

PROTECCION JURIDICA DEL SOFTWARE

Contenido

INTRODUCCION	2
¿Qué es el software?	2
Código objeto y código fuente.....	3
Clasificación del Software	4
PROTECCION JURIDICA DEL SOFTWARE POR MEDIO DEL DERECHO DE AUTOR	5
Evolución de la protección jurídica del software por medio del derecho de autor.....	5
Uruguay y la protección jurídica del software por medio del derecho de autor	7
¿Qué aspectos del Software protege el derecho de autor?	7
¿Cuáles elementos no literales NO se protegen por medio del derecho de autor?	9
PROTECCION JURIDICA DEL SOFTWARE POR MEDIO DE LA PATENTE.....	12
EEUU y su protección jurídica del software por medio de la patente	12
Evolución de la protección jurídica del software por medio de la patente en los EEUU	12
Caso Diamond.....	13
Caso Arrhythmia Research.....	14
Caso State Street Bank.....	14
Europa y su protección jurídica del software por medio de la patente.....	15
Uruguay y su protección jurídica del software por medio de la patente.....	15
PROTECCION JURIDICA DEL SOFTWARE POR MEDIO DEL SECRETO COMERCIAL O TRADE SECRET	16
EEUU y su protección jurídica del software por medio del trade secret	17
Protección del código fuente y código objeto bajo el instituto del trade secret americano.....	18
Trade secret en los ADPIC. Uruguay y su protección al software por medio del trade secret o secreto comercial	18
Conclusión	20

INTRODUCCION

Este trabajo tiene como objeto analizar las distintas posibilidades que ofrece el ordenamiento jurídico uruguayo para proteger al software. Para obtener una comprensión cabal sobre esta materia, analizaremos también algunas de las normas y casos jurisprudenciales de los EEUU, por ser éste un país pionero en la regulación de este tema.

Es así que analizaremos el instituto del derecho de autor, el de la patente y el del secreto comercial como posibles herramientas para proteger al software. Se analizarán los requisitos que debe de cumplir el software para ser protegido por alguno de estos institutos, así como también la protección que le brindan los mismos.

¿Qué es el software?

Por medio del término software, programa o programa de ordenador¹ se hace referencia a un conjunto de instrucciones escritas que, al ser implementadas e interpretadas por una computadora - hardware- producen cierto resultado. Por tanto, software es cualquier instrucción que tiene el objetivo de ser usada por un hardware, máquina de ordenador o computadora², con el fin de obtener un resultado determinado. Simplificando todavía más, software son instrucciones que se le dan a cualquier hardware para producir un resultado determinado.

Según Baylos Corroza, la máquinas de ordenador son características de la era moderna y se diferencian ellas de las máquinas de la era industrial, las cuales producían y producen siempre la misma operación para producir siempre un mismo resultado. En cambio, las máquinas de ordenador se caracterizan por su versatilidad, pues ellas producen siempre un resultado distinto según la finalidad que se persiga y según las instrucciones que recibe. No actúan estas máquinas de ordenador por si mismas en virtud de sus engranajes y piezas, sino que ejecutan las instrucciones que se le dan. Por ello dice este autor que estas máquinas de ordenador o hardware hacen lo que se les manda a hacer en cada caso.³

Aclaremos un poco los términos. Ordenador es aquello que comprende tanto al hardware y al software. El hardware, o parte dura, es todo aquello que tiene una materialidad y está entonces integrado por componentes materiales, piezas, circuitos integrados, y demás; software es la parte blanda del ordenador y no tiene ninguna materialidad sino que está constituido por los elementos intelectuales destinados a regir el funcionamiento del ordenador.⁴

¹ En este trabajo los términos Software, Programa y Programa de Ordenador serán utilizados como sinónimos.

² En este trabajo los términos Hardware, Computadora y Máquina de Ordenador serán utilizados como sinónimos.

³ Baylos Carroza, Tratado de Derecho Industrial, 2da edición, pág. 629.

⁴ Idem

El objeto de estudio de este trabajo es esta segunda parte inmaterial de todo ordenador, llamado software, programa, o programa de computación.

Hay numerosas definiciones de software. Veamos alguna de ellas para ilustrar más claramente el objeto de estudio de este trabajo.

La Organización Mundial de la Propiedad Intelectual define al Software por medio de un grupo de sus expertos en la materia como *“un conjunto de expresiones expresadas mediante palabras, códigos o planes o en cualquier otra forma que, al ser incorporadas en un dispositivo de lectura automatizada, es capaz de hacer que un ordenador –un aparato eléctrico o similar capaz de elaborar informaciones- ejecute determinada tarea u obtenga determinado resultado”*.⁵

En el año 1980 los EEUU introducen una modificación a su ley de derechos de autor incorporando la definición de programas de computación como *“un conjunto de instrucciones para ser usados directa o indirectamente por una computadora con el fin de obtener determinado resultado”*⁶

Código objeto y código fuente

El software o programa contiene instrucciones dirigidas al ordenador para obtener determinado resultado. Estas instrucciones que tiene todo software son las llamadas código fuente y las instrucciones llamadas código objeto. Pero no es que cada software o programa tenga uno u otro tipo de instrucciones, sino que por el contrario todo software comprende inevitablemente un código fuente y un código objeto. Veamos qué son cada una de estas instrucciones.

Código fuente son aquellas instrucciones escritas por los seres humanos. Son por tanto legibles y entendibles para el ser humano. El software se escribe o se crea, en una primera instancia, en este lenguaje. Estas instrucciones tal como están escritas no pueden dirigirse al hardware, puesto que éste no entiende este lenguaje y es entonces necesario que estas instrucciones se traduzcan o se conviertan en otro tipo de instrucciones que si sean interpretables y entendibles por el hardware. Estas otras instrucciones son las llamada código objeto que veremos en el siguiente párrafo.

El código objeto, o código de máquina, es el resultado de la transformación del código fuente -el escrito por el ser humano- al código objeto, con el fin de que el hardware pueda entender y ejecutar las instrucciones, ya que, como recién dijimos, el hardware no entiende las instrucciones dadas por el código fuente. Por ello, luego de escribir el código fuente, el mismo debe ser traducido o transformado a código objeto, por medio de un proceso automatizado que lleva a cabo el hardware.

En palabras de A. Liberman, citado por Baylos Carroza, las instrucciones en versión código objeto que son invisibles, pueden ser expresadas en forma visible para el hombre gracias al método llamado de numeración binaria que consiste en representar los impulsos magnéticos por las cifras 0 o 1.⁷

⁵ Definición incluida en el art. 1 del Proyecto de Disposiciones Tipo para Leyes en Materia de Derecho de Autor, de la OMPI.

