

Análisis económico de la protección al *software*: Hacia un nuevo modelo de protección

La preocupación respecto al modelo de protección jurídico ideal para el *software*¹ no es nueva², ni aislada internacionalmente³, por lo que recapitular de forma ordenada su devenir contribuye al revivido interés político⁴ por legislar sobre el modelo de protección actual, en particular en ausencia de literatura reciente⁵ que condense y analice los muy interesantes postulados políticos y económicos que se han planteado.

En el propósito de contextualizar tales postulados, detrás de la protección apropiada para el *software*, puede afirmarse que el discurso político ha evolucionado desde un momento totalmente incipiente de la industria, en el que de forma precipitada sendas comisiones gubernamental e internacional⁶ coincidieron en el confortable y residual voto por el Derecho de

¹ Sobre la noción de *software* (Entre muchos otros): (i) (Calle D'Alleman, 2012) Págs. 37 - 38, (ii) (Lipzyc, 2001) Pág. 104, (iii) (The software industry and developing countries, 2012) Pág. 2, (iv) (Gómez Segade, 2001) Pág. 840, (v) (Palacio Eastman, 1999) Pág. 201, (vi) (Nimmer & Krauthaus, 1987), (vii) (Uribe Arbeláez, 2005) Pág. 197, (viii) (Soler Matute, 2004) Pág. 23, (ix) Num. 1 Artículo 2.6.1.3.3. Decreto 1066 de 2015.

² Sobre el desarrollo del debate acerca de la protección del *software* (Entre muchos otros): (i) (Téllez Valdés, 1989) Págs. 20 – 21, (ii) (Nimmer & Krauthaus, 1987) Pág. 734 – 748, (iii) (Rengifo García, 2003) Pág. 206.

³ Sobre las distintas formas de protección del *software* ensayadas internacionalmente (Entre muchos otros): (i) (Leith, 2007) Pág. 160, (ii) (Gómez Segade, 2001) Pág. 848, (iii) (Palacio Eastman, 1999) Pág. 211.

⁴ (i) <http://www.grassley.senate.gov/news/news-releases/grassley-leahy-call-copyright-study> Accedido en octubre de 2015, (ii) (Bessen, The patent troll crisis is really a software patent crisis, 2013) Pág. 2, (iii) (Swinson, 1991) Págs. 177 – 178, (iv) Art. 4 Ley 1834/2017.

⁵ En general, el grueso de la literatura relacionada, principalmente internacional, se ubica entre 1989 y 1997, en ediciones y artículos que serán citados, y algunos reseñados, a lo largo de este escrito. De reciente aparición nacional: (i) *El software o programa de ordenador y el concepto de invención patentable*, escrito por Salas Pasuy, Brenda. Capítulo 11 de la edición *Derecho de patentes*, publicado por la Universidad Externado de Colombia en 2016. Este capítulo cuenta con una buena selección de precedentes jurisprudenciales que configuran una aproximación a las “patentes implementadas por programas de ordenador”, (ii) *Protección jurídica del software. Crítica a su actual regulación y redefinición de su naturaleza frente al derecho*, tesis doctoral escrita por Calle D’Alleman, Sol Beatriz y publicada por Editorial Ibáñez en 2012. Con este trabajo coincidimos en la necesidad de debatir y postular nuevos modelos de protección para el *software*, si bien son diferentes nuestras aproximaciones teóricas a la materia, objetivos del trabajo y soluciones propuestas.

⁶ (i) La establecida *Commission of New Technological Uses of Copyrighted Works* (“CONTU” por sus siglas en inglés) en 1974, fue el encargo del gobierno de los Estados Unidos de América para estudiar la mejor vía jurídica de protección para el *software*. Sobre sus conclusiones: 1.- (Menell, Tailoring legal protection for computer software, 1987), 2.- (Samuelson, Davis, Mitchell D., & Reichman, 1994), genial artículo con el cual coincidimos en mucho; (ii) Por su parte, la OMPI también coordinó una comisión de expertos en el año 1971, los cuales redactaron un tipo de norma modelo para la protección del *software*, que tuvo feliz resultado en 1977; (iii) La versión japonesa de estas iniciativas data de los 70’s, cuando la Agencia de Asuntos Culturales del Japón (La cual cumple, entre otras funciones, la de llevar el registro público de obras protegidas por el Derecho de Autor) adelantó un análisis del sistema del Derecho de Autor aplicado al *software*, en concomitancia con la expedida Ley 48 de 1970; y el informe presentado por el

Autor como modelo de protección⁷, hacia un llamado actual por fortalecer la protección de la propiedad intelectual respecto del *software* basado en su funcionalidad⁸, y no en el ahora criticado argumento de equivalencia literaria del código fuente⁹.

Esta intuitiva evolución de pensamiento, pues en su momento ningún estudio empírico existía¹⁰ para motivar una respuesta política a la “necesidad”¹¹ de ubicar el *software* como obra literaria a fin de reglamentar su protección, hoy se funda en un reconocimiento del déficit de protección para estos creadores¹², debido a que el mérito atribuido al *software* que han codificado nos les beneficia completamente, tomando en consideración la litigiosidad que exhibe una fuga parcial de beneficios hacia terceros¹³.

Ministerio de Industria y Comercio del Japón, de 1972, en el cual se explora la protección del *software* bajo un modelo *sui generis*. Comentando estos reportes: (Teruo, 1992).

⁷ (i) (Berkemeyer, 2009) Pág. 127, (ii) En cuanto a la facilidad derivada del aprovechamiento de una infraestructura de Tratados internacionales: (Preinfalk Lavagni, 2010) Pág. 155, (iii) En cuanto al facilitista trabajo de la CONTU: (Menell, Tailoring legal protection for computer software, 1987) Pág. 1330, (iv) (Carranza Torres, 2008) Pág. 220: “[A] la hora de dar solución al problema se recurrió a lo que se tenía más a mano, el *copyright*” y señalando sus consecuencias, más adelante añade (Pág 221): “*El ADPIC dispuso también que todos los países miembros de la OMC debía adaptar sus legislaciones internas en determinados plazo, so pena de afrontar sanciones comerciales. **La globalización del accidente histórico era ya una realidad***” (Negrilla y subraya fuera del texto original).

⁸ (i) (Leith, 2007) Pág. 159, (ii) (Calle D'Alleman, 2012) Pág. 228.

⁹ (i) (Carranza Torres, 2008) Págs. 205 y 226, (ii) (Palacio Eastman, 1999) Pág. 221, (iii) (Villalba & Lipzyc, 2001) Pág. 14, (iv) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2319.

¹⁰ (i) (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 33, (ii) Además: (Carranza Torres, 2008) Pág. 220: “*Ningún informático, ningún representante de la industria, ningún avezado usuario de software formó parte de la CONTU*”.

¹¹ (i) (Model provisions on the protection of computer software, 1978) Pár. 5, (ii) (Measures to enhance international cooperation in the field of legal protection of computer software, 1979) Pág. 2: “[W]ith respect to the desirability of legal protection of computer software have gained additional importance. Nevertheless, to the knowledge of the International Bureau of WIPO, the existing possibilities of protection under national laws do not seem to have changed as compared with the time when the Model Provisions were prepared; although the question seems to be under active consideration by the governments of several countries, the adoption of legislation on the protection of computer software does not seem to have materialized so far. One of the reason for this may be precisely the current dynamism of the software sector, which as yet has left no time for the stabilization of legal concepts”.

¹² Incluso para defensores del régimen actual: (Dam, 1995) Pág. 335.

¹³ (i) Una estadística sobre la litigiosidad en aumento por asuntos relacionados con patentes, entre ellas las implementadas por programas de ordenador en: <http://www.managingip.com/Article/3518856/Managing-Patents-Archive/US-patent-litigation-up-15-in2015-with-NPEs-increasing-share.html>, (ii) (Bessen, The patent troll crisis is really a software patent crisis, 2013) Pág. 6: “*Research shows that a software patent is four times as likely as a chemical patent to be involved in litigation; a software patent on a method of doing business is thirteen times more likely to be litigated. Nor are these high rates of litigation a result of the newness of software patents*”.

Adicionalmente, toda la atención académica que han despertado las intervenciones gubernamentales en trámites de competencia¹⁴ y el ruido mediático que generaron los casos judiciales en los cuales derechos sobre un cúmulo de “*invenciones implementadas por ordenador*”¹⁵ (“*Patent trolls*”)¹⁶ han sido fundamento, ciertamente merecen un debate político que ordene y cierre el asunto del espacio para la propiedad intelectual respecto del *software*.

Por aparte, transitando un camino pavimentado en el Derecho de Autor, obviamente en clave utilitarista y cuando sí había industria del *software*, el discurso económico que acompaña la selección para la protección del *software* entre un modelo basado en el Derechos de Autor, el Sistema de Patentes, en la Protección a Secretos Empresariales¹⁷, o uno *sui generis*; pasa revista

¹⁴ (i) IPPro The Internet News. 4/Dec/2015: “*Apple finally gets half a billion in damages*”, (ii) Bloomberg Business. 25/Sep/2015: “*Google said to be under U.S. Antitrust scrutiny over Android*”, (iii) The Washington Post. 2/Jan/2015: “*The patent troll crisis is really a software patent crisis*”, (iv) Bloomberg Business. 7/Feb/2014: “*Samsung antitrust probe over patent use closed by U.S.*”, (v) Bloomberg Legal BNA. 25/Oct/2011: “*Apple antitrust claims against Samsung dismissed*”, (vi) The Wall Street Journal. 10/Sep/2001: “*Microsoft drafts settlement proposal hoping to resolve antitrust lawsuit*”.

¹⁵ La expresión es una adaptación mejorada de anglosajonas tales como: “*computer-related inventions*” o “*software-based inventions*” que la doctrina de los Estados Unidos de América empezó a emplear a partir del caso (Diamond v Diehr, 1981). A través de la expresión “*invenciones implementadas por computador*” (IIC) o “*invenciones implementadas en ordenador*” queremos aprovechar la terminología empleada por la Oficina Europea de Patentes (“OEP”), por la cual se da noción del tipo de propiedad industrial concedida a este tipo de nuevas creaciones. Según la OEP: “*Aunque el término ‘patente de software’ es de uso frecuente, es equívoco. En la Oficina Europea de Patentes (OEP) más que a patentes de programas informáticos nos referimos a invenciones implementadas en ordenador. Una invención implementada en ordenador es aquella que implica el uso de un ordenador, una red informática u otro aparato programable en el que una o más de sus funciones se llevan a cabo total o parcialmente gracias a un programa de ordenador*” ([¿Patentar software? Normas y usos en la Oficina Europea de Patentes](#)). Volveremos sobre el alcance de esta descripción en el Capítulo II.

¹⁶ (i) “*jungla de patentes*” (Uribe Arbeláez, 2005) Pág. 202, (ii) (Bessen, The patent troll crisis is really a software patent crisis, 2013) Pág. 2: “[P]olicymakers haven’t given enough attention to the fact that the patent crisis is mostly about patents on software. Understanding that fact is essential to fixing what’s wrong with the patent system”... “*why are there so many lawsuits over software patents? The report states that ‘many recent patent infringement lawsuits are related to the prevalence of low quality patents; that is, patents with unclear property rights, overly broad claims, or both. Although there is some inherent uncertainty associated with all patent claims, several of the stakeholders with this opinion noted that claims in software related patents are often overly broad, unclear or both’*”, (iii) http://www.ipprotheinternet.com/ipprotheinternet/IPPro_issue_008.pdf: “USPTO director calls for different approaches to defeat patent trolls” Pág. 5

¹⁷ (i) (Smith & Mann, 2004), (ii) Tratando extensamente estas tres opciones: (West, 1995), (iii) En igual sentido: (Calle D’Alleman, 2012) Págs. 107 – 161.

al costo de restringir en parte la innovación futura beneficiando la actual¹⁸, e igualmente por la evaluación de externalidades como la *estandarización* de aplicaciones y su graves consecuencias contra la libre competencia¹⁹, todo lo cual favorece el mantenimiento del *statu quo*, predecible y en relativo equilibrio entre costos y beneficios²⁰, a favor de un sistema de protección por el Derecho de Autor.

