*

**Software freeware:** *El software freeware es software que se puede usar, copiar y distribuir libremente pero que no incluye archivos fuentes. Para la FSF el software freeware no es software libre, aunque tampoco lo califica como semi-libre ni propietario. El software freeware se asemeja más al software libre que al software shareware, porque no se debe pagar para adquirirlo o utilizarlo.*

*Síntesis de los tipos de software según su licencia: Los diferentes tipos de software según su licencia pueden agruparse de varias formas, por ejemplo, por la disponibilidad de los archivos fuentes o por el costo que representa para el usuario.” (Amaya, 2010)*

*“También es posible agrupar el software según los fines que persigue, aunque en este caso el resultado no ayuda mucho porque lo que interesa es diferenciar el software propietario del software libre. (Hispavista, 2011)*

*“Según la disponibilidad de los archivos fuentes el software puede agruparse en abierto (libre, de dominio público y semi-libre) y cerrado (freeware, shareware, de demostración y propietario).” (Hispavista, 2011)*

De acuerdo al costo que representa para el usuario el software puede agruparse en software gratuito (libre, de dominio público, semi-libre y freeware) y en software no gratuito (shareware, de demostración y propietario).

Para una mejor comprensión acerca del tipo de software según sus características encontramos pertinente exponer un gráfico que ilustra de mejor forma esta clasificación según el tipo de uso, precio y funcionalidad del software:

Tipo de licencias de software	Permiso de copia y redistribución	Precio cero	100% funcional	Permiso de uso ilimitado en el tiempo	Disponibilidad y permiso de modificar archivos fuentes
propietario	No	No	si	si	no
shareware o de evaluación	Si	No	si	no	no
de prueba o demostración	Si	si	no	no	no
freeware	Si	si	si	si	no
Libre	Si	si	si	si	si

**Cuadro N° 1 Tipo de uso, precio y funcionalidad de las licencias de software. Fuente: Página web: <http://estuinge.galeon.com/legalidad.htm>**

*“Es interesante exponer que desde 1998 para acá varias de las grandes compañías han decido entregar archivos fuentes, influenciados por la*

*filosofía del software libre. El software semi-libre y de dominio público también cumplen con estas características. En definitiva, la disponibilidad del código fuente no es exclusiva del software libre, por lo cual no se debe asociar el software que dispone de archivos fuentes con el software libre, tampoco se debe utilizar el término software freeware para hacer referencia al software libre. La copia y redistribución son palabras en cierta forma similares, sólo que la redistribución es una copia que se hace a gran escala, sea cual fuere el fin que se persiga.” (Hispanista, 2011)*

Como queda advertido existen tipos de software los cuales guardan características inherentes dentro de estos dos grandes segmentos denominados software propietario y software libre, de ahí que a continuación es importante emprender en el análisis más sucinto de dos tipos de software libre que por sus características pueden ser convenientes para una transferencia efectiva de tecnología informática en nuestro país.

### **2.2.2. Análisis sobre las formas de licenciamiento Creative Commons y Licencia Pública General de GNU o (GPL)**

Una vez expuesto la clasificación de software libre y propietario y sus tipos es importante analizar dos tipos de licenciamiento que pueden transformarse en verdaderos instrumentos que garanticen una efectiva transferencia de tecnología y que por su parte fundamente una nueva legislación a futuro para este sector.

De ahí que la forma de licenciamiento denominada como Creative Commons

se hace efectiva cuando un docente decide producir materiales educativos con la intención de mejorar su tarea educativa, está creando una obra sujeta a la normativa sobre propiedad intelectual y derechos de autor. Automáticamente esta obra genera unos derechos de copia cuyo titular es el autor de la misma y entre otras cosas prohíbe la reproducción y distribución de la totalidad o parte de la misma sin autorización expresa de los propietarios del citado Copyright, en el caso que el autor decida poner a disposición del público en general la citada obra, esta deberá autorizarse explícitamente para cada uso que vaya a hacerse de ella o se estará contraviniendo la ley.

Para evitar tener que andar licitando cada uso que vaya a hacerse y facilitar la divulgación de obras de propiedad intelectual de carácter libre, se creó el proyecto Creative Commons en 2001 a raíz del caso Bono y la Copyright Term Extension Act, en que la viuda del señor Sonny Bono pedía una ampliación del periodo de CopyRight. (Aliprandi, 2012)

Si la regla por defecto del copyright es “todos los derechos reservados”, el significado expreso de una licencia de Creative Commons es que “algunos derechos están reservados.” Por ejemplo, la ley del copyright brinda al poseedor del copyright el derecho exclusivo de hacer “copias” de su trabajo. Una licencia de Creative Commons, podría, en efecto, anunciar que este derecho exclusivo fue otorgado al público. (Aliprandi, 2012)

Según el estudio de la tratadista Italiana Simone Aliprandi, las licencias Creative Commons garantizan cuatro libertades; atribución de la obra, no comercial, sin derivados y compartir igual. De igual forma las combinaciones de éstas dan lugar

a once tipos de licencia distintas, aunque las más habituales son las seis siguientes:

**1- Reconocimiento:** *Use esta obra como desee pero indique la autoría de la misma.*

**2- Reconocimiento-Compartir igual:** *Use la obra como desee, indicando la atribución de la misma y bajo el mismo tipo de licencia.*

**3- Reconocimiento-Sin derivados:** *Utilice la obra tal y como es, otorgando el reconocimiento al autor.*

**4- Reconocimiento-No comercial:** *Utilice la obra con propósitos no comerciales y otórgueme la atribución.*

**5- Reconocimiento-No comercial-Sin derivados:** *Utilice la obra tal y como es, con propósitos no comerciales y con atribución al autor.*

**6- Reconocimiento-No comercial-Compartir igual:** *Utilice la obra con propósitos no comerciales, bajo el mismo tipo de licencia y con atribución.*

(Aliprandi, 2012)

Tal y como vemos, el reconocimiento está presente en todas las licencias, esto indica que cuando alguien usa una de nuestras obras bajo alguna de estas seis licencias Creative Commons debe, obligatoriamente, reconocer expresamente la autoría de la misma en los créditos de su obra y esto nos incluye a nosotros mismos cuando utilizamos cualquier tipo de obra bajo cualquiera de estas licencias Creative Commons.

En cuanto a la ventajas y desventajas que este tipo de licenciamiento de

software ofrece en la transferencia de tecnología se puede afirmar que según el tratadista Simone Aliprandi especialista en el tema de Creative Commons, estas son:

***Ventajas del Creative Commons.-***

- *Hay un beneficio mutuo entre creador y usuario (beneficio compartido).*
- *Hay una probabilidad mayor de ser citado como autor.*
- *Menor posibilidad que el usuario haga un mal uso del contenido de la obra.*
- *Es gratuito.*
- *Presenta diferentes alternativas de licencia.*
- *No queda excluida la posibilidad de obtener un permiso expreso del autor de la obra para realizar actividades no permitidas por la licencia.*
- *Se sustenta en los derechos del copyright.*
- *Pueden tener mayor difusión las obras.*
- *Puede licenciarse casi todo tipo de obra.*
- *Puede ayudar a combatir la piratería.*
- *Puede licenciar como autor único o como sociedad de autores.*
- *Facilitar el trabajo a otras personas al poder usar las obras para complementar su propio trabajo, siempre guardando la autoría del documento original.*
- *Se protege la propiedad intelectual, pero se tiene acceso a la cultura, a la información y para ser más amplio a la educación.*

***Desventajas del Creative Commons.-***

- *No queda muy claro cuáles son los beneficios para los creadores.*
- *No es respuesta alternativa al copyright.*
- *Siempre hay que asegurarse si el trabajo está bajo una licencia creative commons al ser usado.*
- *Creative commons no tiene control sobre los resultados.*
- *Las licencias no todos los países las tienen por lo que hay que adaptar la obra a la jurisdicción internacional.*
- *No asegura que en todos los países se van a respetar esos derechos, pues depende de la legislación al respecto en un determinado país y de la licencia seleccionada.*
- *Si el plagio de la obra se da en un país donde no hay leyes al respecto o las leyes de creative commons no están vigentes, no se puede hacer nada para evitarlo. (Aliprandi, 2012)*