⁶ U.S. Code, Título 17, artículo 101

Explicando todavía un poco más en profundidad lo que es este código objeto, Paul Henri Steinauer, citado por Baylos Carroza, explica que “la base de actuación del ordenador está en la lógica formal que es lógica pura porque hace abstracción del contenido de los razonamientos para valerse de su estructura, comprendiéndose como lógica algorítmica el cálculo de términos y el cálculo de enunciados y en que solo existen las nociones de verdad y falsedad que integran la lógica apropiada, porque la mayor parte de los ordenadores están basados en el hecho de que una corriente eléctrica pasa, o de que no pasa (la verdad en el ordenador es 1, la falsedad es 0)”⁸

Es así que decimos que código objeto no es más que una continuidad de unos y ceros o letras y números (sistema binario) y es, como dijimos anteriormente, inentendible para cualquier versado en la materia.

Pero que quede claro, las instrucciones que sigue el hardware o computadora son las llamadas código objeto y no las llamadas código fuente.

Clasificación del Software

A grandes rasgos y clasificándolos por su funcionalidad, podemos decir que hay dos tipos de software o programas. Están primero los programas llamados de aplicación y los programas llamados de operación de sistemas. Los programas de aplicación son aquellos que hacen determinada actividad para el usuario de la computadora o hardware, como por ejemplo el Word, Excel o Autocad. Los programas llamados de operación de sistemas son aquellos que administran las funciones internas del hardware o facilitan el uso de los programas de aplicación.

Un ejemplo de programa operación de sistema son aquellos programas que traducen el código fuente a código objeto, traducción necesaria, como ya vimos, para que todo programa sea entendido y funcione en determinada computadora.

Como a todos alguna vez les ha pasado, muchas veces un software no es compatible con determinado hardware (muchos programas de Microsoft no son compatibles y por tanto no pueden ser utilizados por los hardware marca Apple) y por tanto no se puede usar en una computadora determinada. Aquí es donde entran en juego los programas operacionales de sistemas que hacen posible que las instrucciones de un programa sean entendidas por un determinado hardware. En otras palabras, estos programas operacionales son también software aunque desconocidos como tales por muchos usuarios comunes que no tienen un conocimiento profundo en la materia.

⁷ Baylos Carroza, Tratado de Derecho Industrial, 2da edición, pág. 630

⁸ Idem, pág. 630.

PROTECCION JURIDICA DEL SOFTWARE POR MEDIO DEL DERECHO DE AUTOR

Evolución de la protección jurídica del software por medio del derecho de autor

En sus orígenes no fue pacífica la decisión de proteger al software por medio del derecho de autor. Los autoristas se oponían a la inclusión del software como derecho de autor pues el software carece -según ellos- de todo valor artístico o literal.⁹

A su vez, en la década del 60, cuando los programadores se acercaron a la Oficina de Copyright de los EEUU, sus funcionarios comenzaron a emitir certificados de registros pero bajo “la regla de la duda”. Esta regla de la duda significaba que los funcionarios no estaban seguros de que el programa objeto o de máquina fuera pasible de protección por medio del derecho de autor. Estos funcionarios tomaban nota de la diferencia entre el texto fuente de los programas y de los ejecutables -o código objeto-, de su carácter utilitario y de la imposibilidad de ser leídos por seres humanos.¹⁰ Es decir que la Oficina de Copyright de los EEUU en una primera etapa tuvo la duda en cuanto a si el código ejecutable o código objeto podía ser protegido por medio del derecho de autor (o Copyright, según su terminología). Pero parece que esta oficina no tuvo nunca dudas acerca de la posibilidad de que los programas fuente fueran pasibles de ser protegidos por las normas del derecho de autor.

El congreso de EEUU en el año 1975 crea el establecimiento de la Commission in New Technological Uses of Copyrighted Works (CONTU), la cual debía de estudiar y hacer recomendaciones al congreso de los EEUU sobre aspectos de nueva tecnología. Entre sus fines, esta comisión debió recomendar la protección que debía dársele a los programas de ordenador o software. El informe de la CONTU al respecto, expresaba que la forma inevitable de protección del software era por medio del Copyright y cualquier representación del software debía de ser protegido por dicho instituto jurídico -el Copyright-. No manifestó esta comisión ninguna cuestión sobre la protección sui generis que podría proteger al código ejecutable -código objeto o de máquina-¹¹. Por tanto, y al entender de esta comisión, todo el software -sus códigos fuente y de máquina- debían ser protegidos por el Copyright. Es así que el congreso de los EEUU sanciona una nueva ley en el año 1980 la cual introduce la definición de programas de computación en su artículo 101 (ley de Copyright americana).¹² Esta definición, como ya vimos, expresamente informa que los programas de computación están compuestos por el código fuente y por el código objeto o de máquina. A pesar de que la ley de Copyright americana no incluya en su artículo 102 (a) específicamente a los programas de computación como materia protegida por este instituto, la historia legislativa del congreso americano sugiere que dichos programas son pasibles de protección por medio del Copyright como obras literarias.¹³ Los programas de computación quedan

⁹ Delia Lipszyc, Derecho de autor y derechos conexos, pág. 106

¹⁰ Beatriz Busaniche, El Dilema del Copyright en el campo del Software, pág. 2, disponible en <http://www.bea.org.ar/2010/08/el-dilema-del-copyright-en-el-campo-del-software/>

¹¹ Idem, pág. 3

¹² Delia Lipszyc, Derecho de autor y derechos conexos, pág. 107

¹³ Apple Computer, Inc. V. Franklin Computer Corp., 714 F.2d 1240 (1983)

entonces comprendidos en los trabajos literarios, que si son nombrados expresamente en el art. 102 (a) de la ley de derecho de autor americana.

Los estudios realizados en la década de 1970 por el profesor alemán Eugen Ulmer tuvieron gran influencia en la materia. Este autor concluye en sus estudios, los cuales fueron hechos a pedido de la Unesco y de la OMPI, que la protección del software por medio de derecho de autor es perfectamente posible. Según Delia Lipszyc, esta opinión fue la que se impuso y expresa a su vez esta autora que “se reconoció que el programa de ordenador constituye una obra del intelecto resultante de un proceso creativo similar al que demanda una obra literaria o un libro cinematográfico que, cuando es original en su composición y en su expresión, está protegido por el derecho de autor”¹⁴

Es así que en el año 1980 el software quedó definitivamente comprendido en la ley del Copyright norteamericana, como ya hemos dicho. Luego, los EEUU lanzaron su ofensiva para que otras jurisdicciones tomaran el mismo camino. Es así que otros países tomaron la misma decisión de EEUU en cuanto a la protección del software por medio del derecho de autor: Hungría, Australia, Japón, India, Alemania, Francia, Reino Unido, Canadá, Corea del sur, Indonesia, entre otros.¹⁵

La Unión Europea en el año 1991, por medio de la Directiva 91/250 establece, a su vez, de manera muy similar al art. 10 de los ADPIC, que el software debe ser considerado como una obra literaria y, por ende, protegido bajo el régimen de derechos de autor o Copyright. De esta manera Europa se suma en el año 1991 al club de las jurisdicciones que reconocían específicamente al derecho de autor o Copyright como instituto jurídico apto para proteger a los programas de ordenador o software.