Por lo anterior es claro que las presentaciones de modelos de protección jurídicos apropiados para el *software*, desde las ópticas política y económica, han tenido puntos de partida y ritmos de evolución opuestos, pues en la actualidad el discurso político es el que despierta un debate que el análisis económico dejó anestesiado en los 90's; pero no por ello se debe desaprovechar el *momentum*, si gracias a ello se puede formular una mejor normativa²¹, aprovechando las importantes conclusiones y propuestas que se recogen en abundancia de la literatura foránea²² y la jurisprudencia mundial – inexistentes en los 70's cuando se fijó el modelo de protección del *software*-.

¹⁸ (i) (Pérez Gómez-Tétré, 2006-2007) Pág. 18, (ii) (Dam, 1995) Pág. 363., (iii) (Menell, Tailoring legal protection for computer software, 1987) Pag. 1340: “*From the perspective of the public interest, the optimal system for protecting intellectual work equates the marginal benefit of enhancing the scope of intellectual work protection with the marginal cost of greater protection. That is, it equates the benefits of the availability of more and better products with research costs, losses due to monopolistic exploitation, administrative costs, and inhibiting effects on inventive activity*”.

¹⁹ En este sentido: (i) (Menell, An analysis of the scope of Copyright protection for application programs, 1989), (ii) (Klein, 1993).

²⁰ (i) Análisis que se explica en el excelente artículo: *Some economic consideration in the intellectual property protection of software* (Dam, 1995). En este artículo se resalta la posible falta de claridad en los contornos del derecho concedido al *software* por un hipotético sistema *sui generis*, (ii) En decidido voto por el Derecho de Autor, como modelo de protección para el *software* de cuantos se han estudiado: (Dukrok & Hwang, 2010), (iii) En igual sentido: (Swinson, 1991).

²¹ (i) Vaticinando lo que ocurriría: (Nimmer & Krauthaus, 1987) Págs 746 – 747. En este artículo se indica que primero hay un estado de incertidumbre sobre el tipo de protección actual para el *software*, tratando de acomodarlo a alguna de las categorías existentes, luego sigue una etapa de litigios judiciales, en la cual generalmente queda el sinsabor del déficit de protección, y posteriormente sigue una etapa de legislación, que trae consigo la inevitable crítica y expectativa sobre la consecución de los resultados esperados, (ii) En sentido similar: (Smith & Mann, 2004) Pág. 242.

²² (i) (Leith, 2007) Pág. 160, (ii) (Salas Pasuy, 2016) Pág. 437.

Ahora bien, el punto común de ambos discursos que se analizará a lo largo del texto es el relacionado con las normas jurídicas para reservar al creador de un *software* los frutos de su explotación económica, pero no aquellas que buscan disuadir la reproducción servil (“*Piracy*”) del *software*, sino las que deberían prevenir la imitación mediante reescritura de los conceptos esenciales del *software* innovador (“*Cloning*”)²³.

Lo anterior, pues la protección contra el *piracy* ha sido muy analizada²⁴, en especial desde la adopción del Acuerdo sobre los Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio (“ADPIC”)²⁵ y el Tratado de la OMPI sobre Derecho de Autor (“WCT” por sus siglas en inglés)²⁶, a partir del cual se han promulgado en Colombia y el mundo un número suficiente de leyes relacionadas con la penalización de la copia ilegal de material protegido por el Derecho de Autor, incluido el *software*²⁷.

²³ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2381, (ii) (Dam, 1995) Pág. 328, (iii) (Rengifo García, 2003) Págs. 215 – 216.

²⁴ Un documento que pasa revista al debate legislativo y jurisprudencial en Colombia, respecto del *piracy*, puede leerse en: www.wipo.int/edocs/mdocs/es/.../OMPI_PI_JU_LAC_04_25.doc

²⁵ (i) Ley 170 de 1994, (ii) El artículo 61 de ADPIC establece la obligación a los Miembros de legislar sobre la piratería de marcas y obras.

²⁶ Ley 565 de 2000.

²⁷ (i) De hecho, poco antes de la promulgación de la Ley 170 de 1994, se promulgó la Ley 44 de 1993 (modificatoria de la Ley 23 de 1982, en cuanto tipificó conductas penales relacionadas con la infracción a derechos de autor). Con todo, a partir de la Ley 565 de 2000, siguieron las leyes: 1.- 599 de 2000 (Que promulga el Código Penal actual, en cuyos artículos 270 y siguientes se tipifican delitos contra los derechos reconocidos sobre la propiedad intelectual), 2.- Ley 603 de 2000 (Que reglamenta el contenido de los informes de los administradores de las sociedades, y sanciona penalmente a quienes falten a la verdad sobre el estado de la propiedad intelectual explotada en sus empresas), 3.- 813 de 2003, 4.- 906 de 2004, 5.- 1032 de 2006, y 6.- 1520 de 2012 (Que aumentan las penas); entre otras. (ii) Una crítica sobre la efectividad de las normas penales como vehículos de protección para el *software* se encuentra en (Calle D'Alleman, 2012) Pág. 153, (iii) Tratando un tema relacionado, cuando se importan criterios de protección incoherentes culturalmente: (Márquez Escobar, 2004), (iv) Sobre el problema de la gratuidad en los contenidos y si es o no un problema moral la piratería: (Gozzer & Vicente, 2013) Págs. 37 – 38, (v) Sobre los efectos de la Digital Millenium Copyright Act: (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1640.

En cambio, respecto del *cloning*, aunque el problema se ha explicado suficientemente, no se ha legislado sobre él; por esto, se hará una explicación del problema que se presenta cuando en el empeño de evitar el *cloning*, el creador de un *software* acude a la oferta actual de normas jurídicas²⁸ en el Derecho de Autor (Capítulo I), el Sistema de Patentes (Capítulo II) y la Protección a Secretos Empresariales (Capítulo III), y así exponer la inapropiada adecuación de estas normas para el caso del *software*.

Adicionalmente, indicaremos las razones que en nuestro concepto bloquean el surgimiento de una protección *sui generis*²⁹ para el *software* y concluiremos con un marco general para la protección del *software* (Capítulo IV), que se acompaña de un proyecto normativo (Capítulo V), el cual soluciona el problema del *cloning*, y otros asociados a la convulsiva³⁰ e insuficiente protección del *software* a través del Derecho de Autor.

Naturalmente, estos últimos apartados están impregnados de juicios de valor sobre la conveniencia de un modelo de protección diferente, justificados en un sincretismo contemporáneo, a fin de redondear la misión contextualizadora del presente escrito y en licencia académica que caracteriza al autor.

²⁸ (i) (Smith & Mann, 2004) Pág. 242, (ii) (Calle D'Alleman, 2012) Pág. 40

²⁹ Nos referimos a una reglamentación “autónoma y diferente” al Derecho de Autor, el Sistema de Patentes y la Protección a Secretos Empresariales, pues está claro que toda la propiedad intelectual es especial (Art. 671 Código Civil). Al respecto: (Baylos Corroza, 2009) Pág. 521 – 525 y 633.

³⁰ (Hong & Cheng-ssu, 1999) Pág. 34.

**Capítulo I. Mejor malo conocido, que bueno por conocer: Pros y contras de la
protección del *software* vía el Derecho de Autor**

En el marco teórico y normativo del Derecho de Autor³¹ se sostiene un próspero crecimiento de la industria del *software*, en Colombia de dos dígitos³², lo cual sugiere funcionalidad y, en consecuencia, innecesaria la modificación del modelo jurídico de protección actual³³.

Con todo, estudios económicos y jurídicos³⁴ mencionan una intervención legislativa en Brasil³⁵, India³⁶ y China³⁷, que han catapultado sus industrias de *software* a niveles de primer orden en innovación y ventas a escala mundial³⁸, con una característica común: No es la tradicional protección por el Derecho de Autor la promulgada, sino una *lex specialis*³⁹ que

³¹ (i) El artículo 15-e) de la Decisión 486 de la Comunidad Andina de Naciones, excluye de la materia patentable al *software*, (ii) El artículo 23 de la Decisión 351 de la Comunidad Andina de Naciones, específicamente comprende al *software* dentro de las obras protegidas por el Derecho de Autor, (iii) El artículo 10 – 1 de ADPIC expresamente señala que la protección del *software* se registrará por las normas contenidas en el Convenio de Berna.

³² (i) Fuente: (Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia, 2014). También, pueden consultarse: (ii) (Caracterización del sector de teleinformática, software y TI en Colombia. Año 2015. , 2016), (iii) Estudio de la industria del *software* internacional, nacional y departamental. Intersoftware. Accesible en: http://64.76.85.60/tda2/hermesoft/portal/home_1/rec/arc_3620.pdf, (iv) “Radiografía de la industria colombiana del 'software'”. Edición de El Tiempo, enero 28 de 2014, (v) <http://www.portafolio.co/especiales/innovacion-colombia-2015-startups-y-apps-crecimiento/empresa-colombiana-silicon-valley-s>. Noviembre 8 de 2015.

³³ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2368.

³⁴ (The software industry and developing countries, 2012) Págs. 2 y 100.

³⁵ Ley de protección de programas de computador No. 9609 de 1998, (i) Según (The software industry and developing countries, 2012), la industria del *software* en Brasil está orientada a satisfacer las necesidades locales (Pág. 51), (ii) Concordando con esta conclusión y presentando una dimensión sobre la expansión del mercado del *software* en el vecino país: (Junqueira Botelho, Stefanuto, & Veloso, 2007) Pág. 99.

³⁶ Ley de tecnologías de la información de 2000, (i) Este es un país cuya industria del *software* está orientada hacia el mercado externo, sin descuidar la demanda interna a través de políticas públicas: (The software industry and developing countries, 2012) Pág. 38, (ii) (Athreye, 2007) Pág. 7: “[T]he Indian software industry accounted for over 28 percent of India's GDP growth between 2000 and 2002”.

³⁷ Ley de medidas de registro de *software* de 2006. Así como en Brasil, la industria del *software* está orientada a satisfacer las necesidades locales.

³⁸ Otros países con ventas importantes a nivel mundial son: Filipinas, Irlanda, Israel, Uruguay y Sri Lanka. Ahora bien, dependiendo de la dotación de población calificada y anglo-parlante, la política pública de fomento a la industria del *software* puede obtener resultados orientados hacia el mercado interno o externo. Al respecto: (The software industry and developing countries, 2012) Cap. 2. En general, puede afirmarse que el estímulo para esta industria en Brasil y la India provino de una primera etapa de protección al sector y demanda proveniente de compras públicas, seguida de una etapa posterior de libre competencia. Con todo, políticas paralelas de exenciones tributarias también forjaron parte del crecimiento en el mercado del *software*.

³⁹ (Ascarelli, 1970) Págs. 306 – 307.

precisa ciertos aspectos del objeto de protección, promoviendo una demanda local o una oferta orientada al mercado global⁴⁰, en combinación con una política de incentivos tributarios.

Si bien el éxito económico de la industria de *software* en estos Estados tiene origen en causas pretéritas y endógenas⁴¹, todos combinaron normas jurídicas que especializaron la ley de Derecho de Autor vigente y concedieron beneficios tributarios a los creadores de estos bienes, lo cual parece ser la clave para el estímulo y desarrollo de una industria que toca transversalmente a toda la economía, en donde quiera que se estimule⁴², y que puede complementar la sonada ventaja competitiva colombiana en los servicios *outsourcing*⁴³.

⁴⁰ Los resultados económicos evidencian que las políticas públicas aplicadas en los mercados Brasileiro e Indio son ejemplos de instrumentos para incentivar la producción de *software* dirigido al mercado interno, como es el caso Brasileiro, o externo, como es el caso Indio. En este sentido: (i) (The software industry and developing countries, 2012), (ii) *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland, and Israel*. Arora, Ashish y Gambardella, Alfonso. Oxford University Press. New York 2005. (iii) Lo que puede concluirse de los anteriores estudios es que la intervención legislativa en la industria sirve para potencializarla en el doble propósito de abastecer el mercado interno y ofertar en el externo. Ahora, es válido afirmar que la industria del *software* en Colombia crece a través de una demanda interna similar a la ocurrida en India, y presumiblemente la orientación exportadora puede ser el camino que continúe, probablemente al mercado Latinoamericano, pues en conjunto supera el peso individual de Estados Unidos de América: (Caracterización del sector de teledinformática, *software* y TI en Colombia. Año 2015. , 2016).