*“Por otro lado la Licencia Pública General (inglés: General Public License o GPL) otorga al usuario la libertad de compartir el software licenciado bajo ella, así como realizar cambios en él. Es decir, el usuario tiene derecho a usar un programa licenciado bajo GPL, modificarlo y distribuir las versiones modificadas de éste.” (Muro, 2007, pág. 15)*

*“La licencia GPL adopta el principio de la no ocultación, respaldando el concepto moral que establece que todo software desarrollado con el uso de material licenciado bajo GPL debe estar disponible para ser compartido con el resto de la humanidad. GPL fue creada para mantener la libertad del software y evitar que alguien quisiera apropiarse de la*

*autoría intelectual de un determinado programa. La licencia advierte que el software debe ser gratuito y que el paquete final, también debe ser gratuito, asegurándose siempre de mantener los nombres y créditos de los autores originales. Como aspecto curioso, se debe considerar que, si se reutiliza un programa "A", licenciado bajo GPL, y se reutiliza un programa "B", bajo otro tipo de licencia libre, el programa final "C", debe de estar bajo la licencia GPL. Este concepto se introduce con el denominado copyleft a fin de garantizar que cualquier aprovechamiento de un programa bajo licencia GPL redunde sobre la comunidad.” (Muro, 2007, pág. 15)*

Entre las ventajas y Desventajas que exponen los tratadistas venezolanos: Mauricio García de Ceca y Wilnèl José Verdú Guerrero sobre la Licencia Pública General o GPL se puede mencionar:

### ***Ventajas de GPL***

- *Cualquier código fuente licenciado bajo GPL, debe estar disponible y accesible, para copias ilimitadas y a cualquier persona que lo solicite.*
- *De cara al usuario final, el software licenciado bajo GPL es totalmente gratuito, pudiendo pagar únicamente por gastos de copiado y distribución.*
- *Se ha establecido la idea global que GPL contribuye al mejoramiento y evolución del software, ya que la disponibilidad y acceso global de los programas permite la expansión del conocimiento depositado en cada pieza de software.*

### *Desventajas de GPL*

- *“Si el desarrollador incluye código fuente bajo GPL en otro programa, todo el programa final está obligado a seguir las condiciones y términos de la licencia GPL.*
- *El software licenciado bajo GPL carece de garantía. El autor del software no se hace responsable por el malfuncionamiento del mismo.*
- *De cara al desarrollador, no se puede establecer ningún cobro por las modificaciones realizadas. Únicamente se pueden establecer cobros asociados a copiado y distribución.*
- *Aunque GPL posibilita la modificación y redistribución del software, obliga a que se haga únicamente bajo esa misma licencia.” (García de Ceca & Verdú, 2012)*

En definitiva, estos dos procesos de licenciamiento de software están revolucionando la transferencia de tecnología en la actualidad por lo que dejarlos mencionados en esta parte del trabajo investigativo era necesario en virtud de contemplar nuevas aristas de posibles soluciones para este problemático sector que en definitiva busca desarrollar un espacio importante de progreso económico y social en esta sociedad contemporánea.

## CAPÍTULO III

### 3. La Transferencia de Tecnologías del software libre y software propietario

#### 3.1. Presupuestos Estatales sobre adquisición de software propietario

Actualmente los programas o software informático que utilizan las compañías y el sector público, en muchos de los casos, son licenciados. Esto quiere decir que se pagan importantes sumas de dinero por tener los permisos necesarios para ponerlos a operar en todos los equipos. Sin embargo, existen herramientas que pueden cumplir las mismas funciones de manera gratuita o a mucho menor costo.

*“El gran problema es que no existe un apartado específico para determinar cuánto se gasta realmente en la renovación y compra de licencias de software pero que, sin duda, es un monto considerable.”* (García, 2013).

Así lo asevera la periodista Daniela García en un artículo noticioso para el portal web *crhoy.com* realizado en el mes de julio de 2013, y en el cual se deja advertir la problemática en los gobiernos de la región latinoamericana respecto del gasto en software y en programas computacionales.

De acuerdo con el especialista en informática, Óscar Retana, el software libre da la posibilidad a los clientes de comprar únicamente las soluciones a las necesidades que tienen. Es decir, que si solamente necesita un programa que le ayude con la contabilidad, tendrá estas funciones únicamente; eso facilita la inversión de los recursos anteriormente utilizados en el pago de las licencias e incluso servidores de almacenamiento en nuevas necesidades de la entidad.

*“Independientemente de pensar de antemano cuál solución queremos tener, lo primero que se tiene que pensar es qué se necesita. El camino más usual es poner las necesidades en un cartel (de licitación) y veo quién me las pueden solucionar” (Retana, 2010)*

Por su parte, el Director del Centro de Informática de la Universidad de Costa Rica, Alonso Castro, comenta que de acuerdo con un estudio que realizó la universidad- un usuario de Office promedio tan sólo utiliza el 10% de las funciones que tiene disponibles, mientras que uno avanzado llega apenas al 20%. Por otro lado, los programas de software libre tienen la capacidad para cumplir con el 80% de las capacidades que tiene una licencia.

Este panorama daría a entender que los licenciamientos en software libre son los más indicados para desarrollarlos y que por lo tanto las empresas públicas y servicios estatales deberían migrar a la utilización de esta clase de tecnología, sin embargo es preciso señalar que no todo el panorama en el software libre es bueno, pues existen escenarios de alta conflictividad sobretodo en la relación que el software libre tiene con la propiedad intelectual y con el desarrollo de herramienta de ayuda para la utilización de estos programas.

Por lo general el software propietario en su licenciamiento trae consigo una garantía del producto la cual a su vez trae una ayuda técnica al desarrollo del programa por diferentes eventualidades que puedan ocurrir, como desperfectos en la elaboración del producto, esto no ocurre en el software libre que de ser gratuito pasa a ser desarrollado por varias personas que indistintamente pueden poner límites al desarrollo del licenciamiento creando con esto un caos y la falta de garantía en la

utilización de programas de software libre.

Esto quiere decir que no todo es bueno en este proceso de migrar del software propietario al software libre, es más los costos pueden representar también un obstáculo a la hora de desarrollar el software libre, dado que el programa constituye "algo nuevo", ello supone afrontar un costo de aprendizaje, de instalación, de migración, de interoperabilidad, cuya cuantía puede verse igual o mayor al pago de licenciamientos de software propietario.

Es por ello que se debe emprender en que, si bien los costos del software propietario son en la actualidad excesivos y muy onerosos para los estados y sus instituciones, también es cierto que migrar hacia un software libre de manera desorganizada puede salir más caro, de ahí que en primer lugar hay que tener en claro como dice el tratadista Martin Carranza Torres, que aunque desde la perspectiva de las empresas que desarrollan software propietario, *“se ensayan algunos argumentos sobre la bondad de ciertos productos sobre otros, discusión no lo bastantes relevante en lo sustancial. No se dirige la atención al fondo de la cuestión: los principios ético-filosóficos que sustentan a una y otra posición. Porque, a no dudarlo, esos principios están, soterrados, pero están.”* (Carranza M., 2004).

Esta posición que nace desde ese análisis financiero y se adentra en la interrogante de avizorarnos en el tiempo sobre cuál de los dos tipos de desarrollo son el mejor y más adecuado para nuestro país, hacen concluir que el debate debe ser manejado en un equilibrio total que facilite la toma de decisiones técnicas y liberarnos de coyunturas políticas o jurídicas, en palabras de Dr. Carranza:

*“Ocurre que los principios filosóficos, políticos, jurídicos y económicos del software libre y de su comunidad, pero fundamentalmente de los primeros, son erróneos, porque en nombre de una mal entendida libertad violan la justicia. Y si bien es cierto que una sociedad sin libertad es algo denostable, no lo es menos una sociedad sin justicia.”* (Carranza M. , 2004)

En definitiva es verdad que actualmente se gasta mucho en el pago de licenciamiento por parte de los estados y de corporaciones empresariales que requiere de la utilización de software para su diario trajinar, pero la solución de migrar al software libre también tiene que ser analizada porque si bien uno de los principios de este movimiento es la gratuidad en el servicio, esta misma gratuidad puede resultar cara a la hora de garantizar la aplicación y desarrollo efectivo de programas de computación en las instituciones estatales y empresas privadas.