Finalmente, en el marco de la ronda de Uruguay del GATT se firmaron los ADPIC, el cual establece que el software será protegido como obra literaria en virtud del Convenio de Berna.¹⁶ Ello significa que según los ADPIC el software debe ser protegido por medio del derecho de autor. Los ADPIC son un acuerdo obligatorio para todos los miembros de la OMC y es por ello que todo miembro de esta organización debe proteger al software por medio de las normas del derecho de autor o Copyright.¹⁷ En este sentido, los ADPIC dispusieron que todos sus miembros deberían de adaptar sus legislaciones internas en este sentido, so pena de sufrir sanciones comerciales.

A su vez la OMPI, en el año 1996, por medio del Tratado OMPI sobre Derechos de Autor -uno de los “Tratados de Internet”- dispone en su artículo 4, que “Los programas de ordenador están protegidos como obras literarias en el marco de los dispuesto en el artículo 2 del Convenio de Berna”. Dicha protección se aplica a los programas de ordenador, cualquiera que sea su modo o forma de expresión”.

¹⁴ Delia Lipszyc, Derechos de autor y derechos conexos, pág. 107

¹⁵ Delia Lipszyc, Derechos de autor y derechos conexos, pág. 108 y Martín Carranza, El Derecho de la Innovación Tecnológica, pág. 220.

¹⁶ ADPIC, art. 10.1.: “Los programas de ordenador, sean programa fuente o programas objeto, serán protegidos como obras literarias en virtud del Convenio de Berna (1971)”.

¹⁷ Artículo 2 del acuerdo por medio del cual se establece la Organización Mundial del Comercio.

Uruguay y la protección jurídica del software por medio del derecho de autor

En Uruguay la ley 17.616 del año 2003, en su artículo 3, reconoce a los “Programas de ordenador, sean programas fuente o programas objeto” como protegibles bajo las normas del Derecho de Autor.

Sin perjuicio de este reconocimiento expreso que hizo la mencionada ley, perfectamente se puede sostener que antes de la vigencia de la ley 17.616, la ley 9.739 ya protegía a los programas de ordenador o software, ya que éstos se asemejan a las obras literarias o a los “escritos de toda naturaleza”.¹⁸ Así lo entendió el Tribunal de Apelaciones en lo Civil de 4 Turno, quien manifiesta por sentencia nº 134 del 2006 que el programa de ordenador constituye una obra del intelecto la cual es protegida por el derecho de autor y ello aún antes de la sustitución del art. 5 de la ley 9.739 por el art. 3 de la ley 17.616.¹⁹

Además y también antes de la ley 17.616, cabe destacar que el Poder Ejecutivo del Uruguay, por Decreto Nº 154/989 del 11 de abril de 1989, ratificó el criterio del Consejo de Derecho de Autor de que los programas de computación eran inscribibles en el Registro de Derecho de Autor por entender que estos trabajos eran materia susceptible de protección bajo la ley 9.739.²⁰

¿Qué aspectos del Software protege el derecho de autor?

1. Elementos literales: el código Fuente y el código objeto

La ley 9.739 en su artículo 5, texto dado por el art. 3 de la ley 17.616, manifiesta que “la producción intelectual, científica o artística comprende: Programas de ordenador, sean programas fuente o programas objeto”.

A su vez los ADPIC, en su artículo 10, expresan que “Los programas de ordenador, sean programas fuente o programas objeto, serán protegidos como obras literarias en virtud del Convenio de Berna (1970)”.

Este tratado fue ratificado por el Uruguay por el decreto- ley 16671 del año 1994. Además, los ADPIC, como ya lo hemos dicho, son obligatorios para todos los países miembros de la OMC y entonces en todos ellos rige –o debería regir- este tratado.

Por ello es claro que el derecho de autor no solo protege al código fuente, que es aquel escrito por el ser humano y comprensible por él, sino que también protege al código objeto que es una

¹⁸ “Escritos de toda naturaleza” es la terminología que usó el legislador en el artículo 5 de la ley 9.739, que fue sustituido por la ley 17.616

¹⁹ LJU SUMA 139017, Tomo 139, año 2009.

²⁰ Mario Barreto, Evolución del Tema Protección Jurídica del Software, disponible en la base de datos on line La Ley.

transformación del código fuente y el cual no es comprensible para el ser humano. Por tanto, y de acuerdo al derecho positivo vigente, el derecho de autor protege a ambos códigos.

El derecho americano también manifiesta expresamente que el código objeto es también pasible de ser protegido por el derecho de autor. Es así que en su definición de programa de computación expresa que éste es un conjunto de instrucciones para ser usadas *directamente* o *indirectamente* por una computadora. Cuando esta definición habla de “directamente”, se refiere al código objeto, puesto que estas instrucciones son leídas y ejecutadas sin más por el hardware. En cambio, cuando esta definición habla de “indirectamente” se refiere al código fuente, puesto que estas instrucciones deben ser primero traducidas o transformadas al código objeto para ser entendidas y ejecutadas por el hardware.

Se configurará una infracción a las normas del derecho de autor aunque no se copie enteramente el código fuente u objeto, ya que de la misma manera que un plagio no deja de ser tal por la simple sustitución de alguna palabra, la infracción a los derechos de autor sobre un software no es evadida por una modificación menor de alguno de sus códigos. No se requiere entonces una copia íntegra del código para que se configure una infracción. En estos casos se deberá de hacer un análisis caso a caso para establecer si hay o no una similitud substancial entre los programas que acredite una infracción.

2. Elementos no literales del Software

Si bien teóricamente sería posible la protección de los elementos no literales del software, en la práctica estos elementos carecen de protección por medio de este instituto jurídico.

Decimos que teóricamente estos elementos podrían ser protegidos por medio del derecho de autor, puesto que si estos elementos constituyen una expresión de una idea y ellos caen dentro de lo que es pasible de protección por estas normas, podrán sí ser protegidos por el Derecho de Autor .