⁴¹ (i) (Junqueira Botelho, Stefanuto, & Veloso, 2007) Págs. 102 – 104, (ii) (Adler, 1986) Pág. 15, (iii) Vale decir que en Brasil e India los gobiernos crearon empresas estatales, con la misión de crear *software* y *hardware*: (Adler, 1986) Pág. 19.

⁴² Tratando el tema del impacto de la industria del *software* en diferentes economías, lo mismo que las causas de surgimiento: (i) (Estudio de la caracterización de productos y servicios del sector de *software* y tecnologías relacionadas de Colombia, 2014), (ii) (The software industry and developing countries, 2012), (iii) (Lema, Quadros, & Schmitz, 2012), (iv) *From Underdogs to Tigers: The Rise and Growth of the Software Industry in Brazil, China, India, Ireland, and Israel*. Arora, Ashish y Gambardella, Alfonso. Oxford University Press. New York 2005.

⁴³ (i) El diario Portafolio, en su edición de marzo 26 de 2012, dentro de una sección especial de negocios, tituló: “Colombia es la joya de la tercerización en la región”. El artículo es accesible en: <http://www.portafolio.co/negocios/colombia-es-la-joya-la-tercerizacion-la-region>, (ii) Sobre la focalización en ciertos sectores de la economía colombiana: (Conpes 3582, 2009) Pág. 29: “La focalización o priorización consiste en el direccionamiento de un conjunto de acciones y recursos hacia un determinado sector productivo o área del conocimiento determinado. Esta definición no implica que se busque reemplazar al mercado en sus funciones de asignar recursos a las actividades más productivas, sino en vez de ello, alcanzar retornos sociales que no se alcanzarán a partir de la acción de los agentes privados”. Según este mismo estudio, uno de los sectores en los cuales se deben “focalizar” esfuerzos es el del *software* (Págs. 31 y 51), (iii) Sobre “focalizar” incentivos para aumentar acceso y velocidad a Internet, como política pública para acortar la brecha de ingresos: (Aproximación al impacto de las TIC en la desigualdad de ingresos en Colombia, 2018) Pág. 26.

Ahora, en términos generales, el estímulo tributario ciertamente incentiva la creación de *software*⁴⁴, pero el presente trabajo se concentrará en responder al fenómeno del *cloning*, con o sin incentivos tributarios de por medio, con o sin incentivo al consumo. En otras palabras, el trabajo se ocupa del efecto en la creación del *software* debido a que su actual sistema de protección permite estos *free riders*⁴⁵, sin desconocer que una política pública responsable para esta industria exige el análisis y tratamiento de los aspectos tributarios aplicables⁴⁶, lo mismo que la formulación de remedios a las externalidades en el consumo y en el mercado laboral asociado⁴⁷.

⁴⁴ (i) (Junqueira Botelho, Stefanuto, & Veloso, 2007) Págs. 104 – 105, (ii) En Colombia no han sido satisfactorias las deducciones tributarias por inversiones en ciencia, tecnología e innovación: (Conpes 3834, 2015) Págs. 6- 7: “*Dentro del análisis presentado en este documento, se destaca que el instrumento ha sido usado en un bajo porcentaje, que este se ha concentrado especialmente en empresas contribuyentes de tamaño grande y con renta líquida, y que han existido dificultades en las condiciones de acceso al instrumento. Por lo tanto, este documento define lineamientos alrededor de dos objetivos de política: i) incentivar la inversión privada en actividades de CTI a través de las deducciones tributarias; ii) mejorar las condiciones y alternativas de acceso a las deducciones tributarias. Para lograrlo se recomienda, entre otras acciones, ampliar el ámbito de aplicación del instrumento para que incluya proyectos de innovación como lo establece la Ley 1739 de 2014, establecer un esquema de calificación automática para empresas altamente innovadoras, y habilitar a empresas en etapa temprana y Pymes que, aunque no generan renta líquida hoy en día, podrán deducir estas inversiones en innovación en el futuro cuando deban realizar contribuciones a través del impuesto de renta*”. El mismo documento señala lo siguiente en la página 16: “[C]ada peso de inversión pública mediante deducciones tributarias apalancaría cuatro pesos del sector privado”. (iii) (Fedesarrollo, 2005) Pág. 64: “*Dado que las cantidades de los montos de incentivos otorgados son relativamente pequeñas, y que es algo tan reciente, es difícil evaluar las ventajas y desventajas de este incentivo. Sin embargo, el diseño del instrumento utilizado en Colombia presenta un elemento negativo en lo referente a la duración de la exención tributaria que según la ley es de 10 años a partir de la entrada en vigencia de la ley (1 de enero de 2003). Esto implica que el incentivo tiene un mayor efecto sobre las empresas que estaban en proceso de desarrollo del software antes de promulgada la ley y comenzaron a producirlo justo a inicios de 2003, ya que pueden gozar del alivio fiscal por más tiempo. Por el contrario, una firma que se planea desarrollar un nuevo software para comenzar a producirlo en el 2010 sólo puede aplicar el incentivo a partir de este año lo que implica que disfrutará del beneficio por menos tiempo*”. (iv) Con una posición escéptica en torno a la conexión entre el incentivo tributario y la mayor cantidad y calidad de productos de propiedad intelectual, y en particular reseñando el abolido incentivo al *software* de origen Colombiano: (Buitrago, Patent boxes and the erosion of trust in trade and in governance, 2019) Págs. 18 – 19.

⁴⁵ Puede leerse una explicación del efecto económico de esta externalidad del mercado en: (i) (Mas-Collel, Whinston, & Green, 1995) Págs. 359 – 365, (ii) (Posner, Antitrust law, 2001) Pág. 246.

⁴⁶ (Buitrago Díaz, 2012) Págs. 475 – 476: “[C]omparado con los incentivos que otros países ofrecen, y con los requisitos exigidos para obtenerlos, los colombianos no solo se quedan cortos, sino que parecen muy complicados, dados los requisitos, calificaciones y aprobaciones necesarios para obtenerlos...”.

⁴⁷ Al respecto de estas externalidades, los estudios sectoriales: (i) (Allen, Armstrong, Reid, & Riemenschneider, 2009), (ii) (Caracterización del sector de teleinformática, software y TI en Colombia. Año 2015. , 2016), (iii) (Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia, 2014), (iv) (Fedesarrollo, 2005), (v) (Heshusius Rodríguez, 2009).

En consecuencia, el análisis económico se condensa en las externalidades del Derecho de Autor como modelo de protección para el *software*, concretamente en lo que se refiere a la permisión de trabajos independientes igualmente protegibles⁴⁸. Esto ocurre porque el Derecho de Autor solo protege la particular expresión de los algoritmos⁴⁹ que componen el *software* y no el concepto esencial en ellos (*idea en sí*⁵⁰), aunado a que por idénticas que sean las funcionalidades entre algoritmos, si son escritos en lenguajes de programación diferentes serán considerados obras independientes⁵¹; lo cual, demandaría el trabajo económicamente imposible de desarrollar el *software* en todos los lenguajes de programación apropiados, si evitar el *cloning* se intenta.

Adicionalmente, se critica el plazo de protección del *software* mediante el Derecho de Autor, pues se impone un lesivo precio para el mercado teniendo en cuenta que su lapso, mínimo

⁴⁸ (i) (Nimmer & Krauthaus, 1987) Pág. 750, (ii) (Swinson, 1991) Pág. 176.

⁴⁹ (i) (Swinson, 1991) Págs. 150, 154 y 192, (ii) (Teruo, 1992) Pág. 107, (iii) (Paterson, 2013) Pág. 188. Sobre el trabajo intelectual implícito en la elaboración de algoritmos, y por ende por qué se trata como una creación intelectual protegible: (iv) (Calle D'Alleman, 2012) Página 242, (v) (Córdoba Marentes, 2006) Pág. 143, (vi) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2351 y 2383 - 2384, (vii) Describiendo erróneamente a los algoritmos como “el abecedario de los programas”, a efectos de restarle aptitud como propiedad intelectual protegible: (Uribe Arbeláez, 2005) Pág. 221.

⁵⁰ Sobre la difícil tarea de separar la “*idea en sí*” de lo que es verdadera expresión protegida por el Derecho de Autor: (i) (Rengifo García, 2003) Págs. 216 – 219, (ii) (Peña Valenzuela, 2013) Pag. 43, (iii) (Gómez Segade, 2001) Pág. 858, (iv) (Ginsburg, 1994) Pág. 2569, (v) (Palacio Eastman, 1999) Pág. 214, (vi) (Guerrero Gaitán, 2006-2007) Pág. 128, (vi) (Preinfalk Lavagni, 2010) Pág. 129.

⁵¹ (i) (Swinson, 1991) Pág. 148 – 149. Más adelante, proponiendo que no se permita lo anterior, sobre la base de una interpretación adecuada del Derecho de Autor: Pág. 195, (ii) Por ejemplo, en el ámbito nacional, el laudo arbitral de agosto de 2006 dentro del proceso de (As Colombia Ltda. v. Informática y Gestión S.A, 2006) Pág. 27: “*El software es una obra utilitaria y el hecho de que dos software compartan o utilicen o lleguen a funcionalidades idénticas o similares, no significa que inexorablemente uno sea copia o reproducción del otro. En otras palabras: no necesariamente la identidad de funcionalidades significa copia o reproducción dado que se puede arribar a ellas por caminos o esfuerzos independientes tal como aconteció en el presente caso*”, (iii) (Carranza Torres, 2008) Pág. 207. De esta última obra llamamos la atención sus comentarios sobre el caso (Whelan v. Jaslow, 1986) Págs. 238 a 243, virtualmente idéntico al caso debatido en el proceso arbitral antes mencionado, pero con un resultado diferente pues en este el *software* del demandante sí recibió protección a su estructura, secuencia y organización, aunque el *software* del demandado estaba escrito en un lenguaje de programación diferente. (iv) Aparte: Una buena colección de jurisprudencia sobre *software*, en la cual se puede encontrar este laudo arbitral, es posible consultar en: <http://www.cerlalc.org/>.

de 50 años *post mortem auctoris*⁵², no es un término adecuado a la naturaleza del bien entre manos⁵³.

Pese a lo anterior, la protección del *software* mediante el Derecho de Autor (Sistema del *civil law*) o *copyright* (Sistema del *common law*) es la protección jurídica resuelta por los Tratados internacionales más importantes de propiedad intelectual⁵⁴, por número de Estados partes⁵⁵; lo cual, sin duda alguna, es mérito suficiente para exigir especial justificación en recomendar un sistema de protección diferente, o uno *sui generis*.

Para profundizar en los aspectos relevantes de la protección jurídica del *software* vía el Derecho de Autor, el presente capítulo expone incentivos (1) y externalidades (2) a la producción de *software* que provoca tal régimen jurídico, señalando los patrones encontrados en doctrina y jurisprudencia.

⁵² (i) Artículo 7-1 Convenio de Berna para la protección de las Obras Literarias y Artísticas (Ley 33 de 1987). Sobre el “mínimo convencional” (Baylos Corroza, 2009) Pág. 674 y 678, (ii) En Colombia, los artículos 11 y 27 de la Ley 23 de 1982, señalan un término de 80 años *post mortem auctoris*, si el titular es una persona natural, y de 70 si es una persona jurídica (Modificación introducida por el artículo 4 de la Ley 1915/18), (iii) La ley coreana de protección al *software* (“CPPL” por sus siglas en inglés), consagra una protección de 50 años: (Byungkwon, 1989) Pág. 181.

⁵³ (i) (Arenas Correa, 2013) Pag. 167: “La protección de cincuenta años – ochenta en Colombia, cuando se trata de personas naturales - conferida a los programas de computador bajo el amparo del sistema de protección de los derechos de autor es excesivamente larga para la realidad del mercado en cuestión y se equipara, para efectos del análisis práctico del mercado informático, a un monopolio al infinito, pues rebasa en mucho la obsolescencia técnica de los programas, el periodo de protección que les confiere mediante Adpic (art. 12) y la Ley 23 de 1982”, (ii) Igual reparo podría hacerse al tiempo de protección vía el Sistema de Patentes, pues las patentes de invención, y procedimientos, son protegidas por 20 años. Una revisión sobre la relación costo beneficio de los plazos de protección de la propiedad intelectual, pueden leerse en: (Lemley, Ex Ante versus Ex Post Justifications for Intellectual Property, 2004), (iii) En igual sentido: (Menell, Tailoring legal protection for computer software, 1987), (iv) (Carraza Torres, 2004) Pág. 169: “¿pondera suficientemente la ‘libertad’ [Refiriéndose al actual régimen de Derechos de Autor]? Muy probablemente, la ‘libertad’ no esté valorada como debiera estarlo. Un mínimo de cincuenta años de monopolio para algo como el software, que evoluciona vertiginosamente, es una exageración. Pero eso es lo que debemos negociar. No la exclusividad”. (v) En sentido contrario: (Teruo, 1992) Pag. 97: “There is no affirmative ground to shorten the duration of copyright for computer programs, distinguishing them from other kind of Works of a scientific nature”.