Por ello es importante analizar ya en un ámbito nacional la situación actual de nuestra legislación con respecto al software propietario y si existe algún ejemplo de desarrollo de esta clase de programas en nuestro país.

### **3.2 Situación jurídica actual del software en el Ecuador**

En un principio en nuestro país se precisaba que la apertura para un nuevo esquema de derechos y el establecimiento de un estado constitucional de derechos y justicia era el ambiente clave para desarrollar una legislación que regule de manera efectiva el desarrollo de software y hardware en nuestro país así como de todas las tecnologías relacionadas con este tema, sin embargo, como lo vamos a advertir en

este subcapítulo en los últimos momentos se advierte una cierta pérdida de objetivos dentro de la legislación.

Pasemos entonces a analizar lo que dentro de la Constitución de la República establece referente al software y desarrollo tecnológico, dentro de la Carta Fundamental *“se garantiza la soberanía nacional, y se definen los sectores estratégicos entre los cuales están las tecnologías como hardware y software: Art 3. Son deberes primordiales del Estado: 2. Garantizar y defender la soberanía nacional.”* (Constitución de la República del Ecuador , 2008)

*“Además, se garantiza el acceso a las tecnologías, la capacitación, su desarrollo y la integración regional. Art 16. Todas las personas, en forma individual o colectiva, tienen derecho a: (...) 2. El acceso universal a las tecnologías de información y comunicación.”* (Constitución de la República del Ecuador , 2008)

*“Art. 234. El Estado garantizará la formación y capacitación continua de las servidoras y servidores públicos a través de las escuelas, institutos, academias y programas de formación o capacitación del sector público; y la coordinación con instituciones nacionales e internacionales que operen bajo acuerdos con el Estado.*

*Art. 322. Se reconoce la propiedad intelectual de acuerdo con las condiciones que señale la ley. Se prohíbe toda forma de apropiación de conocimientos colectivos, en el ámbito de las ciencias, tecnologías y saberes ancestrales. Se prohíbe también la apropiación sobre los recursos genéticos que contienen la diversidad biológica y la agro-biodiversidad.*

*Art. 334. El Estado promoverá el acceso equitativo a los factores de*

*producción, para lo cual le corresponderá: 1. Evitar la concentración o acaparamiento de factores y recursos productivos, promover su redistribución y eliminar privilegios o desigualdades en el acceso a ellos. (...) 3. Impulsar y apoyar el desarrollo y la difusión de conocimientos y tecnologías orientados a los procesos de producción.” (Constitución de la República del Ecuador , 2008)*

También dentro de la Carta fundamental se deja expresado la responsabilidad que la administración estatal tiene para con el desarrollo tecnológico, en base a ello se puede determinar que la legislación ecuatoriana promueve y tiene una visión abierta hacia la utilización del software libre como un mecanismo eficaz de transferencia de tecnología.

*“Art. 347. Será responsabilidad del Estado: (...) 8. Incorporar las tecnologías de la información y comunicación en el proceso educativo y propiciar el enlace de la enseñanza con las actividades productivas o sociales.*

*Art. 385. El sistema nacional de ciencia, tecnología, innovación y saberes ancestrales, en el marco del respeto al ambiente, la naturaleza, la vida, las culturas y la soberanía, tendrá como finalidad: 1. Generar, adaptar y difundir conocimientos científicos y tecnológicos. (...) 3. Desarrollar tecnologías e innovaciones que impulsen la producción nacional, eleven la eficiencia y productividad, mejoren la calidad de vida y contribuyan a la realización del buen vivir.*

*Art. 386. El sistema comprenderá programas, políticas, recursos, acciones, e incorporará a instituciones del Estado, universidades y escuelas*

*politécnicas, institutos de investigación públicos y particulares, empresas públicas y privadas, organismos no gubernamentales y personas naturales o jurídicas, en tanto realizan actividades de investigación, desarrollo tecnológico, innovación y aquellas ligadas a los saberes ancestrales.*

*Art. 387. Será responsabilidad del Estado: 1. Facilitar e impulsar la incorporación a la sociedad del conocimiento para alcanzar los objetivos del régimen de desarrollo.*

*Art. 423. La integración, en especial con los países de Latinoamérica y el Caribe será un objetivo estratégico del Estado. En todas las instancias y procesos de integración, el Estado ecuatoriano se comprometerá a: (...) 2. Promover estrategias conjuntas de manejo sustentable del patrimonio natural, en especial la regulación de la actividad extractiva; la cooperación y complementación energética sustentable; la conservación de la biodiversidad, los ecosistemas y el agua; la investigación, el desarrollo científico y el intercambio de conocimiento y tecnología; y la implementación de estrategias coordinadas de soberanía alimentaria.”*  
(Constitución de la República del Ecuador , 2008)

En conclusión, la Constitución de la República del Ecuador deja sentadas las bases en el establecimiento de principios que procuran el desarrollo tecnológico de la nación y con esto el inicio de un proceso de independencia y soberanía en este ámbito, quedando advertida que la postura del estado y del gobierno actual es dar facilidades al establecimiento de software libre dentro de la transferencia de tecnología en nuestro país.

Esta postura es más visible si exponemos que el Gobierno de la República de

Ecuador promueve y viene ejecutando una Estrategia de migración hacia el software libre, pues mediante “*Decreto Ejecutivo No. 1014 emitido el 10 de abril de 2008, se dispone el uso de Software Libre en los sistemas y equipamientos informáticos de la Administración Pública de Ecuador. Es interés del Gobierno ecuatoriano alcanzar soberanía y autonomía tecnológica, así como un ahorro de recursos públicos.*”

*La Subsecretaría de Gobierno Electrónico es responsable de elaborar y ejecutar planes, políticas y reglamentos para el uso de Software Libre en el Gobierno Central. Como órgano regulador desarrolla la “Estrategia para la implantación de Software Libre para la Administración Pública Central de Ecuador”. La Estrategia define políticas, objetivos y planes de acción en base a cuatro ejes estratégicos:*

- 1. Promulgación de estándares y normatividad,*
- 2. Formación de masa crítica,*
- 3. Planificación, seguimiento y control y*
- 4. Difusión del Software Libre” (Estrategia para la implantación de Software Libre para la Administración Pública Central de Ecuador, 2009)*

Se definen como políticas a la utilización de estándares abiertos, la minimización de compra de licencias propietarias, la contratación de servicios en proyectos informáticos, la reutilización del software y el uso preferencial de programas navegadores como medios de acceso.

Previo a la promulgación del Decreto de Software Libre en abril del 2008, la mayoría de instituciones de la Administración Central utilizaban software privativo en sus sistemas informáticos. Actualmente, todas estas entidades tienen planificado

o se encuentran ejecutando procesos de migración y prácticamente todos los nuevos proyectos informáticos consideran la adopción de herramientas de Software Libre.

Sistemas transversales del Estado ecuatoriano se han desarrollado totalmente con Software Libre, el Sistema Nacional de Compras Públicas, el Sistema Nacional de Recursos Humanos y el Sistema de Gestión Documental. Estos sistemas son un referente de soberanía y autonomía tecnológica, así como de ahorro de recursos públicos, áreas que son de interés del Gobierno.

Sin embargo, de haber invertido en la elaboración de esta estrategia y de la promulgación del decreto ejecutivo se advirtieron algunos puntos contradictorios y anomalías en su aplicación, por lo que se llegó a la conclusión que era necesario emprender en una legislación más integral y mejor elaborada.