El Derecho de autor protege, en palabras de Delia Lipszyc , las creaciones de autores que presenten individualidad resultantes de su actividad intelectual ²¹ Esta individualidad puede plasmarse en los que se llaman los elementos no literales del software. Sin embargo y como veremos en los siguientes párrafos, es muy difícil proteger a estos elementos por medio del derecho de autor.

Los elementos no literales se extienden sobre elementos del software distintos a los literales. Los elementos no literales son la estructura o la manera en que un software está organizado. Ellos están dados por los diagramas de flujo generales así como por la organización más específica de relaciones entre módulos, listas de parámetros y macros.²² Estos elementos también merecen la protección del derecho de autor en cuanto ellos consistan en expresión y no meramente en ideas.

²¹ Delia Lipszyc, Derecho de autor y derechos conexos, pág. 11

²² Computer Associates v Altai 982 F2d 693 (1992)

Es así que podrá haber infracción sobre los derechos de autor a un determinado software por más que no se haya copiado los códigos de manera literal, y entonces quedará acreditada dicha infracción cuando el infractor copie algo que pertenece a la individualidad o creatividad de su autor.

Con los libros sucede algo similar. Se puede infringir un libro cuando se copie de él no solo sus términos y sus letras, sino que también se configurará una infracción cuando se copien elementos no literales del libro pero que igual hacen a la individualidad del autor. Si todos los capítulos de un libro están organizados de la misma manera, en el mismo orden, con el mismo nombre, y a su vez con los mismos personajes y la misma trama, entonces diremos que el copiadador tomó más de lo que podía tomar de la obra original y diremos entonces que habrá infringido los derechos de autor del creador del libro.

Lo mismo sucede entonces con el software. La protección del mismo no es solamente respecto a sus términos o elementos literales, y entonces habrá que ver si el programador de un código posterior en el tiempo, a pesar de no haber hecho una copia literal del mismo, copió los aspectos no literales de un código anterior al cual tuvo acceso.

Por tanto, si bien entendemos que teóricamente los elementos no literales del software pueden ser protegidos por medio del derecho de autor, es muy difícil que ellos no caigan en alguna de las cosas que no protege este instituto jurídico y que trataremos más adelante en el capítulo “¿Cuáles elementos no literales NO se protegen por medio del derecho de autor?”.

La jurisprudencia americana es muy estricta en dar protección a estos elementos. Para que los elementos no literales sean pasibles de protección deben de superar un estricto análisis que muy pocas veces termina protegiendo a los elementos no literales. Por más información de este análisis, ver caso *Computer Associates International Inc. V. Altai, Inc.*, 982 F. 2d 693 (1992), EEUU.

Aquí queremos mencionar la opinión de Martín Carranza, quien en su claro libro “El derecho de la Innovación Tecnológica” manifiesta lisa y llanamente que los elementos no literales del software NO son protegibles por el derecho de autor²³, y quizás tenga razón, puesto que el análisis que deben superar los elementos no literales, por lo menos a nivel jurisprudencial en los EEUU, es muy arduo y la protección que reciben es prácticamente nula (ver Altai).

¿Cuáles elementos no literales NO se protegen por medio del derecho de autor?

Para responder esta pregunta nos basamos en el famoso caso ya citado *Computer Associates International Inc. v. Altai, Inc.* Este caso tenía como meta establecer en qué medida y bajo qué condiciones se otorgaría protección a los elementos no literales del software.

De este análisis, como veremos a continuación, concluiremos que la protección de los elementos no literales del software por medio del derecho de autor es muy restrictiva y prácticamente nula. Es por ello que Martín Carranza expresa que los elementos no literales del software no son protegidos por este

²³ Martín Carranza, *El Derecho de la Innovación Tecnológica*, pág. 223

instituto jurídico, y que lo que único que protegen esta normas -las del derecho de autor- son los elementos literales, como el código fuente y el código objeto.²⁴

Entonces aquí radica la mayor deficiencia del derecho de autor para proteger eficazmente al software: dicho instituto no protege a los elementos no literales del software, que muchas veces constituyen la parte verdaderamente importante del software.

Veamos, a la luz del caso *Altai*, cuáles son los elementos no literales que NO tienen protección bajo las normas del derecho de autor o Copyright.

Ideas

Es un principio angular del derecho de autor que las ideas no son pasibles de protección por medio de este instituto -y tampoco lo son por ningún instituto del derecho de propiedad intelectual o industrial-.

Imaginemos un software que comprenda una idea fantástica, nueva y revolucionaria. Este software obtendrá protección de sus códigos -fuente y objeto- en cuanto sean originales, pero nunca obtendrá protección de la idea implementada por medio del software. Esta idea estará siempre en el dominio público y por tanto libre de ser usada por cualquier persona.

Es por esto que se dice que el derecho de autor no protege a una idea, sino que protege a la expresión de una idea. Hablando sobre software, la expresión de la idea no sería otra cosa que la elaboración del código fuente y su correspondiente código objeto.

Elementos que son incidentales o necesarios para la realización del fin que se persigue con el software

Ningún elemento que sea necesario para realizar la idea que persigue el software es pasible de protección por las normas de derecho de autor.²⁵

En este sentido la CONTU, en su informe al congreso de EEUU, manifestó que el lenguaje -código- protegido por el derecho de autor (o Copyright, según su terminología) se puede copiar sin infringir dichos derechos de autor cuando haya una manera limitada de expresar la idea. Esto significa que cuando las instrucciones de un software (código fuente y código objeto) sean la única manera -o de las únicas- de realizar lo que el software se propone -la idea-, esas instrucciones (ese software) no serán pasibles de protección por medio del derecho de autor y pertenecerán entonces al dominio público.²⁶

Cuando la idea de un software pueda ser llevada a cabo por medio de la escritura de distintas instrucciones o códigos, diremos entonces que cada uno de esos códigos constituye expresión pasible de ser protegida por medio del derecho de autor.

²⁴ Martín Carranza, *El Derecho de la Innovación Tecnológica*, pág. 224

²⁵ *Computer Associates v. Altai*, 982 F.2f 693 (1992)

²⁶ Informe de la CONTU citado en el caso *Altai*, ya mencionado en este trabajo.