⁵⁴ (i) WCT (Ley 565/00), (ii) ADPIC (Ley 170/94), (iii) En el mismo sentido: Directiva 91/250/CEE (Modificada por la Directiva 09/24/CEE), (iv) Decisión 351 de la Comunidad Andina de Naciones.

⁵⁵ 176 países (89% del planeta).

1. ¿Qué incentiva la protección del *software* a través del Derecho de Autor?

El tratamiento de obra literaria que se ha dado al *software* ciertamente trajo consigo al inicio, y perdura hasta ahora, el importantísimo incentivo a la producción intelectual que proviene de la confianza en un sistema de protección jurídico cuya práctica se conoce (El del Derecho de Autor), y del cual el objeto y sujeto involucrados gozarían *mutatis mutandis*⁵⁶.

Particularmente, los creadores de *software* cuentan hoy con un marco legal que da bienvenida a producciones intelectuales sin revisión del mérito atribuido a la creación, siempre que sea original⁵⁷, sin necesidad de registro constitutivo de derechos⁵⁸, aunque haya depósitos legales con fines de oponibilidad y transmisión del conocimiento⁵⁹, y con una larga lista de prerrogativas (derechos morales y patrimoniales) que perduran más allá de la vida del creador⁶⁰.

Pero si bien lo anterior se da por sentado en la actualidad, nada de esto estuvo disponible en un principio, pues diferenciar el *hardware* del *software* era inconcebible⁶¹, y en su momento no se tenía claridad sobre el alcance de la protección para la descripción, material auxiliar, código fuente u objeto⁶² del *software*. Sin duda, todo lo anterior favorece la inversión de recursos en la elaboración de *software*, como también lo que a continuación se describe:

⁵⁶ (i) (Nimmer & Krauthaus, 1987) Pág. 745, (ii) (Aguirre Soriano, 2014) Pág. 91.

⁵⁷ (i) (Calle D'Alleman, 2012) Pág. 119, (ii) (Paterson, 2013) Pág. 189, (iii) (Gómez Segade, 2001) Págs. 859 – 860, (iv) (Buitrago, 2007) Pág. 160, (v) Sobre la dificultad de asir la originalidad en un *software*: (Palacio Eastman, 1999) Pág. 214, (vi) (Dam, 1995) Págs. 337 – 338, 359 – 360.

⁵⁸ (i) (Berkemeyer, 2009) Pág. 127, (ii) (Guerrero Gaitán, 2006-2007) Pág. 123.

⁵⁹ (i) Artículo 9 Ley 23 de 1982, (ii) Artículo 2.6.1.1.22. Decreto 1066 de 2015, (iii) (Uribe Arbeláez, 2005) Págs. 196 – 197.

⁶⁰ (i) (Berkemeyer, 2009) Pág. 127, (ii) (Preinfalk Lavagni, 2010) Pág. 273.

⁶¹ (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 32.

⁶² Sobre el alcance de la protección para el material de apoyo: (i) (Soler Matute, 2004) Pág. 30, (ii) (Lipzyc, 2001) Pág. 104, (iii) (Byungkwon, 1989) Pág. 175.

1.1. La creación cumulativa del *software* entre innovadores.

La elaboración de un *software* implica la definición y consecución de ciertos objetivos, desde una fase inicial de ideación de la funcionalidad del *software*, pasando por un diseño y codificación de este, hasta llegar a la etapa de integración y prueba⁶³ antes de liberarlo al cliente que lo ha solicitado (*software* a la medida) o al mercado en general.

Si cada vez que se creara un *software* debiera iniciarse completamente de la nada, se estarían desperdiciando valiosos recursos al recodificar eficientemente las funcionalidades de los *softwares* conocidos o lo que es igual, se estarían replicando ensayos fallidos por lograr el mejor y mayor resultado posible en el procesamiento de datos, con el menor empleo de tiempo y recursos del *hardware*. En otras palabras, quien elabora un *software* siempre se apoyará en conceptos y resultados previamente conocidos en el estado del arte y librerías⁶⁴, aunque sea mínimamente, para así evitar caer en la prohibida ineficiencia⁶⁵.

⁶³ (i) (Soler Matute, 2004) Las etapas de elaboración del *software* comprenden: 1. Análisis de requerimientos (o conceptual) a partir de los requerimientos iniciales facilitados por el cliente (Págs. 43 y 44); 2. Análisis funcional o documento de especificaciones (Pág. 44); 3. El diseño del programa (Pág. 45); 4. La codificación, que resulta con el código fuente (Pág. 46); 5. Integración y prueba del *software*, que resulta con el ejecutable (Código objeto) y la documentación de pruebas (Pág. 48). (ii) También sobre los pasos para la confección de un *software*: (Lipzyc, 2001) Pág. 105, (iii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2328, (iv) Citando a Charria Fernando. *Derechos de autor en Colombia*: (Uribe Arbeláez, 2005) Pág. 197.

⁶⁴ (i) (Nguyen, Gomulkiewicz, & Conway, 2012) Pág. 530: “*The LGPL was created by the Free Software Foundation for use with software programs known as ‘libraries’. Libraries provide collections of functionality to application programs; the library’s functionality may be utilized by the application if it links to the library. The LGPL permits an application program to invoke a library’s functionality without the resulting program (application plus library) triggering the derivative works and source code sharing requirements of GPL §2(b)*”, (ii) (Merges, Menell, & Lemley, 2010) Pág. 302.

⁶⁵ Esta es la naturaleza “cumulativa” que caracteriza a la industria, de la cual comentan muchos. Entre ellos: (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2322 – 2323, 2330 – 2331 y 2380, (ii) (Blind Knut & Firedewald, 2005) Págs. 13 – 14.

En consecuencia, si se cuenta con el conocimiento técnico adecuado, observar el desempeño del *software* permitirá intuir cuál ha sido la ideación y diseño de este, e incluso si se cuenta además con información relacionada con el lenguaje de programación empleado podría saberse cuál es la posible codificación hecha, lo cual permite que rápidamente pueda hacerse evolucionar al *software* a través del aporte de terceros creadores⁶⁶.

Ya que el Derecho de Autor no protege ideas sino la particular expresión de estas, la condición de que el *software* sea protegido como una obra literaria permite entonces aprovechar la concepción eficiente de sus funcionalidades en productos sustitutos y/o complementarios⁶⁷, siempre que tales productos guarden originalidad⁶⁸. En presencia de originalidad en el producto sustituto y/o complementario, el Derecho de Autor lo protegerá con independencia del *software* inicial o innovador, sin considerarlo una obra derivada⁶⁹.

⁶⁶ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2330 – 2331, (ii) La característica “cumulativa” de esta industria llevó al desarrollo generoso y rentable de *software* creado por una base mundial de colaboradores, bajo licencias de código abierto: (Uribe Arbeláez, 2005) Pág. 214: “*El software libre se inspira en una ética social de espíritu comunitario, que promueve el compartir como un imperativo del ser humano, preconizando el libre flujo de información para beneficio y deleite de todos*”.

⁶⁷ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2319: “*If behavior is the primary source of value in a program, then the potential for two programs to be behavioral equivalents is of considerable competitive significance. Two programs that produce identical behavior may, from the standpoint of consumers, be perfect market substitutes. The lawsuits concerning VP-Planner and The Twin illustrate this point. There, the ability to create a program with identical behavior was far more than a laboratory curiosity*”.

⁶⁸ Distinguiendo una originalidad “objetiva” y “subjetiva” en el caso del *software*: (Preinfalk Lavagni, 2010) Pág. 170.

⁶⁹ Algún despistado podría refutar lo anterior afirmando que el Derecho de Autor concede un monopolio de la expresión del algoritmo y como tal, permite a su titular evitar la creación de obras derivadas; pero en realidad, la protección provista por el Derecho de Autor, ni siquiera la provista por el Sistema de Patentes, crea verdaderos monopolios, ni alcanza hasta cubrir funcionalidades coincidentes bajo el expediente de la obra derivada. Sobre este punto: (i) (Dam, 1995) Pag. 368, (ii) (As Colombia Ltda. v. Informática y Gestión S.A, 2006) Pág. 27: “*Si bien está comprobado en el dictamen que los dos programas comparten algunas funcionalidades, del mismo dictamen se puede deducir que las partes recorrieron caminos de trabajo distintos para llegar a ese resultado. Los medios usados, las subrutinas, los códigos y la arquitectura son distintos en cada aplicativo*”, (iii) Proponiendo que el Derecho de Autor se interprete en el *common law* de forma que el algoritmo expresado en un mismo nivel de abstracción se considere una obra derivada: (Swinson, 1991) Pág 195, (iv) Narrando los resultados de una propuesta hecha a la CONTU para que el Derecho de Autor protegiera la “lógica” del *software*: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2353 – 2354.

Todo lo anterior describe por qué el Derecho de Autor incentiva desde el punto de vista jurídico la producción de *software*, y cómo vía el aprovechamiento de este incentivo se beneficia el mercado de una mayor oferta de *software* producto⁷⁰, pues si bien los creadores de *software* deben asumir el costo de ser originales, no soportan el costo de hallar las eficiencias que el mercado premia con precios y fidelidades, lo cual libera valiosos recursos de ser empleados en duplicar esfuerzos por descubrir funcionalidades de *software* previamente desconocidas, no ensayadas o infructíferas.

En el propósito de multiplicar la producción intelectual⁷¹, el Derecho de Autor ha servido a la humanidad un par de siglos como el vehículo jurídico más equilibrado en costos y beneficios, en comprobada utilidad que la industria local y mundial del *software* ha sabido usufructuar. Lo anterior es particularmente importante a efectos de garantizar la interoperabilidad de varios *softwares*⁷², pues así se asegura una oferta amplia de productos más idóneos a los complejos y variados requerimientos de los usuarios, lo mismo que mercados secundarios igualmente dinámicos, en beneficio de industrias conexas.

1.2. La protección jurídica que reciben los elementos gráficos del *software*.

La evolución de lenguajes de programación y *hardware* por igual hizo ampliar el ropero del *software* de los años 70's, de una interfaz monocromática y cuadrículada, hacia una

⁷⁰ (Dam, 1995) Pag. 326.

⁷¹ (i) (Ascarelli, 1970) Pág 277, (ii) (Landes & Posner, 1989) Pág 332.

⁷² (Arenas Correa, 2013) Pág. 191.

comunicación con el usuario amigable, intuitiva, icónica y colorida, lo cual sin duda ha aportado mucho al desarrollo y éxito económico de las empresas dedicadas a la elaboración de *software*.

Cabalmente el Derecho de Autor es protección ideal de esos elementos gráficos, no existiendo otro modelo de protección jurídico apropiado⁷³, porque tales elementos gráficos tienen el mismo ADN de las obras artísticas y en consecuencia, la estética con que se elabore un *software* tiene asidero en las mismas razones que motivaron la codificación del Derecho de Autor, sin que importe la funcionalidad valorada que tengan⁷⁴.

En cambio, la atrayente iconografía que exponga un *software* es materia irrelevante por el Sistema de Patentes, aunque es materia abonada para el Sistema de Marcas⁷⁵, y claramente su abierta exposición le excluye de la Protección a Secretos Empresariales; por estas razones, si una protección *sui generis* se lanza a debate, deberá prestarse atención a estos contornos.