Actualmente nuestro país cuenta con esa legislación integral en el Código Orgánico de la Economía Social del Conocimiento o también conocido como Ley COESC (Código Ingenios). *“Esta ley elaborada desde una wiki por un equipo multidisciplinario se ha venido trabajando desde el 2013, y ha sido aprobada en junio del 2015, la cual consta de cuatro libros y 570 artículos.”* (Asociación software libre del Ecuador , 2015)

El libro III (de la Gestión de los Conocimientos), en el apartado segundo (del Software Libre y Formatos Abiertos), contiene tres artículos que intentan complementar el Decreto 1014. El artículo 135, que define el Software Libre desde sus cuatro libertades, el artículo 136 que menciona la obligatoriedad del uso de SL, y el artículo 137 que habla de la libre elección de software.

La obligatoriedad recae en el sector público; y las instituciones del Sistema Nacional de Educación y del Sistema de Educación Superior. En el caso de excepción, esta deberá ser sometida a autorización por medio de solicitud a la entidad rectora del Sistema Nacional de Contratación Pública.

Sin embargo, esta solicitud de autorización queda excluida para casos en que se deba actualizar un software privativo previamente adquirido, o para casos en que existan razones técnicas o comerciales para no proceder a la migración. Es decir, las excepciones permitidas en esta ley no son muy claras, en virtud que el Código Ingenios siempre establece la necesidad de crear un Reglamento, tanto para motivar la autorización de adquisición de otro tipo de software, cuanto para interponer excepciones para el no traslado hacia un software libre.

Por otra parte el último artículo que hace mención de la libre elección de software, indica que los usuarios tienen derecho a la libre elección del software en dispositivo que admitan más de un sistema operativo, dándole la posibilidad de elegir entre un equipo con software o sin él, o con software libre o privativo, y los proveedores estarán obligados a ofrecer estas alternativas.

En pocas palabras, la ley ecuatoriana permite exclusivamente software privativo, cuando no exista alternativa de software libre. Ninguna de las leyes citadas, ni el Decreto 1014 ni el Código Ingenios, consideran el software libre como un proceso transversal, que involucra Sistema Educativo, Sociedad Civil y Estado en construir un ecosistema sostenible en el tiempo.

Todo este marco regulatorio es el que actualmente se encuentra regulando la

utilización del software libre y ha zanjado la discusión de cuál de los dos softwares es mejor, promoviendo la balanza hacia el lado del software libre.

### **3.2.1. Análisis del decreto 1014 de 2008 sobre Software Libre**

El día jueves 10 de abril del 2008 se emitió el decreto 1014 por parte del Ejecutivo que promueve el uso de software libre en las instituciones públicas del Ecuador, este decreto ejecutivo se encuentra basado en los siguientes ejes centrales:

*1. Cumplimiento de recomendaciones Internacionales: La Carta Iberoamericana de Gobierno Electrónico aprobada por la “IX Conferencia Iberoamericana de Ministros de Administración Pública y Reforma del Estado”, que recomienda el uso de estándares abiertos y software libre como herramientas informáticas.*

*2. Con los objetivos fundamentales de: Alcanzar la soberanía y autonomía tecnológica. Alcanzar un ahorro significativo de recursos públicos.*

(Asociación software libre del Ecuador , 2015)

Se decretó establecer como política pública la utilización de software libre en los sistemas y equipamientos informáticos de las Entidades de la Administración Pública Central, tomando como definición de Software Libre las cuatro libertades promulgadas por Richard Stallman.

Este decreto además indicaba que se debía evaluar periódicamente los sistemas informáticos que utilizan software propietario con el fin de migrarlos a software libre. Lo cual desde esta perspectiva se advierte que fue tibia manera de fomentar la industria nacional, sugiriendo dar prioridad en el proceso de selección

al software producido en el País, entre una lista de alternativas que va desde lo nacional totalmente a lo internacional totalmente. Lo cual no hace sentido porque hasta ahora no existe un mercado potencial o una industria creciente en materia de software libre, por lo tanto, es un intento vano de querer fomentar la industria local de Software Libre.

Este escenario se suma a las excepciones y previsiones que dentro del decreto se oponen, las cuales son contradictorias al Decreto en sí porque:

*“Previo a la instalación, primero las Entidades deben verificar la existencia de capacidad técnica que brinde el soporte necesario para el uso de Software Libre. En la realidad ecuatoriana para muchos productos de Software Libre no hay suficiente capacidad técnica ni en las Entidades, ni en las empresas privadas. Entonces, esto se vuelve un argumento muy fuerte para impedir que la mayoría de productos puedan implementarse.*

*Faculta la utilización de software propietario, cuando no exista un producto en SL que lo reemplace, o si por una evaluación fría de “costo-beneficio” no resulte conveniente migrar a Software Libre o dar continuidad a un desarrollo de Software Libre, en razón de que el software privativo está funcionando satisfactoriamente.” (Asociación software libre del Ecuador , 2015)*

De ahí que en el análisis que realiza la Asociación de Software libre considera que el Decreto es un tanto flojo, con objetivos pobres. Parece que hubiera sido emitido para salir al paso, sin un estudio debido de la realidad del país. El Decreto no da una obligatoriedad, y tienen excepciones que, por la realidad del País, vendrían a ser más radicales que el mismo propósito del Decreto.

También no se está considerando fomentar la industria local, ni promover un mercado que pueda satisfacer los requerimientos y demandas de software libre. El Decreto por lo tanto es obsoleto, no se integra con la matriz productiva del País, ni está alineado a los objetivos de Educación, Investigación y Desarrollo, y los Planes de Gobierno emitidos, sean estos:

1. *Plan del Buen Vivir,*
2. *Plan de Seguridad Integral*
3. *Plan de Gobierno Electrónico.* (Asociación software libre del Ecuador , 2015)

A parte del Decreto 1014, existen recursos en el marco jurídico del País que hacen referencia a la soberanía tecnológica, como los antes mencionados y por ello es importante pasar a analizar, sobre todo los planes de desarrollo que interactúan en su misión de aplicar lo establecido en las normas de una manera más eficaz.

### **3.3. La transferencia de tecnologías en Ecuador según el Plan Nacional del Buen Vivir 2013-2017**

Como parte de la Planificación Nacional Estatal se ha elaborado lo que se denomina como el Plan Nacional del Buen Vivir, dentro de este documento de programación para el gobierno existe un tema que contempla a las: “Tecnologías, innovación y conocimiento” como parte de sus estrategias, y además plantea 12 objetivos con sus políticas que sirven de guía.

Si se revisa el Plan Nacional del Buen vivir se visualiza que dentro del Objetivo 10, se establece que el estado en el periodo de tiempo de 2013 a 2017 debe: “Impulsar la transformación de la matriz productiva” y su política 10.3 reviste

especial importancia que puede aplicarse en la generación de una industria local de software libre, con ayuda de inversión pública. (Plan Nacional del Buen Vivir 2013-2017, 2014)

*“Objetivo 10: Impulsar la transformación de la matriz productiva. Una producción basada en la economía del conocimiento, para la promoción de la transformación de las estructuras de producción. (..) La transformación de la matriz productiva supone una interacción con la frontera científico-técnica, en la que se producen cambios estructurales que direccionan las formas tradicionales del proceso y la estructura productiva actual, hacia nuevas formas de producir que promueven la diversificación productiva en nuevos sectores, con mayor intensidad en conocimientos, bajo consideraciones de asimetrías tecnológicas entre países.”* (Asociación software libre del Ecuador , 2015)

Si se sigue revisando el mismo documento sobre la planificación estatal se encuentra que dentro del listado de políticas que contiene el objetivo 10 se tiene que en la política 10.3 ordena diversificar y generar mayor valor agregado en los sectores prioritarios que proveen servicios. (Plan Nacional del Buen Vivir 2013-2017, 2014)

También en la política 10.7 se establece que el estado debe impulsar la inversión pública y la compra pública como elementos estratégicos del Estado en la transformación de la matriz productiva. (Plan Nacional del Buen Vivir 2013-2017, 2014)

Pero si el objetivo 10 del Plan nacional del Buen vivir expresa la necesidad de impulsar la transformación de la matriz productiva y esto manda a que el Estado

ecuatoriano impulse sectores como la investigación tecnológica en software libre o propietario, dentro del objetivo 11 del mismo programa gubernamental se advierte un objetivo mucho más comprometido con el impulso de software en nuestro país.