Pero cuando los elementos de un software –elementos literales y no literales- son necesarios para realizar la finalidad perseguida por éste, diremos que la expresión se unió con la idea y que por tanto esa expresión deja de ser pasible de protección por medio de las normas de derecho de autor. Cuando las expresiones –protegibles por el derecho de autor- se unen de manera tal que no se puedan separar de la idea, diremos entonces que la expresión deja de ser expresión y pasa a formar parte de la idea. En estos casos, estas expresiones, al unirse con la idea, dejan de ser protegidas por el derecho de autor, puesto que las ideas no son protegidas por medio de este instituto jurídico.²⁷

Lo que sucede en el campo del software es que muchas veces hay pocas maneras de implementar eficazmente una tarea y entonces es probable que distintos programadores trabajando de manera independiente lleguen al mismo resultado. De ser así, ninguno de esos trabajos podría ser protegido por las normas del derecho de autor por la interconexión entre de la idea y la expresión.²⁸

Elementos dados por factores externos

Al desarrollar un programa los programadores toman en cuenta factores externos que deben incluir en el diseño del programa. Estos factores entonces integran el programa no por la decisión libre del programador de elegir determinado elemento dentro de otros tantos, sino que muchas veces se ven obligados a tomar ciertos elementos para su programa por alguna de las siguientes razones: 1. aspectos mecánicos de la computadora para la cual el programa está dirigido; 2. Requisitos de compatibilidad de otros programas con los cuales está diseñado para funcionar en conjunto; 3. Estándares de diseño de los fabricantes de computadoras; 4. Demandas de la industria a la que apunta el programa; 5. Prácticas de programación ampliamente aceptadas en la industria informática.²⁹

Cualquier elemento de un programa que se deba a una de estas 5 razones recién mencionadas, no gozará de protección alguna de las normas del derecho de autor.

Elementos tomados del dominio público

Esto tampoco es ninguna especificidad del Software, sino que es un principio del derecho de autor que todo elemento que se encuentre en el dominio público es libre de ser usado por cualquier persona sin necesidad de recabar autorización de nadie.

²⁷ Con esto no nos estamos adhiriendo a la teoría que expresa que en un análisis de infracción de derecho de autor basta con identificar la idea de un software para manifestar luego que si hay varias maneras de llevar a cabo esa idea, por medio de distintos códigos, el código primario es protegible por medio del Derecho de Autor. Esta teoría simplista fue impuesta en el caso *Whelan Associates v. Jaslow Dental Laboratory*, (797 F.2d 1236) y muy criticada por la jurisprudencia americana posterior que entiende que todo software no tiene una sola idea, si no varias.

²⁸ *Computer Associates v. Altai* 982 F.2d 693 (1992)

²⁹ *Idem*

Es así que de haber algún elemento del dominio público en un programa, el mismo no gozará de la protección de las normas del derecho de autor.

PROTECCION JURIDICA DEL SOFTWARE POR MEDIO DE LA PATENTE

Como vimos, los ADPIC expresamente establecen que el software deberá ser protegido por medio del derecho de autor en virtud del Convenio de Berna,³⁰ y como consecuencia de ello numerosas jurisdicciones modificaron sus normativas internas para cumplir con esta norma.

Sin embargo, los ADPIC no se refieren expresamente a si el software debe o no ser protegido por medio de las patentes, pero si expresan que “las patentes podrán obtenerse por todas las invenciones, sean de productos o de procedimientos, en todos los campos de la tecnología..”³¹. Se ha interpretado que la expresión “en todos los campos de la tecnología” comprende también al software y que entonces en los países miembros de la OMC le cabe al software también la protección por medio de la patente.³²

EEUU y su protección jurídica del software por medio de la patente

El Software en sí mismo no es materia patentable en los EEUU, ni en ninguna otra jurisdicción que conozcamos. Sin embargo, las invenciones basadas en software³³ -software-based inventions- si son patentables en este país, y por tanto este tipo de invenciones podrá recibir patentes si es que ellas cumplen con los demás requisitos de patentabilidad.³⁴

En el régimen de los EEUU el software puede llegar a constituir materia patentable como parte de una máquina o como parte de un proceso, si es que estas invenciones –basadas en software- producen una transformación física fuera de la computadora o que sin llegar a producir dicha transformación física, tengan una aplicación práctica útil.³⁵

Evolución de la protección jurídica del software por medio de la patente en los EEUU

En un principio la USPTO automáticamente rechazaba todas las solicitudes de patentes de invenciones basadas en software. Esta oficina rechazaba estas solicitudes de patentes por entender que el software no era materia patentable según el artículo 101 de la ley de patentes americana. EL software está constituido por algoritmos matemáticos, por fórmulas matemáticas. Los algoritmos matemáticos son considerados ideas abstractas o leyes naturales y ninguno de éstos constituye materia patentable, ni

³⁰ ADPIC art. 10

³¹ ADPIC art. 27.1

³² Martín Carranza, El Derecho de la Innovación Tecnológica, pág. 217

³³ Las invenciones basadas en software son también llamadas invenciones implementadas en computadora.

³⁴ Stephen Fishman, Web and Software Development, pág. 212

³⁵ Robert Merger y otros, Intellectual Property In The New Technological Age, pag 1064

en los EEUU³⁶ ni en ningún otro sistema jurídico que conozcamos. Es por esto que se rechazaban las invenciones basadas en software por entender que ellas recaían sobre ideas abstractas o leyes naturales -algoritmos matemáticos- y por tanto sobre materia no patentable.³⁷

En este sentido, la USPTO entendía que otorgar una patente sobre este tipo de invenciones sería lo mismo que otorgar una patente en el uso de la fórmula $1 \text{ más } 1 \text{ igual a } 2$, y es así que en una primera etapa no se concedían patentes a invenciones basadas en software.³⁸

Aunque se mantenga firme el hecho de que el software en sí mismo no es materia patentable – por constituir una idea abstracta o una ley natural-, la SCJ de los EEUU y otras cortes de alta jerarquía de este país ampliaron el espectro de lo patentable aceptando como materia patentable invenciones basadas en software siempre y cuando las mismas cumplan ciertos requisitos que explicaremos con los siguientes casos.

Caso Diamond

El primero caso en este sentido está dado por *Diamond V. Diehr* 450 U.S. 175 (1981). En este caso la SCJ de los EEUU³⁹ entendió patentable un proceso industrial que utilizaba un software para cumplir su función de curar caucho. Este software incorporaba una fórmula matemática conocida en el rubro de curar caucho, llamada Arrhenius equation. La SCJ entiende que al no estar esta patente dirigida a la fórmula matemática en sí misma –materia no patentable como recién hemos visto- sino que por el contrario al estar dirigida a un proceso industrial de curar caucho -el cual contiene diversos pasos y comprende la utilización de esta fórmula matemática en el software-, la presente invención constituye materia patentable. Entonces, dice esta corte, dicha patente no está dirigida a impedir que otros utilicen la fórmula matemática, si no que por el contrario está dirigida a impedir que otros usen esta fórmula matemática en conjunción con los otros elementos del proceso que se reivindica en esta patente. Estos otros elementos son: poner el caucho en una prensa, cerrar el molde, determinar constantemente la temperatura del molde, recalcular constantemente el tiempo adecuado para curar el caucho haciendo uso de esta fórmula matemática -llamada Arrhenius equation- y de la computadora, abrir la puerta en forma automática en el tiempo adecuado. En esta sentencia la SCJ de los EEUU manifestó algo que nos parece fundamental para entender el porqué de la concesión de este tipo de patente: una reivindicación que verse sobre materia patentable no dejará de serlo simplemente porque utilice una fórmula matemática, un programa de computación o una computadora digital. Sin perjuicio que esta sentencia fue novedosa, hay que tener en cuenta que en esta patente hay sí una transformación de la materia, puesto que por medio de este proceso el caucho pasa de un estado A a otro estado B.