Ahora bien, vale distinguir lo concerniente a la apariencia de la interfaz del *software* con su carácter funcional, pues el Derecho de Autor podrá ocuparse del grado de similitud entre productos comparables a fin de establecer si hay copia⁷⁶, pero no se involucra en el efecto práctico que acompañe el ornamento, tal como se puso de relieve en el famoso caso decidido por

⁷³ (i) (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 69. La protección del *software* a través del sistema de marcas no está directamente relacionado con el *cloning*, motivo por el cual no se menciona en este trabajo, sí en: (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Págs 201 – 226; (ii) Sin embargo, es posible rastrear casos infructuosos en los cuales el Sistema de Marcas fue invocado para reclamar protección del *look –n- feel* de un *software*: (Dynamic, Inc. v. Structural Software, Inc. , 1991), (iii) Para (Gómez Segade, 2001), no es clara la aplicación de la normativa del Derechos de Autor, Sistema de Marcas o la Protección a Secretos Empresariales para la protección de las interfaces del *software* (Págs. 898 – 899), (iv) En contradicción con la anterior cita: (Bouza López, 1997) Pág. 344.

⁷⁴ (Paterson, 2013) Págs. 177 y 187.

⁷⁵ (Schortgen, 1994) Pág. 160.

⁷⁶ (i) (Arenas Correa, 2013) Pág. 199, (ii) (Nimmer & Krauthaus, 1987) Pág. 746, (iii) (Swinson, 1991) Pág. 194, (iv) (Calle D'Alleman, 2012) Pág. 198, (v) (Drexl, 2011) Pág. 170, (vi) (As Colombia Ltda. v. Informática y Gestión S.A., 2006) Pág. 27.

la Corte de Apelaciones del Primer Circuito de los Estados Unidos de América, confirmada por la Corte Suprema de Justicia de los Estados Unidos de América, entre Lotus Development Corp. (En adelante: “Lotus”) v. Borland International Inc. (En adelante “Borland”), 49 F.3d 807,34 (1996).

El núcleo esencial de este caso fue si la jerarquía del menú del *software* de Lotus en su hoja de cálculo podía ser traspuesta, tal cual, en el *software* de Borland, sin violar el Derecho de Autor atribuido al primero; lo mismo que si Borland, sin violar Derechos de Autor, podía ofrecer a sus usuarios una interfaz igual a la del *software* de Lotus, que podía ser más familiar para los usuarios de Borland en comparación con el *software* que había creado.

El caso se definió a favor de Borland, en consideración a que no había reescrito literalmente sus elementos⁷⁷, lo cual permitió el surgimiento de productos sustitutos y complementarios elaborados por nuevas firmas⁷⁸, lo mismo que el ahorro de costos asociados al entrenamiento de los usuarios para el manejo de funcionalidades comparables con las ofrecidas por el producto inicial⁷⁹.

El caso anotado deja claramente establecido que las funcionalidades del *software* quedan por fuera de la protección del Derecho de Autor, como debe ser, y que escudriñar en las líneas

⁷⁷ (i) Considerando que en ese caso hubo una violación a Derechos de Autor, sin apelar a ningún argumento importando de la competencia desleal: (Bouza López, 1997) Pág. 345, (ii) Opinando que la sentencia en este caso fue un retroceso: (Carranza Torres, 2008) Pág. 251.

⁷⁸ (Blind Knut & Firedewald, 2005) Págs. 13 y 14.

⁷⁹ (i) (Dam, 1995) Pág. 334, (ii) (Carranza Torres, 2008) Pág. 247.

del código fuente de un *software* la protección a su iconografía es tan equivocado, como pensar que un mismo chiste es gracioso sin importar quien lo narre.

1.3. El conjunto de Tratados internacionales en materia de Derechos de Autor.

La inexistencia de un Tratado internacional para la protección del *software*, como hubo iniciativa de negociar a finales de los 70's⁸⁰, no ha sido óbice para que las empresas dedicadas a la elaboración de *software* encuentren un respaldo jurídico trascendente en la protección jurídica de sus creaciones a través del Derecho de Autor. De hecho, la importancia del artículo 10 - 1 de ADPIC radica en la inmediata protección del *software* en 142 países (72% del planeta), vía Derechos de Autor⁸¹.

Esta ventaja inigualable supera todo intento por armonización de las legislaciones en torno a la protección del *software* vía otros sistemas, y en buena parte reduce costos de transacción asociados a la distribución del *software* a nivel global, pues permite a las firmas dedicadas a la creación de *software* conocer sus derechos y cómo exigir su cumplimiento de manera más o menos inmediata, más o menos inmune a interpretaciones normativas exóticas, lo mismo que anticiparse al tipo de licenciamiento que pueden negociar a través de las fronteras⁸².

⁸⁰ (Committee of Experts on the Legal Protection of Computer Software Second Session, 1983) Pág. 3

⁸¹ (i) (Dam, 1995) Pág. 375, (ii) (Villalba & Lipzyc, 2001) Pág. 14, (iii) (Preinfalk Lavagni, 2010) Pág. 155.

⁸² (Smith & Mann, 2004) Pág. 248.

Como en toda industria, si la demanda del producto aumenta, ora por factores endémicos (p.e. un incentivo local que amplía el mercado doméstico) o endógenos (p.e. un mismo requerimiento en varias jurisdicciones), se propician economías de escala que redundan en índices económicos positivos y multiplican el comercio internacional, este último naturalmente en función de los costos de oportunidad del país importador.

La relativa seguridad jurídica que concede un sistema organizado de Derechos de Autor a nivel global, contribuye mucho a la inercial vocación internacional de la industria del *software*, lo cual explica la tendencia exportadora detectada por los estudios económicos realizados en la industria del *software*⁸³, la prudencia hacia un sistema *sui generis* de protección⁸⁴ y la inclinación hacia leyes especiales para la protección del *software* dentro del marco del Derecho de Autor⁸⁵.

2. Algunas externalidades negativas aparejadas a la protección jurídica del *software* mediante el Derecho de Autor

Cuando muchos expertos en propiedad intelectual fueron convocados para analizar la mejor forma de protección jurídica para el *software*, poco se conocía en realidad del dinamismo que tendría la industria y consecuentemente cuáles serían los intereses jurídicos en juego a los cuales el Derecho debía responder⁸⁶ y, como suele suceder, debieron conocerse los primeros casos judiciales para descubrirlos.

⁸³ (Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia, 2014) Pág. 61.

⁸⁴ (i) (Palacio Eastman, 1999) Págs. 215 y 216, (ii) (Nimmer & Krauthaus, 1987) Pág. 748, (iii) (Dam, 1995) Págs. 372 – 373, (iv) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 7.

⁸⁵ (Nimmer & Krauthaus, 1987) Pág. 737.

⁸⁶ (i) (Measures to enhance international cooperation in the field of legal protection of computer software, 1979) Pag. 2, (ii) (Nimmer & Krauthaus, 1987) Pág. 736.

Ocurridos estos, siguió un período de análisis por la academia dando como resultado textos doctrinales que recogen el sentir de la industria, echando de menos una protección adecuada a la funcionalidad del *software*⁸⁷; aunque, desde el punto de vista teórico, se deben hacer reparos adicionales por las inconsistencias⁸⁸ que el régimen del Derechos de Autor presenta frente a su aplicación en el *software*.

Esto último se encuentra explicado en casi todo texto relacionado, porque la equivalencia del *software* a las obras literarias es severamente cuestionable y arduamente debatido entre doctrinantes si se tiene en cuenta que las letras y símbolos con los cuales se codifica un algoritmo no es literatura para humanos⁸⁹, son solo instrucciones dirigidas a un *hardware* para

⁸⁷ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2350, (ii) (Landes & Posner, 1989) Pág. 352.

⁸⁸ (Laboratorios California S.A. v. System Software Associates Inc. y S.S.A. Colombia S.A., 2001) “*Las modernas legislaciones, tendencias jurisprudenciales, y en general, la actual realidad jurídica del software ubica su protección y en especial la protección de las relaciones y derechos que en torno a él se desarrollan, dentro del ámbito de los derechos de autor. Dicha ubicación se ha presentado más que por un desarrollo coherente y lógico de carácter jurídico, en virtud a inevitables determinantes externos de tipo económico*”... “*Existen muchas razones eminentemente jurídicas que sustentan de forma coherente esta posición, de la misma manera que existen razones valederas que rebaten estos argumentos. Lo único que pretende dejarse planteado es que la protección del software por el derecho de autor que en la actualidad se tiene como norma en la mayoría de países, en gran medida fue determinada por los importantes intereses económicos que esta materia reviste. Si no se podía proteger el software mediante el régimen de patentes, invenciones y modelos de utilidad, lo más adecuado era ajustar, así fuese apretadamente, el software al derecho de autor. De lo contrario se hubiera tenido que plantear una nueva disciplina jurídica, con lo que las medidas prontas, eficaces e inmediatas necesitadas en dicho instante hubieran tenido que esperar un lapso insostenible. Sin embargo, sería conveniente que paralelamente con la protección otorgada, desde el punto de vista de la lógica jurídica que se le viene otorgando al software mediante las normas del derecho de autor, se elaborara un régimen acorde y ajustado con lógica a la realidad de un objeto novedoso y particular como lo es el software*”.

⁸⁹ (I) En contra de la inclusión del *software* como materia del Derecho de Autor: (i) (Calle D'Alleman, 2012) Pág. 42, (ii) (Palacio Eastman, 1999) Pág. 221, (iii) (Leith, 2007) Pág. 159. Este artículo trae un análisis actualizado y concienzudo del Manifiesto. Utiliza igualmente la expresión “*industrial copyright*”, que se dice viene del modelo de protección para el *software* de la OMPI/78. Esa expresión también la utiliza: (Nguyen , Gomulkiewicz , & Conway , 2012), (iv) (Carranza Torres, 2008) Pág. 226, (v) (Botana Agra, 2001) Pág. 169, (vi) (Smith & Mann, 2004) Pág. 256, (vii) (Paterson, 2013) Pág. 188, (viii) (Hong & Cheng-ssu, 1999) Pág. 34. (II) A favor de la inclusión del *software* como materia del Derecho de Autor: (i) (Ríos Ruíz, 2002) Pag. 83, (ii) (Gómez Segade, 2001) Pag. 856, (iii) (Dam, 1995) Pag. 326, 376 – 377, (iv) (Preinfalk Lavagni, 2010) Pág. 156. (III) En procura de una reforma al actual régimen: (i) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1364, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2399, 2406 y 2413, (iii) (Swinson, 1991) Pág. 193. Este último artículo, más orientado hacia

que se obtenga un resultado, y porque el Derecho de Autor no es el campo para creaciones del intelecto concebidas para fines utilitarios⁹⁰.

Teniendo en cuenta lo anterior, a continuación se describirán las principales inconformidades halladas en torno a la aplicación del Derecho de Autor como modelo ideal de protección para el *software*, las cuales ponen en evidencia graves externalidades negativas y el déficit de protección predicado.

2.1. Dicotomía entre idea y expresión de la idea/Test de Abstracción-Filtración-Comparación.

Aún sigue siendo célebre la prueba de Abstracción - Filtración - Comparación (“Test AFC”) para la distinción entre idea y expresión de la idea en cuanto a *software* se refiere⁹¹, empleado por la Corte de Apelaciones del Segundo Distrito de Estados Unidos de América en el sonado caso *Computer Associates International Inc. (En adelante: “CA”) v. Altai Inc.* (En

una *lex specialis* del Derecho de Autor, los dos primeros más hacia un híbrido o protección *sui generis*. (IV) En una posición neutral o indecisa: (i) (Lipzyc, 2001) Págs 108 a 111, (ii) (Teruo, 1992) Pág. 98, (iii) (Nimmer & Krauthaus, 1987) Pág. 748 – 750, (iv) (Córdoba Marentes, 2006) Pág. 141, (v) (Salas Pasuy, 2016) Pág. 439, (vi) (Baylos Corroza, 2009) Pág. 762.

⁹⁰ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2310 – 2311, (ii) (Buitrago, 2007) Págs. 158 – 159, (iii) (Leith, 2007) Pág. 159, (iv) Narrando las dificultades iniciales de la Oficina de Derechos de Autor de Estados Unidos de América con los primeros asuntos de registro de *software*: (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 32, (v) Respondiendo a este argumento: (Dam, 1995) Pág. 325: “*The argument is occasionally heard that the usefulness of software makes it constitutionally inappropriate for software protection because the constitutional text indicates that “useful Arts” are to be protected by patent only*”... Más Adelante en el mismo texto se deja ver la posición del autor a favor de la inclusión del *software* como materia protegible por el Derecho de Autor: Pág. 326: “*The conclusion that the usefulness of software is an argument for, rather than against, copyright protection of software is thus fully justified not just by the history of the copyright laws but also by the text of the Constitution when read in the light of the usage of the constitutional terms by the Founders at the time of the drafting and by both the Congress and the Supreme Court in the century that followed*”, (vi) A su lado: (Swinson, 1991) Págs. 180 y 212.