El objetivo 11 expresa que el estado debe asegurar la soberanía y eficiencia de los sectores estratégicos para la transformación industrial y tecnológica, esto quiere decir que dentro de esta programación se pretende ejercer soberanamente la gestión económica, industrial y científica, de los sectores estratégicos.

Esto permitirá generar riqueza y elevar en forma general el nivel de vida de nuestra población. Para el Gobierno convertir la gestión de los sectores estratégicos en la punta de lanza de la transformación tecnológica e industrial del país, constituye un elemento central de ruptura con el pasado. Fortalecer a las empresas públicas como agentes en la transformación productiva. Soberanía estratégica en la gestión de los recursos naturales. Y además impulsar una economía basada en el conocimiento, la ciencia y la tecnología. Apropiación científica y tecnológica de nuestros recursos.

Desde este punto de vista tanto el objetivo 10 como 11 del Plan nacional del Buen Vivir resaltan en la necesidad de migrar hacia nuevas tecnologías y consolidar a nuestro país como un generador y proveedor de conocimiento tecnológico a futuro, lo cual es muy admirable pero que debe ser aplicado de manera eficaz, sino solo se estaría expresando una carta retórica de antiguos planes gubernamentales.

Por otro lado, también tiene relación con la transferencia de tecnología de software el Plan Nacional de Gobierno electrónico o conocido como PNGE por sus

siglas, esta planificación mucho más específica en temas electrónicos como el software es un documento fundamentado en la Carta Iberoamericana de Gobierno electrónico del año 2007, formula 12 principios que precautelan el derecho de los ciudadanos a relacionarse con el Estado electrónicamente.

Entre uno de ellos está el principio 7 de “Adecuación tecnológica” que recomienda el uso de estándares abiertos y de software libre en razón de la seguridad, sostenibilidad a largo plazo y la socialización del conocimiento.

*“Principio de adecuación tecnológica: Garantiza que las administraciones elegirán las tecnologías más adecuadas para satisfacer sus necesidades, por lo que se recomienda el uso de estándares abiertos y de software libre en razón de la seguridad, sostenibilidad a largo plazo y la socialización del conocimiento”.* (Plan Nacional de Gobierno Electrónico , 2014)

Esta planificación más específica tiene relación con el marco normativo del Decreto 1014, con el Plan Nacional de Buen Vivir y además con el Plan Nacional de Seguridad Integral. En este documento, en el capítulo 4 que trata sobre “La Seguridad Integral desde el Buen Vivir”, expone que la soberanía y seguridad integral involucra de sobre manera a la Soberanía Tecnológica y Ciencia, por lo tanto, constituye una de las garantías que debe proporcionar el Estado, que de hecho está en la Constitución de la República de 2008.

Se menciona además que en el Buen Vivir en la Seguridad Integral también se contempla entre las amenazas, la inseguridad cibernética.

*“La integralidad del concepto de seguridad también se ve reflejada en los cinco*

*ámbitos que se interconectan entre sí y enmarcan su accionar: Defensa y Relaciones Internacionales; Seguridad Ciudadana y Justicia; Gestión de Riesgos y Ambiente; Soberanía Tecnológica y Ciencia e; Inteligencia Estratégica para el fortalecimiento democrático.” (Plan Nacional de Seguridad Integral, 2011)*

En el mismo Plan Nacional de Seguridad dentro del capítulo 4.3 se realiza una exposición del “Ámbito de la Seguridad Integral”, e indica lo siguiente:

***“Defensas y Relaciones Internacionales:** La Soberanía no se limita al ejercicio del poder de decisión sobre un territorio determinado, como se ha concebido tradicionalmente, sino que se extiende a todos los campos en los que se desarrolla la vida, para cumplir el rol de protección de los derechos, libertades y garantías de los ciudadanos y ciudadanas. De allí que se reconozca la necesidad de la defensa del ejercicio de las soberanías, cuya coexistencia se produce de manera articulada e interdependiente. Garantizar la soberanía implica, en este sentido, la defensa del Estado y de sus recursos ecológicos, alimentarios, energéticos, económicos, tecnológicos y del conocimiento.*

***Movilidad Nacional:** En el Buen Vivir en la Seguridad Integral se orienta al ejercicio de los derechos del ser humano y de la naturaleza, con todas las actividades que él implica, en un marco de seguridad ante las amenazas existentes (...) y que entre otras son: el crimen organizado en sus diferentes manifestaciones, (...) terrorismo, inseguridad cibernética, delincuencia común, (...), entre otras”.* (Plan Nacional de Seguridad Integral, 2011)

Además, la Soberanía Tecnológica toma particular relevancia, indicando que

es importante contar con las capacidades soberanas en materia de investigación, e indica que la ciencia y la tecnología debe responder a los objetivos nacionales, y además que en el marco del cambio de la matriz productiva la industria de la defensa ha tenido prioridad nacional. De la misma manera el Software Libre debe constituir prioridad nacional y la matriz productiva debe aportar a ello, ya que no se puede concebir una apropiada Defensa con tecnologías 100% propietarias.

*Soberanía Tecnológica y Ciencia Aplicada a la Seguridad: Por otro lado, para garantizar la seguridad del Estado y sus habitantes es importante contar con las capacidades soberanas en materia de investigación, preservando nuestros intereses nacionales y protegiendo los esfuerzos y desarrollos en este campo. Es el caso de la industria de la Defensa, la cual se ha constituido en una prioridad nacional, en el marco del cambio de la matriz productiva. En este sentido, la ciencia y la tecnología deben responder a los objetivos nacionales. La investigación necesaria debe incorporar un enfoque científico y tecnológico multidisciplinar (ciencias físicas y químicas, ciencias de la vida, ingeniería, informática, telecomunicaciones), tomando en cuenta que el fortalecimiento del factor humano es de particular importancia para el campo de seguridad". (Asociación software libre del Ecuador , 2015)*

En definitiva esta planificación es sistemática y el estado ha propiciado la estructuración de una mejor legislación mucho más integral que procure el definitivo despegue de la implementación de software libre en las instituciones estatales, lo cual hace concluir que la postura gubernamental busca estructurar de la mejor forma un marco regulatorio general e integral que coadyuve al desarrollo de nuevas tecnologías el cambio de matriz productiva y la consecuente transferencia de este

tipo de tecnologías.

### **3.4. Transferencia efectiva de tecnología de software.**

Para advertir de mejor forma en este trabajo investigativo la transferencia efectiva de tecnología software en nuestro país o la que más convenga a nuestros intereses nacionales es preciso entrar previamente al análisis de casos concretos, por lo tanto se va a exponer dos casos de transferencia y utilización de software libre que están generando efectos satisfactorios, el uno es a nivel regional, específicamente en el país de Costa Rica y el otro ejemplo es a nivel nacional, específicamente el caso de la utilización de software libre por parte de la Universidad de Bolívar.

Entrando a analizar el caso de Costa Rica, este país centro americano camina a paso lento de la mano del software libre un grupo de ingenieros han encontrado un nicho de negocio al modificar y personalizar el código abierto a la medida de las necesidades de sus clientes, y al brindar el soporte para esas soluciones.

Gridshield, GreenCore y Alkaid son tres de las al menos 30 empresas costarricenses que incursionan en la tendencia mundial de implementar el software libre como un servicio, según datos de la Cámara de Tecnologías de la Información (Camtic). La seguridad, las redes sociales, la nube, los móviles y el manejo de contenidos son varios de los grandes nichos en los que incursionan los proveedores de SL en el mundo. (Empresarios , 2014)

Contrario a la creencia popular, para estas empresas el licenciamiento libre no es gratuito ni mata el negocio del software libre, sino todo lo contrario, es una nueva oportunidad para monetizar sus conocimientos. *“En Costa Rica hay un*

*montón de empresas que estamos facturando cientos o miles de dólares por brindar soporte”.* (Retana, 2010)

Para Jonathan Vargas, fundador de Alkaid otra empresa de software libre en Costa Rica, esa fue precisamente una de las razones por las cuales su compañía comenzó a vender servicios de este tipo de programas, hace ocho años. *“Pocas empresas en ese momento estaban entregando software bajo este modelo y había una tendencia e interés en su demanda”.* (Empresarios , 2014)

El software libre no suele desarrollarse desde cero porque la madurez de un programa libre es su carta de presentación, sino que se dedican a modificarlo, personalizarlo y brindar el soporte. Para la investigación realizada por la revista empresario de Costa Rica existen mitos con respecto a la utilización y desarrollo del software libre, primero muchas de las maravillas y de los oscuros peligros que se le atribuyen al licenciamiento libre son relativos.