³⁶ En el caso *Gottschalk v. Benson*, 409 U.S. 63 (1972) la SCJ de los EEUU manifestó que los algoritmos matemáticos son leyes de la naturaleza e ideas abstractas y por tanto no constituyen materia patentable.

³⁷ Robert Merger y otros, *Intellectual Property In The New Technological Age*, pag 1068.

³⁸ Stephen Fishman, *Web and Software Development*, 5ta edición, pág 211 y 212.

³⁹ Supreme Court of the United States, por su nombre in inglés

A partir del caso Diamond las invenciones basadas en software (o invenciones implementadas en computadora) fueron plenamente reconocidas en los EEUU, y entre los años 1982 y 1994 fueron concedidas en los EEUU unas 3500 patentes basadas en software.⁴⁰

Caso Arrhythmia Research

En el caso *Arrhythmia Research Technology Inc v. Corazonix Corp.*, 958 F.2d 1053 (1992), una corte federal de los EEUU⁴¹ admitió como materia patentable la siguiente invención: se ingresan los datos del corazón de un paciente a una computadora; un software analiza esa información de acuerdo con una serie de principios, emitiendo como resultado el estado de dicho corazón así como las posibilidades de éste de sufrir un ataque cardíaco. Aquí se transforma la información del corazón por medio de una idea abstracta –un algoritmo, una fórmula matemática- en información útil: el estado del corazón. Esto, al entender de esta corte, constituye la aplicación práctica de una idea abstracta (algoritmo matemático, fórmula o cálculo) porque crea un resultado útil, concreto y tangible, y por tanto patentable.

La patente en el caso *Arrhythmia* estaba dirigida al aparato y al proceso que permitían, mediante el software utilizado, saber el estado del corazón del paciente. Por tanto, ésta era una patente que contenía reivindicaciones dirigidas al producto o al aparato y reivindicaciones dirigidas al proceso.

Caso State Street Bank

En 1998 en el caso *State street Bank and Trust Co. V. Signature Financial Group, INC*, 149 F.3d 1368, (1998) la misma corte federal del caso *Arrhythmia* recién descrito, entendió que una máquina que empleaba un software para transformar la información representada en cantidades de dólares de un fondo común, a través de una serie de cálculos matemáticos, en una cifra que determinaba el precio final de ganancia, pérdidas y porcentajes, como un resultado útil, concreto y tangible, y por tanto también patentable. En este caso las reivindicaciones estaban dirigidas a una máquina que contenía un software que en conjunto producían un resultado útil. Aquí entiende esta corte que hay un resultado útil aunque este resultado se manifieste únicamente por medio de números representativos de ganancias, pérdidas y porcentajes. Aquí, como vemos, no hay ninguna transformación física sino solamente un resultado útil.⁴²

Por tanto, tenemos que en los EEUU no solamente puede llegar a constituir patente de invención una invención basada en software –sea como proceso o como máquina, pero nunca como software per se- que produzca alguna transformación física –caso del proceso de curar goma-, sino que también una invención basada en software puede llegar a ser patente –como máquina o como proceso-, aunque no haya ninguna transformación física, si es que dicha invención tiene una aplicación práctica, concreta y tangible. La gran duda será, en casos en que no haya transformación física alguna, el determinar en cada caso concreto si el resultado que produce una invención basada en software tiene el carácter de útil,

⁴⁰ Martín Carranza, *El Derecho de la Innovación Tecnológica*, pág. 212

⁴¹ United States Court of Appeals, Federal Circuit

⁴² Este caso fue muy importante en los EEUU porque no solo legitimó las patentes basadas en software, como hemos visto, sino también porque legitimó las patentes sobre métodos de hacer negocios en dicho país. Stephen Fishman, *Web and Software Development*, pág 212.

concreto y tangible (“useful, concrete and tangible result” en el idioma inglés). Si lo tiene, una invención basada en software podrá ser patente de invención en los EEUU, aunque no implique transformación física alguna de la materia, como sucedió en el caso recién mencionado de State Street Bank.

En palabras de Martín Carranza, “Hoy en día las propias directrices de la USPTO (Manual of Patent Examining Procedure -MPEP-) reconocen expresamente la patentabilidad del software bajo la condición de que tengan aplicación práctica, lo que, como ya se ha señalado, es un criterio muy amplio puesto que siempre el software tiene alguna aplicación de ese tipo”⁴³

Europa y su protección jurídica del software por medio de la patente

La solución europea es prácticamente idéntica a la americana recién desarrollada. Si bien la Convención Europea sobre patentes establece la prohibición del patentamiento de los programas de ordenador por medio de su artículo 52.2, en el artículo siguiente manifiesta que tal prohibición se refiere al patentamiento del programa de ordenador considerado como tal. Como consecuencia de esta normativa, en Europa se han otorgado una cantidad importante de invenciones basadas en software, de manera similar a lo que sucedió en los EEUU.

Es así que los jueces europeos han interpretado que lo que la Convención excluye es la patentabilidad de programas de computadores que no tengan un efecto técnico determinado, y como las invenciones basadas en software lo tienen, ellas son válidas.⁴⁴

El primer caso jurisprudencial en que se reconoció la validez de las invenciones basadas en software –o las invenciones implementadas en computadora- es de la misma época del caso comentado Diamond. A partir de esa fecha y hasta el 2002, la Oficina Europea de patentes habría otorgado más de 20.000 patentes de este tipo.⁴⁵

Uruguay y su protección jurídica del software por medio de la patente

La ley de patentes uruguaya, ley Nº 17.164, en su artículo 13, literal D manifiesta que: No se considerarán invenciones a efectos de la presente ley: D) Los programas de computación considerados aisladamente.

Al referirse la ley expresamente a la prohibición del patentamiento de los programas de computación *considerados aisladamente*, entendemos entonces que una invención que tenga entre unos de sus elementos un software o programa de computación será patentable.