⁹¹ (i) Recordando que no es un problema exclusivo de la protección para el *software*: (Ginsburg, 1994) Págs. 2650 – 2651, (ii) Reseñando un caso espejo en Europa: (Graef, 2014) Pág. 13 - 14.

adelante: “Altai”).

En esencia, el problema jurídico resuelto en el proceso versó sobre si ciertas porciones del código fuente del *software* producido por CA, que habían sido conocidas por ex funcionarios de CA contratados por Altai, constituían ideas en sí mismas o expresión de ciertas ideas; al haber sido incluidas inicialmente en un *software* de Altai y posteriormente eliminadas de éste, pero conservando el *software* de Altai las mismas funcionalidades del *software* producido por CA.

Para resolver este problema, la Corte de Apelaciones del Segundo Distrito de Estados Unidos de América diseñó y aplicó una prueba de tres pasos así:

Primer paso. “Abstracción”: A manera de ingeniería reversa⁹², quien esté aplicando la prueba deberá apartar las estructuras del *software* presuntamente infractor, en cada nivel de codificación, en sentido opuesto a como debieron ser articuladas las líneas del código para lograr la funcionalidad resultante. El resultado de este paso será conocer cuál ha sido el iter lógico del programador, que determinó de esa forma la programación de las líneas de código fuente señaladas de infringir Derechos de Autor.

Segundo paso. “Filtración”: Este paso consiste en examinar cuál ha sido la razón que determinó de esa forma la programación de las líneas de código fuente señaladas de infringir Derechos de Autor, identificando si debióse a una idea, previamente conocida o desconocida, a un criterio de eficiencia o a particularidades dictadas por el lenguaje de programación empleado

⁹² Una noción de qué es la ingeniería reversa, en: (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 115.

para el *software*.

Tercer paso. “Comparación”: Descartadas las anteriores razones, lo remanente es expresión de una idea protegible por Derechos de Autor, no apropiable por terceros sin previa autorización del titular del derecho.

Reconociendo el esfuerzo de esta Corte por dotar a la ciencia jurídica de una herramienta ingeniosa para fallar los difíciles casos de *cloning*, ciertamente la complejidad del test le hace verdaderamente ineficiente para resolver de forma satisfactoria estos asuntos, pues poca materia protegible por Derechos de Autor pasa el umbral de la “A”, debido precisamente al propósito funcional del *software*, y porque basta un lenguaje de programación diferente entre los *software* comparados para anticipar siempre un resultado en contra de la protección del *software* inicial o innovador.

En efecto, la funcionalidad explícita a la que apunta el *software* no deja margen para estilos rococó, pues los algoritmos que componen una aplicación son dictados por la eficiencia, para mayor eficiencia, reduciendo a un número mínimo las opciones de expresión⁹³, a fin de no caer en redundancias que afecten los resultados esperados del *hardware*, ni en costos de

⁹³ (i) (Swinson, 1991) Págs. 158 – 159, (ii) (Paterson, 2013) Pag. 189: “*The ideal in coding is to perform a necessary function in the least amount of lines of code while utilizing standardized structuring and indentation, thus the best method of computer programming is the most efficient. If the AFC test is applied to this practice, in most cases properly executed ‘tight code’ will be filtered out as absolute efficiency (as not to create ‘monopolistic stagnation’). Alternatively, if code is structured in a manner that is unconventional, then this way attract copyright protection despite is inefficiency. If this is the case, it may be seen that copyright rewards inefficiency in code through the filtration process. In other literary works, creative expressions are rewarded with copytight however, it could be the opposite in computer software*”, (iii) Considerando que el algoritmo carece de protección a través del Derecho de Autor: (Preinfalk Lavagni, 2010) Pág. 133.

reentrenamiento⁹⁴ por parte de los usuarios.

Si a esto se suma que, no solo caben muchas subjetividades en cuanto a la determinación de los grados de abstracción aplicables a cada caso, sino también que conforme se avanza en lenguajes de programación son estos cada vez más abstractos, y en consecuencia más simples los comandos de codificación⁹⁵ - por no mencionar que el progreso de la inteligencia artificial hará posible la codificación automática -, realmente la prueba AFC está “mandada a recoger”⁹⁶.

Con todo, si estamos para plantear un nuevo modelo de protección para el *software*, debe rescatarse el propósito implícito de la prueba AFC en no sobreproteger la propiedad intelectual respecto del *software*, a fin de no reservar derechos a ideas abstractas, y paulatinamente permitir la aparición de firmas sucesoras (“*follow-on*”) que hagan avanzar el grado de eficiencia alcanzado por el producto innovador⁹⁷.

2.2. Alcance de los derechos morales y patrimoniales concedidos al *software*.

Punto en el cual el Derecho de Autor asociado al *civil law* tiene más dificultades que en el *common law* es el de los derechos morales concedidos al autor de un *software*⁹⁸, particularmente

⁹⁴ “*training costs*” o “*switching costs*”: (Dam, 1995) Pág. 347 y 349. Para este autor, el concepto de “costos de reentrenamiento” se encuentra más acorde con lo que sucede en la industria, y mejor sustentado que las nociones de *estándares de facto*, la externalidad de redes e interoperabilidad (Pág. 351).

⁹⁵ (Swinson, 1991) Pág. 174.

⁹⁶ (i) (Paterson, 2013) Pág. 186, (ii) (Swinson, 1991) Págs. 195 – 198. En este artículo, contemporáneo a la sentencia comentada, se refina un test de similitudes, basado en el caso (Whelan v. Jaslow, 1986), y se critica el presentado en el caso (Computer Associates International Inc. v. Altai Inc., 1992).

⁹⁷ (Dam, 1995) Págs. 337 y 359.

⁹⁸ (Palacio Eastman, 1999) Pág. 215.

en lo concerniente a los derechos morales de integridad y retracto. Esto, por cuanto los autores del *software* y sus descendientes están facultados, por la eternidad, para exigir que la obra NO sea actualizada sin contar con su consentimiento, o interrumpir su explotación por parte de titulares derivados⁹⁹.

Ciertamente los derechos morales no pueden ser ejercidos en abuso del derecho, y por ello la norma que los consagra en Colombia – y así en otras latitudes – condiciona tales facultades a que sea afectada la dignidad del autor con la intervención que rompa la integridad de la obra¹⁰⁰, y a que sea abonada la indemnización correspondiente si se detiene la explotación de la obra vía el ejercicio del derecho de retracto; pero lo adecuado para la creación a la que nos referimos, es que tales derechos no se reconozcan.

Y así es la óptima reglamentación que se debe contemplar para el *software*, pues los costos de transacción que producen estas facultades tienen que ver tanto con las facilidades propias del Derecho de Autor, como por las características de la industria a la que nos referimos. Lo primero, pues la falta de registros públicos como condición de protección, dificulta mucho rastrear la persona natural que por primera vez expresó el algoritmo correspondiente, a fin de pedirle permiso para intervenir el código fuente; lo segundo, porque la velocidad con que cambian las bases de datos, los lenguajes y paradigmas de programación solamente es superada

⁹⁹ (i) Artículo 30 Ley 23 de 1982, en concordancia con el artículo 6 *bis* del Convenio de Berna para la Protección de las Obras Literarias y Artísticas (Ley 33 de 1987), (ii) (Baylos Corroza, 2009) Pág. 800: “*Hablar, pues, de un «derecho moral» sobre la obra después de que el autor ha muerto, en principio es ya un contrasentido. Lo que sucede es que, aún después de la muerte del autor, existe un interés familiar y en definitiva social -un interés público-, en velar por los valores intelectuales y estéticos, por la fama y el nombre del autor y por la conservación de su obra libre de deformaciones que la degraden o falseen. Y eso inclina a la ley a legitimar a determinadas persona vinculadas al autor en vida, para ejercitar algunas de las facultades que integraban ese derecho moral desaparecido*”.

¹⁰⁰(i) (Preinfalk Lavagni, 2010) Pág. 74, (ii) (Rengifo García, 2003) Pág. 131, (iii) (Calle D'Alleman, 2012) Pág. 274.

por la rotación del personal técnico vinculado a las firmas de la industria¹⁰¹.

Además, existe una conciencia entre el personal vinculado a las firmas respecto al natural y constante mantenimiento¹⁰² que exigen los *software*, lo mismo que a la titularidad de derechos en cabeza de éstas¹⁰³, que tanto el *civil law* como el *common law* deben emular; ya que, en el actual régimen, un excolaborador “muy emprendedor”, podría impedir el mantenimiento del *software* para lanzar su propia versión, ejercitando el derecho moral de integridad que se le confiere¹⁰⁴, cuando no el ejercicio de la titularidad originaria que se le otorga sobre su aporte¹⁰⁵ si

¹⁰¹ (i) (Allen, Amstrong, Reid, & Riemenschneider, 2009) Pág. 96, (ii) (Caracterización del sector de teleinformática, software y TI en Colombia. Año 2015. , 2016) Pág. 155.

¹⁰² (i) Apuntando que el “mantenimiento” no debería considerarse una afectación al derecho de transformación: (Gómez Segade, 2001) Pág. 876, (ii) En Bogotá, D.C. es costumbre mercantil, certificada por la Cámara de Comercio local, que el término “mantenimiento” en los contratos de licencia de *software* sea entendido de forma tal que no comprende los servicios de soporte, sino el de “actualizaciones”.

¹⁰³ (i) Como trabajo de campo se hizo una encuesta pública entre programadores y firmas de la industria del *software*, en la cual se les preguntó si los clientes les exigían transferir derechos patrimoniales de autor. El 57% de los encuestados respondió que no, pero que la transferencia era implícita en los contratos, (ii) Describiendo algunas dificultades jurídicas de aplicar las reglas de titularidad en el *software*: (Gómez Segade, 2001) Pág. 867, (iii) (Guerrero Gaitán, 2006-2007) Pág. 128: “Para el otro supuesto mencionado en párrafos precedentes, la prohibición de obra futura, mantengamos el ejemplo anterior; múltiples desarrolladores sin que medie un contrato laboral. En este escenario, funcionan numerosas empresas pequeñas y medianas de desarrollo de software, con lo cual no podemos acudir a la transferencia realizada ope legis vía contrato laboral. En este caso, teniendo en cuenta la prohibición de algunas legislaciones en cuanto a la transferencia de derechos de obras que aún no existen se debería realizar un acto de transferencia por cada desarrollo que se logre y para cada sujeto, con las correspondientes molestias y dificultades que ello supone”.

¹⁰⁴ (i) (Arenas Correa, 2013) Págs. 169 – 170: “Una norma que puede generar prácticas diversas de explotación por parte de colaboradores en la fabricación de software conjunto, generando incentivos para su conversión en competidores, se encuentra en el art. 19 de la Ley 23 de 1982; esto porque vía ‘derechos morales’ facilita a quienes fueran contratista del grupo desarrollador, especialmente si se trata de un diseño de software por módulos, que los diseñadores particulares de cada módulo utilicen su propia obra para crear nuevos sistemas de soporte lógico que pueden contener diferencias en lo que respecta a otros módulos, generando productos competidores que vía normativa de derechos de autor no encontrarían la protección suficiente”. Esto se debe impedir normativamente, (ii) Resaltando una postura contraria este parecer, fundado en una sentencia del Tribunal Supremo Español: (Preinfalk Lavagni, 2010) Pág. 101: “El TS ha entendido que la actualización de una obra es una vía que le da contenido al derecho patrimonial de transformación, máxime, si desde un inicio se previó tal actualización. Por ello, si los derechos patrimoniales fueron cedidos, el autor no podrá alegar violación a su derecho moral en caso de que el cesionario de los derechos económicos ejecute la actualización. Pero esta postura no concuerda con la doctrina, donde se ha dicho que una de las condiciones extrínsecas u objetivas, que le pueden dar origen al ejercicio de la facultad de modificación de la obra, es la necesidad de actualización”.