El ahorro, la calidad, la seguridad y el respaldo son algunos de los fantasmas que rondan la casa del software libre, en contra y a favor de él. *“Estos aspectos son responsabilidad propiamente del software en sí y sus desarrolladores, no del modelo de licenciamiento”.* (Empresarios , 2014)

El último estudio de la consultora internacional en código abierto Black Duck Software indica que un 61% de las corporaciones consultadas adopta esta modalidad porque lo considera más barato. Una de las ventajas más populares es el ahorro asegurado para las empresas que lo utilicen, pero eso también puede ser un mito. Aunque el código suele ser gratuito, pues ya es abierto y toda la comunidad puede

acceder a él, la implementación y la personalización del producto sí tiene un precio que puede ser tan caro como una licencia privada y tradicional. (Empresarios , 2014)

Si la firma o institución es muy grande (como una universidad) necesita personal capacitado para atender los errores que surjan, y eso también puede ser costoso. Por eso, los ingenieros que se dedican a implementarlo son cuidadosos al hablar de las posibilidades de ahorro. *“Por más maravilloso que sea, no podemos solo enfocarnos en si tiene o no un menor costo “En lugar de invertir varios miles de dólares en licenciamientos, se puede invertir únicamente en implementaciones de excelente calidad”*. (Empresarios , 2014)

La carencia o abundancia de seguridad y calidad en el software de licencia libre también podría ser un mito. Es probable que sobre un navegador de licencia libre como Mozilla Firefox naden miles de ojos que controlan sus fallas y corrigen sus errores casi al mismo tiempo en que se producen. También está claro que otro con menos edad puede contener errores estructurales que lo vuelvan vulnerable a los ataques.

Lo cierto es que esa vulnerabilidad dependerá de la madurez del desarrollo de los programas, como en cualquier otro caso, y del respaldo con el que cuenten. “El código abierto ha demostrado su calidad y seguridad”, luego de presentar los resultados de la encuesta de este año. Ese respaldo puede venir desde varias fuentes. Una de ellas es Internet y los foros que se generan alrededor de cientos de temas que tienen que ver con el desarrollo e implementación de SL, pues es un modelo colaborativo. (Empresarios , 2014)

En definitiva, lo que se está proyectando en Costa Rica es un nuevo modelo de licenciamientos para software libre como los tipos de licencia denominados Creative Commons o GLP, ambos son tipos de licencia que permiten el desarrollo tecnológico su transferencia adecuada y garantizan un servicio que, aunque no es óptimo permite el cumplimiento de obligaciones y deberes que tienen las empresas privadas y públicas en el país centroamericano.

Otro ejemplo que se puede advertir como un modelo de transferencia de tecnología software de manera eficaz se está dando en nuestro país. La Universidad Estatal de Bolívar es la primera universidad ecuatoriana en migrar al Software Libre en sus estaciones de trabajo.

El Instituto Ecuatoriano de la Propiedad Intelectual fue invitado por la Universidad Estatal de Bolívar para participar en el “Séptimo Encuentro Internacional de TICs y Software Libre” el pasado 12, 13 y 14 de octubre en Guaranda, con la presencia de 300 personas inscritas y de reconocidos expositores internacionales de Venezuela, Brasil y Bolivia.

Durante el encuentro de software libre más importantes del país, Rafael Bonifaz, Asesor de Tecnología, quien en representación del IEPI tuvo una destacada participación con una conferencia sobre “El uso del software Libre en el Estado y en la Educación” donde resaltó la importante gestión de la Universidad Estatal de Bolívar al ser primera universidad ecuatoriana en migrar a software libre. *“Esta experiencia será fundamental para ser replicada a nivel nacional luego de la aprobación del Código Ingenios”*. (Bonifaz, 2011)

Por su parte, Diego Saavedra, autor de uno de los proyectos ganadores del ‘Banco de Ideas’ que promueve la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (Senescyt), expuso su proyecto ‘Aprendizaje Libre’ donde enseña a cerca de 400 niños de entre 8 y 12 años a desarrollar video juegos con Software Libre en la ciudad de Loja.

Además de estas conferencias, el “Séptimo Encuentro Internacional de TICs y Software Libre” fue un escenario favorable para abordar el uso del software libre como herramienta que permita potenciar la industria turística, la seguridad informática, la privacidad en Internet; así como, el proyecto Cedía y sus contribuciones al software libre y otros casos de éxito.

Estos dos ejemplos tanto el de Costa Rica como el de la Universidad de Guaranda permiten discernir sobre la forma eficaz en la que debe transferirse la tecnología software en nuestro país es así que en lo principal se puede afirmar que el modelo de licenciamiento (libre o privativo) no debe ser la única razón para escoger un programa sobre otro.

Lo importante es detectar las necesidades que tiene una empresa y la sociedad en general, ese es el primer paso para tomar una decisión. La sugerencia de los especialistas es que tome en cuenta todas las posibilidades para buscar soluciones. Dentro de ellas, el software libre. Si encuentra una solución en este modelo, busque que tenga un respaldo adecuado: que una firma o un grupo de ingenieros lo conozcan y lo implemente es un buen indicador, al igual que la madurez del programa.

Lo importante que advertirá la transferencia de tecnología y si el

emprendimiento del gobierno de migrar a software libre es lo indicado, es manifestar que por primera vez nuestro país se encuentra inmerso y preocupado en un debate contemporáneo que tiene gran relevancia en la estructura organizativa de las empresas a nivel planetario.

No solo se trata de diferenciar cuál de los dos tipos de licenciamiento es el más importante y el más adecuado para nuestro país, se trata de advertir la importancia que tiene el desarrollo de estos programas en la vida cotidiana de nuestra sociedad, razón por la cual el desarrollo tecnológico debe guarda una sinergia con la parte humana, su desarrollo dependerá no solo del análisis jurídico que podamos darle a este asunto, sino que interactuaran otros aspectos humanos en la toma de decisiones que transformen este sector y por lo tanto la sociedad en su conjunto.

De ahí que en el presente trabajo de titulación es pertinente dejar anotado que tanto el sistema software libre como el propietario son eficaces a la hora de promover un desarrollo tecnológico en los tiempos actuales, sin embargo en vista del análisis realizado se puede distinguir que el software libre tiene características propias que permiten a países como el nuestro con poca cultura respeto de los derechos de propiedad intelectual y con una necesidad básica de desarrollo de estos mecanismos, adentrarse de mejor forma en las relaciones de transferencia y ser de mejor aplicación.

Esto quiere decir que desde el ámbito de análisis de este trabajo el software libre tal y como está caracterizado en los últimos momentos es el mejor para ser empleado en las relaciones de transferencia electrónica e informática en nuestro país,

por cuanto existe una transferencia del código fuente que es en donde radica gran parte del conocimiento de este modo es recomendable su regulación de una forma más sistemática a pesar de que dentro del Código Ingenios ya se establece principios jurídicos básicos para su legislación.

Si se analiza el Capítulo III sobre de los derechos de autor, la Sección V sobre Disposiciones especiales sobre ciertas obras el primer párrafo sobre software y bases de datos del Código orgánico de la economía social de los conocimientos, creatividad innovación, se advierte que el legislativo si tuvo las reservas necesarias para comenzar a establecer normas básicas sobre las obras tecnológicas software, regula tanto el software privativo y bases de datos, cuanto el software libre y formatos abiertos.