Como lo dijo la SCJ de los EEUU en el caso Diamond, una invención que verse sobre materia patentable no dejará de serlo simplemente porque utilice una fórmula matemática, un programa de computación o una computadora digital.

⁴³ Martín Carranza, El Derecho de la Innovación Tecnológica, pág. 212

⁴⁴ Martín Carranza, El Derecho de la Innovación Tecnológica, pág. 212

⁴⁵ Idem.

Concluimos entonces que una invención que tenga entre uno de sus elementos un software, no por eso dejará de ser materia patentable bajo el derecho positivo uruguayo, sino que por el contrario podrá constituir una patente si cumple con los requisitos de novedad, actividad inventiva y aplicación industrial, y si es que dicha invención no se encuentra dentro de alguna de las prohibiciones comprendidas en los artículos 13 y 14 de la ley de patentes 17.164.

PROTECCION JURIDICA DEL SOFTWARE POR MEDIO DEL SECRETO COMERCIAL O TRADE SECRET

El secreto comercial o Trade Secret es otro instituto del derecho que ofrece, bajo ciertas condiciones y circunstancias, protección jurídica al software.

Estas normas pueden ser importantes para reforzar la protección jurídica del derecho de autor que ya se tiene sobre un software. Sin embargo y según veremos a continuación, la protección del Trade Secret no se puede acumular con la protección de la patente, puesto que un requisito esencial para recibir la protección del Trade Secret es que la información sea secreta. Por el contrario, un requisito esencial para obtener una patente es divulgar toda la información necesaria en relación a la invención.

Es así que un software puede recibir la protección jurídica que otorga el derecho de autor y a su vez la protección jurídica que ofrece el Trade Secret en aquellos países, claro está, en donde la protección del Trade Secret existe⁴⁶.

A esta altura es fundamental decir que los institutos de la propiedad intelectual no son excluyentes entre sí. Esto significa que una misma cosa, obra o invención, podrá ser pasible de protección por todos y cada uno de los institutos de la propiedad intelectual, si es que se cumple con los requisitos de cada uno de ellos. No hay ninguna norma a nivel nacional ni a nivel de derecho comprado que conozcamos que prohíba de manera general la acumulación de dichos institutos.⁴⁷

A continuación nos parece fundamental ver brevemente el instituto del trade secret en los EEUU, por ser éste un país donde este instituto está plenamente desarrollado. Luego, veremos la incorporación de dicho instituto en los ADPIC y con ello su incorporación a todos los países miembros de la OMC, como lo es el Uruguay.

⁴⁶ Como veremos dicha protección debería de existir en todos los países miembros de la OMC, puesto que los ADPIC –acuerdo obligatorio para todos los países miembros de la OMC- ordenan tal protección en su art. 39.

⁴⁷ Si hay exclusiones parciales, como lo es el dado por el art. 3 de la ley 17616 en cuanto a la protección de los modelos o creaciones que tengan un valor artístico en materia de vestuario, mobiliario, etc. Estos solo serán protegidos por el derecho de autor, según la mencionada ley, si es que ellos no gozan de la protección de la propiedad industrial

EEUU y su protección jurídica del software por medio del trade secret

En los EEUU el software puede ser protegido también por las normas del secreto comercial -Trade Secret-.⁴⁸

De esta manera, el instituto del Trade Secret de los EEUU protege la información que contenga las siguientes características: 1) que la información le signifique una ventaja económica a su titular, 2) Que el titular del secreto comercial cuide de manera adecuada dicha información de manera de mantenerla secreta, 3) Que dicha información sea, en los hechos, secreta, puesto que si no lo es, o deja de serlo, el secreto se extingue y con ello su protección bajo las normas de Trade Secret.⁴⁹

Es así que toda información secreta que le brinde una ventaja económica a su propietario y, a su vez, que su propietario la proteja con la finalidad de mantenerla secreta, podrá ser protegida por las normas del Trade Secret, en los EEUU. La información, sin pretender agotar este tema ni mucho menos, debe ser protegida según la naturaleza de la información y atendiendo las circunstancias de cada caso. Según cada circunstancia, se exigirá una mayor o menor protección para proteger a la información bajo las normas del Trade Secret. A modo de ejemplo, la protección debe ser protegida por medio de la firma de contratos de confidencialidad con aquellas personas a quienes se les suministre dicha información, y a su vez, transmitiendo la misma solamente a los que necesariamente la deban de conocer y a nadie más.

Es importante señalar que el trade secret otorga protección solamente cuando el tercero haya adquirido de manera ilícita dicha información. Por tanto, si un tercero por cualquier medio lícito llega a obtener dicha información, él podrá usarla libremente, y las normas del trade secret americano no otorgan ningún derecho para atacar este uso. Téngase presente que la ingeniería reversa o reverse engineering es lícita. Cualquier persona que adquiera un producto es libre de examinarlo de manera tal de obtener cualquier información relevante en cuanto al mismo.

La Ingeniería reversa o reverse engineering significa el estudio que se hace de un producto final y terminado, para saber cómo este fue producido y para obtener, a partir del producto final, cualquier información relevante al respecto. Pero téngase presente que la ingeniería reversa o reverse engineering es válida solamente cuando la persona que analiza el producto lo adquirió de manera lícita, como puede ser la compra del producto en el mercado abierto al público. Si el producto fue adquirido de manera ilícita, no se considerará dicha actividad lícita y podrá ser pasible de las acciones derivadas del Trade Secret.⁵⁰

⁴⁸ En los EEUU el Trade Secret constituye derecho estatal –state law-. Existe un modelo de protección del Trade Secret llamado Uniform Trade Secret Act. Este fue incorporado por 40 Estados americanos y en el estado de DC. Además, en los estados en donde no se incorporó el UTSA, se protegen de todas maneras a los secretos comerciales por medio del common law. Robert Merges y otros, *Intellectual Property in the New Technological Age* pág. 35 y 36

⁴⁹ Uniform Trade Secret Act, artículo 1, (4)

⁵⁰ Robert Merges y otros, *Intellectual Property in the New Technological Age*, pág. 74

Es posible obtener por ingeniería reversa del código objeto -este código es visible en todo software- una aproximación de lo que es el código fuente –cuando este código es escondido por el programador, puesto que muchas veces se opta por mostrar éste código-. Pero repetimos, cuando el producto se adquirió de manera lícita, la actividad de la ingeniería reversa sobre ese producto es lícita y esta actividad no se podrá atacar en base a las normas de trade secret.

Protección del código fuente y código objeto bajo el instituto del trade secret americano

El software, tanto en su código fuente como en su código objeto, puede ser protegido por las normas del Trade Secret bajo el ordenamiento jurídico americano.