¹⁰⁵ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2327.

no se cumplen con los parámetros de la obra en colaboración y por encargo¹⁰⁶, o como director e incluso colaborador si al *software* se le considerara una obra audiovisual – p.e. si es un videojuego¹⁰⁷ -.

El anterior costo de transacción debe eliminarse de la norma jurídica, como *de facto* se ha eliminado por el mercado, lo mismo que ciertas normas atinentes a los derechos patrimoniales de autor¹⁰⁸, y no a los morales, concretamente el sometimiento del *software* al principio de independencia en la explotación de las obras protegidas por el Derecho de Autor¹⁰⁹, que sirve de paraguas para todos los actos de disposición de obras protegidas, entre ellas los actos de transformación¹¹⁰; así como las excepciones consagradas en favor de la interoperabilidad¹¹¹ y la

¹⁰⁶ (i) Arts. 18, 19 y 20 de la Ley 23 de 1982, (ii) (Gómez Segade, 2001) Pág. 876, (iii) Basados en jurisprudencia dictada sobre el Sistema de Patentes, pero con muchas semejanzas a lo descrito aquí: (Hong & Cheng-ssu, 1999) Pág. 291: “*The court also raised the question of the relationship between the higher version and lower version of a technology [Wangma Co. v Southeast Co.]. Since software may be updated constantly, if the patentee of a lower version of software could prevent others using the higher version of software, the patent would hinder software development. In this case, the court correctly pointed out that a higher version may derive from a lower version, but cannot be covered by the lower version, and using the higher version would not infringe the patent right in the lower version*”.

¹⁰⁷ Al respecto de la protección de los videojuegos como obras audiovisuales: (Carranza Torres, 2008) Pág. 264.

¹⁰⁸ (Gómez Segade, 2001) Págs. 876 - 877: “[T]oda transformación de un programa requiere el consentimiento del autor, a menos que se trate de cualquiera de las excepciones establecidas en el art. 5 de la LIPRO. Por lo tanto, el derecho de adaptación de un programa informático de un sistema a otro (‘conversión’) pertenece exclusivamente al titular del derecho de autor del programa, aunque se pueden introducir matices. La persona que haga transformaciones autorizadas tendrá derecho a una protección adicional del derecho de autor como transformador o adaptador, en tanto que tales transformaciones o adaptaciones constituyen su propia creación intelectual personal (arts.11.5 y 21.2 de la LPI). Asimismo, el autor del programa original estará legitimado para participar de los derechos de explotación de todo tipo de transformación (art. 21.2 de la LPI)”.

¹⁰⁹ (i) Ley 23 de 1982, artículos 77 (“*Las distintas formas de utilización de la obra, son independientes entre ellas; la autorización del autor para una formas de utilización no se extiende a las demás*”) y 183 (Modificado por el artículo 181 de la Ley 1955 de 2019: “*...Será ineficaz toda estipulación en virtud de la cual el autor transfiera de modo general o indeterminable la producción futura, o se obligue a restringir su producción intelectual o a no producir. Será ineficaz toda estipulación que prevea formas de explotación o modalidades de utilización de derechos patrimoniales de autor o conexos, que sean inexistentes o desconocidas al tiempo de convenir la transferencia, autorización o licencia*”), (ii) La Ley española sobre derecho de autor consagra esto mismo en sus artículos 43.1, 43.2, 43.4 y 43.5. Al respecto (Baylos Corroza, 2009) Págs. 893 – 894, (iii) (Lipzyc, 2001) Págs. 277 – 278.

¹¹⁰ Art. 12-b). Ley 23 de 1982

¹¹¹ (i) Artículo 13-a Ley 1915 de 2018, (ii) Artículo 6 Directiva 2009/24/CE, (iii) Proponiendo un mayor número de excepciones y limitaciones a los derechos de autor conferidos al *software*, debido a lo prolongado de su término de protección, en consideración a la naturaleza del bien: (Aguirre Soriano, 2014) Pág. 191, (iii) Repudiando una

reproducción del *software* en la memoria RAM del ordenador¹¹².

Tales reglas de excepción responden adecuadamente a tipos de obras no-utilitarias como el *software*¹¹³, por lo cual señalamos su inconclusa reglamentación en favor de firmas sucesoras (“*follow-on*”) de la innovadora, gracias a los derechos morales de integridad y retracto; y en lo que respecta al principio de independencia de las formas de explotación de las obras, su aplicación para el caso del *software* implica condenar a mayores costos de transacción a todas las firmas dedicadas a la creación de *software*, sean estas innovadoras, sucesoras (“*follow-on*”) o competidoras (“*me too*”), por la necesaria re-negociación de términos que conllevan los cambiantes modelos de negocios y tecnologías a las cuales aplicar los algoritmos correspondientes.

2.3. Efecto económico de la ingeniería reversa.

La ingeniería reversa en la industria del *software*, a diferencia de la reglamentación consagrada para otras actividades económicas, ha sido patrocinada desde un principio en procura de una complementariedad entre *software*¹¹⁴, en futurista propósito de integrar adelantos que hoy

eliminación de la excepción al derecho por descompilación, debido a los prolongado de los plazos de protección: (Samuelson & Scotchmer, *The law and economics of reverse engineering*, 2002) Pág. 1650.

¹¹² (Nguyen, Gomulkiewicz, & Conway, 2012) Pág. 535 – 536.

¹¹³ Por eso no las encontraremos en el Sistema de Patentes, aunque podría un sistema de protección del *software* basado en las patentes contemplar algunas nociones de estas excepciones.

¹¹⁴ (i) (Samuelson & Scotchmer, *The law and economics of reverse engineering*, 2002) Pág. 1580 y 1650, (ii) (Cohen & Lemley, *Patent scope and innovation in the software industry*, 2001) Pag. 17: “*While there is no express statutory provision in the copyright laws permitting reverse engineering [En Colombia sí], virtually every court to consider the issue has concluded that there is a right to reverse engineer a copyrighted program for at least some purposes. The source of that right is generally considered to be the fair use doctrine...*”, (iii) (Charfoos, 2002) Pág. 10: “*Notably, TRIPS is silent on the issue of reverse engineering. It is also clear that by only protecting source and object codes as literary works, reverse engineering is not affected by TRIPS Art. 10. Such silence basically affirms the practice of many countries that allow reverse engineering*”.

por hoy caracterizan a la industria. Es gracias al grado de ingeniería reversa permitido, por lo que actualmente existen conexiones entre *hardware* diseñados para distintos propósitos y el Internet de las cosas¹¹⁵, sin que tal afirmación sea un atrevimiento académico o histórico de nuestra parte.

No obstante, sí nos atrevemos a afirmar que el efecto económico de la ingeniería reversa en la industria del *software* es y ha sido muchísimo más costoso socialmente, que el resultado por la concesión de patentes sobre invenciones implementadas por programas de ordenador. Esto, porque el cuidado empleado por las firmas de la industria para evitar infracciones a Derechos de Autor es inconmensurable¹¹⁶, al punto de haber jugado un papel central en la reaccionaria formación y proliferación de movimientos de *software* libre¹¹⁷.

Por si no fuera poco lo anterior, el que emprende la ingeniería reversa gasta cuantiosos recursos económicos y talentoso personal en descubrir la codificación del *software*¹¹⁸, en procura del autorizado propósito de comunicar su *hardware* o *software* con el inicial – o sin este propósito –; y el que procura evitar la ingeniería reversa también lo hace, para hacer difícil tal

¹¹⁵ (i) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 21, (ii) Sobre un concepto del “Internet de las cosas”: (The Internet of things. Seizing the benefits and addressing the challenges, 2016) Pág. 8: “IoT [“Internet of Things”] refers to an ecosystem in which applications and services are driven by data collected from devices that sense and interface with the physical world. In the Internet of Things, devices and objects have communication connectivity, either a direct connection to the internet or mediated through local or wide area networks”.

¹¹⁶ (i) (Dam, 1995) Pág. 368: “A computer software firm setting out today to write new software must take into consideration the possibility that it will be limited in what it can do by both patents and copyrights belonging to a prior innovator. But there is a crucial difference between patents and copyrights. Since independent creation is a complete defense to copyright infringement actions, a firm does not have to worry about copyrights belonging to others, so long as it does its own work. For patents, independent creation is not a defense, and therefore a patent search is prudent, at least before commencing marketing”, (ii) (Calle D'Alleman, 2012) Pág. 99, (iii) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1330 y 1343, (iv) (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1613, (v) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2409, (vi) Afirmando que las patentes de *software* “quiebran” a las pequeñas y medianas compañías informáticas, en fanática aversión a las patentes: (Uribe Arbeláez, 2005).

¹¹⁷ (Peña Valenzuela, 2013) Pág. 69.

¹¹⁸ (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1614.

descubrimiento¹¹⁹. En otras palabras, ambos terminan consumiendo recursos que habrían podido ser empleados para fines menos censurables¹²⁰, teniendo siempre presente que el proceso de descompilar para hacer ingeniería reversa es MUY caro, en comparación con la simple reproducción servil del código objeto¹²¹, si bien la ingeniería reversa resulta siendo un costo tolerable para firmas sucesoras (“*follow-on*”) y competidoras (“*me-too*”)¹²².

La permisión de la ingeniería reversa para la conectividad e interoperabilidad entre *softwares* lleva consigo el riesgo moral (“*moral hazard*”) de incentivar la imitación del *software* descompilado, una vez conocido su código fuente¹²³; lo cual, conduce al titular inicial del *software* a la compleja tarea de discriminar entre usuarios finales y desarrolladores¹²⁴, con su respectivo acuerdo de licencia, así como codificar en capas¹²⁵ el *software*, para preventivamente evitar o mitigar tal riesgo moral.

¹¹⁹ (Samuelson & Scotchmer, The law and economics of reverse engineering, 2002) Pág. 1649.

¹²⁰ Como advierto sobre las páginas 329 y 330, en: (Dam, 1995).

¹²¹ (Posner, Economic analysis of Law, 2011) Pág. 412.

¹²² (i) (Salas Pasuy, 2016) Pág. 439, (ii) Las empresas competidoras (“*me too*”) de esta industria no necesitan mucho capital para establecerse y empezar a codificar, pero sí para salir al mercado exitosamente (Athreya, 2007) Pág. 8, (iii) Como el clonador aprovecha el camino de “ensayo y error” transitado previamente por el innovador, sus costos son inferiores y tolerables: (Samuelson & Scotchmer, The law and economics of reverse engineering, 2002) Pág. 1587, (iv) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2337, 2378. Y afirmando que un desarrollo puede demandar alrededor de dos (2) años antes de alcanzar la estabilidad para que sea lanzado al mercado: Pág. 2408, (v) (Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia, 2014) Pág. 8: “[L]as empresas de software y servicios relacionados no requieren de elevadas inversiones para entrar al mercado, son intensivas en mano de obra calificada, generan conocimiento y tiene un impacto significativo sobre los demás sectores económicos”.

¹²³ (i) Una noción de “*moral hazard*” se puede leer en, (Cooter & Ulen, 2012) Pág. 48, (ii) (Gómez Segade, 2001) Págs. 886 – 889, (iii) Describiendo este riesgo moral, al opinar sobre la copia de las “interfaces”: (Dam, 1995) Pág. 330.

¹²⁴ (i) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1368, (ii) (Blind Knut & Firedewald, 2005) Pág. 13: “*Besen and Raskind (1991) see patent protection in the software sector as less significant because an effective price differentiation allows the producers of innovative products to appropriate a larger part of the returns on innovations even without formal property rights*”, (iii) (Posner, Economic analysis of Law, 2011) Pág. 53: “[T]he copyright holder cannot perfectly discriminate – that is, that he cannot charge different prices to different purchasers such that no purchaser is turned away who is willing a price above opportunity cost (which in the case of a public good that can be copied at zero cost, would be zero”.

¹²⁵ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2405.

Además, quien logra obtener un producto a través de la ingeniería reversa, podrá acceder al mercado formado por la plataforma, y beneficiarse de la publicidad que se dispense en su lanzamiento y sostenimiento, impidiendo al innovador colmar sus costos directos, indirectos y de oportunidad¹²⁶.

En fin, el efecto económico de la ingeniería reversa es nocivo para la economía en general, lo cual se ampliará al tratar la protección del *software* a través de secretos empresariales (Capítulo III); pero en lo que respecta al Derecho de Autor, terminemos por indicar que es la génesis del *cloning*, y por ende se debe evaluar mejor su amparo jurídico.