Sin embargo el aporte que el trabajo investigativo puede mostrar en esta clase de regulaciones está encaminada a la reglamentación más subjetiva que permita que los principios establecidos en este Código Ingenios se efectuó de mejor forma, por ejemplo al advertir las importancias en el tipo de licenciamiento de software libre denominado como GLP y también el Creative Commons se puede instituir dentro del reglamento para el licenciamiento libre un título que regule estas dos clase de instrumentos jurídicos para el desarrollo del software libre en Ecuador.

Básicamente el aporte de este trabajo en el ámbito jurídico se enmarcaría en la elaboración del reglamento para la aplicación de un software libre, en el cual iría insertados artículos que propendan a la aplicación de licenciamientos en las instituciones del estado enmarcadas dentro de las licencias GLP o de Creative Commons, estas disposiciones tendrían su base en el Art. 136 del Código de ingenios

que en su parte pertinente dice:

*Artículo 136.- Obligatoriedad de uso de software libre y estándares abiertos. - El sector público y las instituciones del Sistema Nacional de Educación y del Sistema de Educación Superior en todos sus niveles de formación, deberán usar obligatoria y exclusivamente software libre y estándares abiertos. En el caso de que no sea pertinente el uso de dicho software libre o estándares abiertos, las entidades públicas obligadas en este artículo, deberán solicitar motivadamente la autorización de adquisición de otro tipo de software a la entidad que se establezca mediante Reglamento. (...) (Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación , 2016)*

Este artículo claramente afirma que de no existir programas software libre que puedan insertarse en las labores de una determinada empresa del sector público estas empresas deberán buscar el software más acorde, pero tomando en cuenta lo que el reglamento regule a este parecer, de ahí que parece oportuno dirigir este tipo de regulación hacia licenciamientos como el GLP y el Creative Commons, esta enmarcación permitirá un manejo más amplio al momento de migrar a un software libre.

## CONCLUSIONES

1. Se concluye que el software nació como un tipo de desarrollo tecnológico que venía supeditado al mercado de hardware, es decir en un primer momento lo que se vendía era un aparato electrónico que permitía el desarrollo de programas computacionales necesarios para el desenvolvimiento de una empresa pública o privada.
2. El desarrollo y evolución tecnología desde 1960 aproximadamente, trastoco esta visión de mercado haciendo necesario separar la parte corporal de un computador de sus sistema y programas, esta nueva visión de desarrollar software por separado del hardware permitió la accesibilidad de todas las personas a comprar un computador.
3. El desarrollo del software en primer momento obedeció a fines académicos, su desarrollo y evolución están muy relacionadas con las investigaciones en Universidades y en campos tecnológicos que de apoco impusieron un tipo de licencia corporativa para proteger sus invenciones, sin embargo, estas protecciones sirvieron de base para desarrollar un mercado multimillonario y con esto aparecieron las grandes corporaciones de tecnología software como por ejemplo Microsoft.
4. Con el aparecimiento del grupo que promovía la utilización y desarrollo de un software libre nace el debate jurídico sobre si debe darse facilidad a este grupo de desarrollar programas software libre irrespetando derechos de propiedad intelectual o debe prohibirse su establecimiento.
5. A nivel regional y nacional, se puede concluir que muy pocas Instituciones

Públicas están utilizando el Software Libre, a pesar de que existe un andamiaje de políticas claras que respaldan la implementación del Software Libre; como muestra de esto se puede destacar el Decreto 1014, el Código Ingenios entre otros.

6. Por otro lado, es importante destacar el Código Ingenios; ley presentada en junio del 2015 misma que incluye la obligatoriedad de migrar del Software Propietario al Software Libre, pero en continuidad de su texto deja expuesta la posibilidad de adquirir una Licencia de Software Propietario, debilitando y contradiciendo así la obligatoriedad de la implementación del Software Libre. Se debe tomar el ejemplo de las leyes Latinoamericanas de países como Venezuela, Uruguay, Brasil, Bolivia en donde la norma es rigurosa en cuanto a políticas de Software Libre dentro del Estado y su aplicación.

## RECOMENDACIONES

1. Es recomendable que el debate jurídico sobre la utilización de software libre o propietario se centre en la pérdida o transformación de derechos e instituciones jurídicas que han servido de fundamento para el desarrollo de la humanidad, me refiero a que lo más importantes es advertir si se están vulnerando derechos de propiedad intelectual o por el contrario nos encontramos ante una transformación de estos derechos.
2. El debate no debe dirigirse solamente al ámbito económico, para determinar cuál de los dos tipos de software es el más apropiado para la transferencia de tecnología en nuestro país, se debe emprender en un debate filosófico jurídico, con el fin de distinguir y ponderar principios básicos de convivencia dentro de la sociedad, de nada nos servirá ahorrar dinero si perdemos prestigio en la merma de valores sociales o culturales más valiosos dentro de la organización social.
3. La migración emprendida por el actual gobierno de ir de un software propietario a un software libre debe realizarse en un marco de cautela pero de toma de decisiones firmes, tanto el decreto 1014 como la actual ley de segundo orden que regula este sector, tienen un cierto grado de cautela lo que limita el establecimiento de empresas dedicadas al desarrollo de software libre, en Costa Rica algo importante que ha podido sujetar este propósito de utilizar software libre, es el establecimiento de más de 30 empresas dedicadas a dar soporte y desarrollar software en ese país, eso debería ser emulado en Ecuador si queremos mantener y desarrollar lo propuesto en normas legales

y planes del Estado.

4. Desde el ámbito jurídico tanto el software propietario como el libre son dos formas de comportamiento del mercado y del desarrollo electrónico que deben ser regulados tomando en cuenta los principios legales y constitucionales, es recomendable que el legislativo y los poderes del Estado emprendan en un debate serio sobre la conveniencia de migrar a software libre, porque sin un respaldo tecnológico verdadero el propósito de abaratar costos por la compra de software propietarios puede salir mal y en definitiva salir más caro la migración hacia el software libre.

## BIBLIOGRAFÍA

1. Aliprandi, S. (2012). *Creative Commons: Guía De Usuario*. Roma Italia: Ledizione .
2. Amaya, J. (2010). *Sistemas de información gerenciales: Hardware, software, redes, Internet, diseño*. Bogota : Eco Ediciones .
3. Asociación software libre del Ecuador . (2015). *Marco Legal del Software Libre en Ecuador*. Quito .
4. Bonifaz, R. (2011). *El uso del software Libre en el Estado y en la Educación* . Guaranda Ecuador: IEPI.
5. Brocca, & Casamiquela, 2. (2005). *Las licencias de software desde la perspectiva del usuario final*. Universidad Nacional del Comahue.
6. Brocca, J., & Casamiquela, R. (2006). *Las licencias de software desde la perspectiva del usuario final*. Neuquén Argentina : Universidad Nacional del Comahue.
7. Carranza, M. (2004). *Problemática Jurídica del Software Libre* . Buenos Aires Argentina : Lexis Nexis .
8. Carranza, M. (2006). *Problemática Jurídica del software libre*. Buenos Aires Argentina: LexisNexis.
9. Castro, A. (2010). *Ing Sistemas Computacionales* . Obtenido de <https://sites.google.com/site/jachsistemascomputacionales/classroom-news>
10. Ceballos, J., & Ramírez, E. (2015). *El Software Libre como una alternativa tecnológica de soporte: en el Centro Universitario del Norte (CUNorte)*. Monterrey México: Centro Universitario del Norte (CUNorte).

11. Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación . (2016). *Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación* . Quito : Registro Oficial.
12. Constitución de la República del Ecuador . (2008). *Constitución de la República del Ecuador* . Quito: Cooperación de estudios y publicaciones .
13. Correa, C. (1994). *Derecho Informático* . Buenos Aires : Depalma .
14. Culebro, M., Gómez Herrera, W., & Torres, S. (2006). *Software libre vs software propietario*. México D.F.: Creative commons.
15. Empresarios . (2014). *Libertad: la nueva marca del software*. San José Costa Rica .
16. Estrategia para la implantación de Software Libre para la Administración Pública Central de Ecuador. (2009). *Estrategia para la implantación de Software Libre para la Administración Pública Central de Ecuador*. Quito: Presidencia de la República .
17. Figuerola, C. (2008). *Software libre y software gratuito para la innovación docente* . Salamanca España: Universidad de Salamanca.
18. García de Ceca, M., & Verdú, W. (2012). *Software libre para el control y gestión de los procesos administrativos* y. Caracas Venezuela: Universidad Nueva Esparta.
19. García, D. (22 de 07 de 2013). Instituciones estatales gastan sumas de dinero “inciertas” por software que podrían tener gratis. *crhoy*, pág. 4.
20. Grupo Océano. (2013). *Océano Uno Color Diccionario Enciclopédico*. Barcelona España: Editorial Océano, S.L.
21. Hispavista. (2011). *Galeon.com*. Obtenido de <http://estuinge.galeon.com/>
22. Ley de Propiedad Intelectual . (2011). *Ley de Propiedad Intelectual* . Quito: Cooperación de estudios y publicaciones .
23. Manovich, L. (2013). *El software toma el mando*. Barcelona España : UOC.

24. Moisset, L., & Hiruela, M. d. (2010). *Protección Jurídica del Software*. Buenos Aires-Argentina : Astrea .
25. Molina, J., & Baena, L. (2007). *Sistemas Operativos en Entornos monousuario y multiusuario, Windows 2003 Server y Linux*. Madrid España: Vision net .
26. Muro, L. (2007). *Licencias de Software*. Caracas Venezuela : Universidad Nueva Esparta.
27. Peña, D. (2008). *Software Libre y Software Propietario: Impacto Jurídico, Económico y Cultural en Colombia* . Bogota : Universidad Externado de Colombia .
28. Pérez, J. (2008). *Definición de software* . Definición de software (<http://definicion.de/software/>).
29. Plan Nacional de Gobierno Electrónico . (2014). *Plan Nacional de Gobierno Electrónico* . Quito : Secretaría Nacional de la Administración Pública .
30. Plan Nacional de Seguridad Integral. (2011). *Plan Nacional de Seguridad Integral*. Quito : Ministerio de Coordinación de Seguridad .
31. Plan Nacional del Buen Vivir 2013-2017. (2014). *Seguridad Integral plan y agendas 2014-2017*. Quito: SEMPLADES.
32. quees.info. (2015). *quees.info*. Obtenido de quees.info: <http://www.quees.info/que-es-software.html>
33. Real Academia Española. (2014). *Diccionario de la Real Academia Española*. Madrid España: RAE.
34. Retana, Ó. (2010). *Por qué, cómo y dónde aprovechar software libre, sin morir en el intento*. San José Costa Rica: Gridshield.
35. Sánchez, V. (2005). *La protección del software*. Quito : Representante de la BSA, Alianza Antipirateria.

36. Sommerville, I. (2005). *Ingeniería del software*. Madrid España : Pearson Educación S.A.
37. Stallman, R. (2004). *Software libre para una sociedad libre*. . España: Ed. Traficantes de Sueños.
38. Stallman, R. (2015). *El software libre es ahora aún más importante*. New york: Free Software Foundation.

## ANEXO 1:

Software libre permitirá hacer realidad la soberanía tecnológica en Ecuador



09 Julio 2015

Por Galo Sosa

Quito (Pichincha).- En la Comisión de Educación de la Asamblea Nacional organizaciones y actores sociales expusieron sus puntos de vista sobre el Código INGENIOS.

El organismo recibió a los representantes de Soporte Libre, de la Asociación de Software Libre del Ecuador (ASLE), de la Industria Energética Inllyaku y del Colectivo Conéctate al Aire, los que coincidieron en destacar que el uso y funcionamiento regular del software libre permitirá hacer realidad la soberanía tecnológica de nuestro país, dando lugar a la sustitución de importaciones y

contribuir, así, al cambio de la matriz energética.

Juan Diego Palacio, del Colectivo Conéctate al Aire, afirma que se sienten orgullos porque INGENIOS abre espacios a la juventud, incluso a menores de edad, para crear cosas y que no hay ninguna diferencia entre un emprendimiento ecuatoriano y un internacional, tan solo es el medio en que se desarrollaron los científicos y creativos.

Ricardo Argüello, de Soporte Libre, manifestó que es necesaria la adopción de software libre en el Estado ecuatoriano y que su entidad ha demostrado que sí es posible tener un modelo de negocio basado completamente en software libre de lado de la empresa privada. La ciudadanía tiene que convertirse en actor participativo del software libre y de la industria, lo que dará paso a la democratización, descentralización e innovación.

Rafael Bonifaz, de ASLE, sugiere dejar de formar oficinistas y definir personas que piensen y se empoderen del conocimiento. Entre tanto, Héctor Velasco, gerente general de la Industria Energética Inllyaku S.A., propuso que el código beneficie a emprendimientos tanto de los que trabajen de la mano del Estado, como en general privados, al pedir nuevas reglas de financiamiento y flexibilización en la contratación de personal a favor de las iniciativas nacionales, para multiplicar el proyecto de los calefones eléctricos que promueven desde hace algunos años. /

Asamblea – AVN El Ciudadano

Tomado de página web: <http://www.elciudadano.gob.ec/software-libre-permitira-hacer-realidad-la-soberania-tecnologica-en-ecuador/>

## ANEXO N° 2

### PROCESO DE MIGRACIÓN A SL EN LAS INSTITUCIONES DEL SECTOR DE CONOCIMIENTO Y TALENTO HUMANO



Henry Vallejo 2016-03-04 Comentarios desactivados en PROCESO DE MIGRACIÓN A SL EN LAS INSTITUCIONES DEL SECTOR DE CONOCIMIENTO Y TALENTO HUMANO

En el marco de la estrategia del cambio de la matriz productiva hacia una economía social del conocimiento, el Consejo Sectorial de Conocimiento y Talento Humano, presidido por el Econ. Andrés Arauz, aprobó la resolución que permitirá avanzar en la migración a software libre de los entornos de escritorio de todas las instituciones pertenecientes al sector como: la Secretaría de Educación Superior, Ciencia,

Tecnología e Innovación (SENESCYT), Ministerio de Cultura y Patrimonio (MCYP), Ministerio de Educación (MINEDUC), Ministerio Coordinador de Conocimiento y Talento Humano (MCCTH), y las respectivas entidades adscritas a estas Carteras de Estado, entre otras.

Una de las experiencias exitosas que se destacan en este proceso, es la gestión realizada en el Instituto Geográfico Militar (IGM), que ya cuenta con la migración del 80% de sus sistemas informáticos a software libre. De este proceso de transformación tecnológica también es parte el Servicio Ecuatoriano de Capacitación Profesional (SECAP) y gracias a su experiencia, ofrecerá capacitación en el manejo de software libre a las instituciones del sector.

La resolución mencionada se suscribió hoy, 3 de marzo de 2016, en cumplimiento del Decreto Ejecutivo N° 1014, en el que el Presidente de la República, Rafael Correa, establece como política pública para las entidades de la Administración Pública Central la utilización de software libre en sus sistemas y equipamientos informáticos, considerando el interés del Gobierno por alcanzar la soberanía y autonomía tecnológica, así como un significativo ahorro de recursos, pues se considera que, desde el 2008 hasta el 2015, el gasto en software y sistemas operativos privativos en el sector público alcanzó un costo aproximado de \$43 millones de dólares.

Esta importante decisión, cuyo plazo de implementación es de 10 meses, responde a los esfuerzos del Gobierno para impulsar la soberanía tecnológica, la seguridad, la optimización de los recursos, sostenibilidad a largo plazo, socialización del conocimiento, fortalecer la industria tecnológica nacional de software libre, evitando

la salida de divisas, a través del fortalecimiento del talento humano, desarrollo de competencias y capacidades en materia de programación. Parte de este reto, también se estableció el compromiso de promover la enseñanza de programación informática en todos los niveles de educación.

“El conocimiento y el talento humano son los pilares fundamentales para promover una sociedad justa y solidaria”.

Ministerio Coordinador de Conocimiento y Talento Humano.

Tomado de la página web: <http://www.asle.ec/proceso-de-migracion-a-si-en-las-instituciones-del-sector-de-conocimiento-y-talento-humano/>