Es posible ocultar el código fuente dentro de todo software. No sucede lo mismo con el código objeto. Este es imposible de ocultar y por ende estará en conocimiento de las personas que usen dicho software. Por tanto, un programador puede sacar al mercado su software ocultando el código fuente, pero inevitablemente el código objeto será conocido por las personas que adquieran el derecho al uso de dicho software.

Es importante en esta instancia mencionar que los programadores muy a menudo deciden esconder el código fuente del software, puesto que sin él, será muy difícil que terceras personas puedan arreglarlo, introducirle mejoras al software o hacer cualquier otro trabajo de adaptación.

En los casos en los que se comercializa el software haciendo visible su código objeto y ocultando el código fuente, éste último podrá ser protegido por el instituto del trade secret. El código objeto, en cambio, es muy difícil que se pueda proteger por estas normas, ya que, como recién dijimos, el código objeto será visible siempre, y este hecho le quita el requisito esencial de ser secreto y por tanto, en principio, el código objeto no podrá ser protegido por estas normas.

Sin embargo, en casos excepcionales, el código objeto puede ser también protegido por el Trade Secret. Y puede serlo, puesto que el requisito de ser secreto para constituir un secreto comercial no es absoluto. Es así que si el software es licenciado solo a un grupo limitado de personas y si conjuntamente con la licencia se firma siempre un contrato de confidencialidad, estableciendo por ejemplo que el licenciatario deberá mantener secreto el código objeto y estableciendo la obligación de no comunicar a terceras personas el contenido del código objeto, éste podrá ser protegido también por el Trade Secret.

Así lo entendió una corte americana en el caso *Trandes Corp v. Guy Atkinson Co.* 996 F2d 655 (1993), en el cual entendió que el código objeto permaneció en secreto pues el software que lo contenía había sido licenciado solamente a dos personas, las cuales firmaron un acuerdo de mantener en secreto el código objeto.

Trade secret en los ADPIC. Uruguay y su protección al software por medio del trade secret o secreto comercial

El Acuerdo sobre los ADPIC obliga a sus estados miembros a proteger la información no divulgada y protege a los secretos comerciales o Trade Secret en su art. 39. Es así que los ADDPIC incorporan el

instituto del Trade Secret. Los ADPIC fueron ratificados por el decreto-ley 16671 y por tanto constituyen derecho positivo vigente en el Uruguay.

La redacción del artículo 39 del acuerdo de los ADPIC recoge el instituto del Trade Secret e incorpora los requisitos ya vistos en relación al Trade Secret americano. Por ello nos animamos a decir que a nivel de derecho positivo, el Trade Secret de los EEUU no es distinto del secreto comercial en el Uruguay. Pero solo a nivel positivo, claro está, puesto que desconocemos al día de hoy cualquier acción en el Uruguay basada en el instituto del Trade Secret. De todas maneras, entendemos perfectamente posible esta acción, puesto que los ADPIC son derecho positivo en el Uruguay y con ello también lo es el Trade Secret del art. 39

Con suma facilidad se advertirá la gran similitud del Trade Secret americano con el Trade Secret o secreto comercial previsto en los ADPIC y que rige también en el Uruguay.

En este sentido el art. 39 de los ADPIC manifiesta que “los miembros protegerán la información no divulgada de conformidad con el párrafo 2”, y dicho párrafo, a continuación, dice que “Las personas físicas y jurídicas tendrán la posibilidad de impedir que la información que esté legítimamente bajo su control se divulgue a terceros o sea adquirida o utilizada por terceros sin su consentimiento de manera contraria a los usos comerciales honestos, en la medida que dicha información: a) sea secreta en el sentido de que no sea, como cuerpo o en la configuración y reunión precisas de sus componentes, generalmente conocida ni fácilmente accesible para personas introducidas en los círculos en que normalmente se utiliza el tipo de información; b) tenga un valor comercial por ser secreta; y c) haya sido objeto de medidas razonables, en las circunstancias, para mantenerla secreta, tomadas por la persona que legítimamente la controla”.

Entonces y según el derecho positivo vigente el software puede ser también protegido por el Trade Secret o secreto comercial si es que se cumplen con los requisitos a, b y c del párrafo 2, del art. 39 de los ADPIC. Estos requisitos son prácticamente los mismos que los requisitos del Trade Secret americano.⁵¹

La acción basada en un secreto comercial, de acuerdo a los ADPIC, permite accionar contra la persona no solo para resarcir el daño causado, sino también para solicitar el cese del uso de dicha información por parte del infractor y, lo que es más importante, a solicitar judicialmente que el infractor no difunda la divulgación de dicha información, o que deje de hacerlo, puesto que si la misma deja de ser secreta, esta protección se pierde para siempre. Es por ello que es fundamental accionar de manera rápida contra cualquier intento de divulgación ilícita de la información secreta e impedir que la misma deje de serlo. Una vez que la información deja de ser secreta, y aunque su divulgación haya sido hecha de manera ilícita, el secreto se pierde y con ello la especial protección que brinda el Trade Secret o secreto comercial.

⁵¹ Nos remitimos a la lectura del UTSA –Uniform Trade Secret Act- de los EEUU para apreciar la similitud del Trade Secret americano con los requisitos del secreto comercial establecido en los ADPIC, art. 39.

Conclusión

No existe ningún instrumento jurídico que proteja en forma perfecta al software. Sin embargo, existen institutos jurídicos que no fueron pensados para proteger a esta clase de trabajos u obras, pero que de todas maneras sirven para dar protección al software. Estos institutos en su protección al software tienen muchos inconvenientes. El derecho de autor, instituto que entendemos es el principal en proteger al software, tiene una duración excesiva (perdura toda la vida del autor y 50 años a partir de la muerte de éste⁵², término en el cual el software ya habrá pasado de moda) y su protección a los elementos no literales es prácticamente inexistente, por los motivos ya manifestados.

El software es pasible de protección por el derecho de autor, por las patentes y por el trade secret, y dichos instrumentos de protección no son excluyentes, sino que por el contrario se pueden acumular y usar en conjunto en vistas de obtener la mayor protección posible. Para esto, es fundamental conocer cada uno de estos institutos para cumplir, de ser posible, con los requisitos legales de cada uno de ellos.

Más concretamente, un software puede llegar a ser protegido por las normas del derecho de autor y por la patente; o por el derecho de autor y el trade Secret; pero nunca por la patente y el Trade Secret a la vez, puesto que los requisitos de uno y otro, como vimos, son diametralmente opuestos.

⁵² Artículo 7 de la ley 17616 y artículo 7 del Convenio de Berna