2.4. Efecto político, económico y jurídico del *cloning*.

Criticar el *cloning* en forma separada a lo expresado en el punto anterior, consiste en no repetir el lamento por el gasto en la elaboración de productos sustitutos y complementarios vía ingeniería reversa, sino en criticar el poco espacio que tiene el innovador para recuperar los costos directos, indirectos y de oportunidad al lanzar en el mercado su *software*¹²⁷, y por el camino revelar las carencias del Derecho de Autor como modelo de protección ideal para el bien analizado.

¹²⁶ (Dam, 1995) Pág. 329.

¹²⁷ (i) Aclarando que el *cloning* es saludable para la economía, en la medida que antes de la aparición de productos sustitutos y/o complementarios se le permita al innovador recuperar todos sus costos: (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1593, (ii) Enumerando un listado amplio de consecuencias adversas asociadas al *cloning*: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2381.

En efecto, las patentes de invención implementadas por programas de ordenador son la reacción al *cloning*, en el intento por contar con suficiente tiempo para recuperar costos antes de que un competidor lance un producto sustituto, amparado por la excepción de originalidad e independencia que el Derecho de Autor resguarda¹²⁸. Con esto se aumentó la tensión respecto a dónde ubicar mejor al *software* dentro de la propiedad intelectual¹²⁹, pues tal salida es tan criticada, como su naturalización en el Derecho de Autor.

No es extraño encontrar en la literatura un argumento de orden moral para censurar el *cloning*¹³⁰, fundado en los mismísimos orígenes económico-jurídicos de la propiedad¹³¹, como tampoco es difícil oír las voces que se levantan contra el fenómeno enarbolando las banderas no-utilitarias del Derecho de Autor¹³². Y así como estos críticos - de los cuales hace parte el autor de estas líneas - se alarman impotentes por el estado avanzado de esta externalidad negativa, producto del régimen legal aplicado al *software*, todos son conscientes de la razonabilidad en consentir cierto grado de cooperación entre firmas innovadoras, sucesoras (*“follow-on”*) y competidoras (*“me-too”*) para no detener el desarrollo de la industria¹³³ - también yo -, lo cual

¹²⁸ (Samuelson & Scotchmer, *The law and economics of reverse engineering*, 2002) Pág. 1623.

¹²⁹ Sobre la necesidad del Derecho en categorizar y ordenar acertadamente los bienes jurídicos: (Baylos Corroza, 2009) Pág. 496

¹³⁰ (i) (Leith, 2007) Pág. 168, (ii) En defensa del *cloning*: (Swinson, 1991) Pág. 191.

¹³¹ (i) (Galán Corona, 2011) Pág. 377, (ii) (Carraza Torres, 2004) Pág. 168: “*La consideración del sujeto desde la perspectiva de la justicia obliga a reconocer legalmente un derecho monopólico de explotación a favor del autor del software. Esto es innegociable. Un régimen jurídico que no reconozca y garantice el ejercicio de la mentada exclusividad no cumple con el requisito de justicia de tratar de manera desigual a quienes son desiguales en mérito, es decir, es lisa y llanamente, injusto. Y lo que es injusto sí que debe ser cambiado*”, (iii) (Ascarelli, 1970) Pág. 303. El problema del argumento moral radica en que su postulación no deja campo para el ejercicio de comparar, de forma transitiva, argumentos paralelos sobre la “injusticia” del problema analizado y la “justicia” de la solución propuesta, y de esa forma decidir sobre “la” óptima solución e implementación adecuada.

¹³² (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2348 y 2350, (ii) (Salas Pasuy, 2016) Pág. 440.

¹³³ (i) (Samuelson & Scotchmer, *The law and economics of reverse engineering*, 2002) Pág. 1655 y 2367, (ii) (Dam, 1995) Pág. 364, (iii) (Nguyen, Gomulkiewicz, & Conway, 2012) Pág. 515, (iv) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2366 – 2367, (v) (Cohen & Lemley, *Patent scope and innovation in the software industry*, 2001) Pág. 21.

derechos de exclusiva pueden imposibilitar – en especial los reconocidos por las patentes-.

Con todo, NO hay evidencia empírica que sugiera una disminución en la producción del *software* debido al *cloning*, o a la proliferación de patentes de invención implementadas por programas de ordenador; pero la concurrencia en la oferta jurídica aplicable para la protección del *software* multiplica costos legales, proporcionalmente la litigiosidad en casos de infracciones¹³⁴ y de competencia desleal, además de aumentar las controversias con la Administración; esto último, porque las firmas innovadoras y *clonadoras*, casi por igual, son proclives a disputar las decisiones que se dictan a favor y en contra del otorgamiento de patentes implementadas por programas de ordenador¹³⁵.

Ahora, la dificultad en determinar el grado de cooperación óptimo entre el tipo de firmas arriba indicado, en obvia alusión a Pareto¹³⁶, es el punto común entre la declaración política, el análisis económico del Derecho y la conclusión jurídica: Debe adjudicarse al innovador el

¹³⁴ (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 56.

¹³⁵ (i) (Bessen, The patent troll crisis is really a software patent crisis, 2013) Pág. 6: “*Research shows that a software patent is four times as likely as a chemical patent to be involved in litigation; a software patent on a method of doing business is thirteen times more likely to be litigated. Nor are these high rates of litigation a result of the newness of software patents*”, (ii) Señalando que las firmas sucesoras (“*follow-on*”) no se consideran así misma infringiendo derechos de terceros, lo que explica parcialmente por qué atacan a las patentes de *software* concedidas: (Dam, 1995) Pág. 331, (iii) (Alice Corp. v. CLS Bank International, 2014), este caso vuelve a la palestra el concepto de “*materia patentable*” aplicado al *software*, (iv) (Gugliuzza & Lemley, 2018) Pág. 774.

¹³⁶ (i) Una descripción del teorema de Pareto en: (Calabresi, 1991) Pág. 1215: “*What is the Pareto test? It is a simple unanimity requirement. It says that a society is not at its optimal position if there exists at least one change which would make someone in that society better off and no one in it worse off*”. El mismo artículo señala por qué no es útil dicho teorema, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2340 y 2407, (iii) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1364: “*Part of the reason for copyright's inability to promote economic efficiency in the provision of computer products is that the public goods and network externality problems suggest conflicting modes of legal protection. Public goods problems are alleviated by expanding legal protection for intellectual work. External benefits from networks are promoted by facilitating access to a standard. Thus, the difficult policy question is how to promote standardization while at the same time encouraging continuing innovation (along the entire spectrum from software to hardware)*”.

suficiente tiempo para incentivarle a seguir creando¹³⁷, sin deprimir el desarrollo de productos innovadores¹³⁸; lo cual exige asegurar la vibrante competencia económica que se evidencia en esta industria, mediante un modelo de protección diferente al Derecho de Autor.

La “menuda” tarea de responder eficientemente al anterior reto se ve gravemente complicada por la velocidad con la cual se mueven los adelantos en la industria del *software*, por unos Tratados internacionales aplicados diplomáticamente sin rigor por los Estados partes y por el seductor pero perjudicial discurso de los promotores del *software* libre, lo cual abordaremos en los capítulos II y IV, respectivamente.

Por lo pronto, hay que aseverar que el efecto jurídico, económico y político del *cloning* es la extensión de una externalidad negativa consistente en la desarticulación del Derecho de Autor y el Sistema de Patentes, lo cual sintetiza el déficit de protección jurídica¹³⁹ que reclaman los afectados por la presencia de *free riders* en la industria y justifica al menos una legislación especial¹⁴⁰, nacional y/o internacional, que nivele los intereses de los involucrados en esta

¹³⁷ (i) (Landes & Posner, 1989) Pág. 326: “*The creator will make copies up to the point where the marginal cost of one more copy equals its expected marginal revenue. The resulting difference between price and marginal cost, summed over the number of copies sold, will generate revenues to offset the cost of expression. Since the decision to create the work must be made before the demand for copies is known, the work will be created only if the difference between expected revenues and the cost of making copies equals or exceeds the cost of expression*”, (ii) (Merges, Menell, & Lemley, 2010) Pág. 13.

¹³⁸ (Samuelson & Scotchmer, The law and economics of reverse engineering, 2002) Pág. 1623.

¹³⁹ (i) (Measures to enhance international cooperation in the field of legal protection of computer software, 1979) Pág. 3: “*Although it seems to be generally accepted that computer software could enjoy copyright protection, provided that the conditions of copyright law are fulfilled, such protection does not – or at least does not always – meet the specific need for protection of computer software. In particular a gap may exist with respect to the – probably most important – aspect of protection, namely, the protection against use of computer software in the control of the operations of a computer (section 5(vi) of the Model Provisions). Copyright laws normally do not grant protection against any use of literary or artistic work. Copyright laws normally protect only against reproduction and public performance and communication to the public... One of the main purposes of the Model Provisions is precisely to fill this gap*”, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2355.

¹⁴⁰ Un buen ejemplo de esto lo es la Ley de Protección de Programas de Computador, expedida en 1986 en la República de Corea (Reformada en los años 1998 y 2000).

Industria.

2.5. Efecto económico de la *estandarización del software*.

El éxito de un *software*, sobre todo si éste es un sistema operativo, puede ser tal que se convierta en un estándar para el mercado, ora porque su eficiencia técnica es inigualable, ora porque la capacitación de usuarios en productos sustitutos es económicamente ineficiente¹⁴¹. El problema de la *estandarización del software* consiste en que la firma innovadora estará en posición de controlar para su beneficio una barrera de entrada para nuevos oferentes¹⁴².

Ambos supuestos fácticos han ocurrido en la industria, ambos sucesos han sido *ratio decidendi* en litigios y ambos sucesos han sido provocados por la protección del *software* vía

¹⁴¹ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2375, (ii) (Swinson, 1991) Págs. 205 – 207, (iii) (Gómez Segade, 2001) Pág. 861, (iv) Opinando sobre el caso (Lotus Development Corp. v. Borland International Inc., 1996): (Carranza Torres, 2008) Págs. 247 – 249, (v) (Dam, 1995) Pág. 351: “*Finally, we can conclude that these three broader ideas (compatibility, de facto standards, and network externalities) turn out to be not only closely related but essentially the same when used to condemn copyright protection. In all three cases, the argument boils down to the proposition that the economy would be better off if follow-on firms could simply copy first-generation innovators' software products where it would be costly for existing users to switch to a new product. All that differs is the verbal justification, and in fact all three arguments apply to roughly the same situation-where the first-generation firm has a leading position in the particular software market and a follow-on competitor seeks to market a replacement product*”, (vi) (Posner, Antitrust law, 2001) Págs. 247 - 248: “[E]conomies of scale in consumption, becomes troublesome from the standpoint of competition. Economies of scale in manufacture are familiar; up to a point, the longer the production run the lower the average cost, as fixed costs gets spread over a larger and larger output. Economies of scale in consumption describe the situation in which the larger the firm's output is, up to some point, the more valuable that output is to its consumers... Literal networking of sharing to one side, computer programs tend to be more valuable the more people use them because training, support by information-technology personnel, and standardization of equipment and procedures are facilitated. It is the same reason that the typewriter keyboard is standardized”.

¹⁴² (i) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1342, (ii) (Nguyen , Gomulkiewicz , & Conway , 2012) Pag. 559: “*The leading case on copyright misuse involving a software license is Lasercomb America, Inc. v. Reynolds, in which the misuse arose from the use of a standard from license purporting to restrict licensees from developing competing products for a period of 99 years*”, (iii) Dando a entender que el argumento político en procura de una excepción de interoperabilidad y en búsqueda de productos sustitutos es inmediato resultado del estándar *de facto* que alcanza un determinado *software*: (Dam, 1995) Pág. 326, 328 y 358, (iv) Una explicación de los efectos de la estandarización en los mercados con externalidades de redes, y el valor que alcanza el producto una vez se ha concertado su estandarización: (Stern, 2018) Pág. 206.

Derechos de Autor (y en parte por las patentes¹⁴³), porque si algo caracteriza la protección del *software* vía Derechos de Autor es que genera más incertidumbre sobre el alcance de los derechos concedidos a la creación, que contornos claros sobre los cuales entrar en negociación con los titulares¹⁴⁴, todo potenciado por un plazo de protección inadecuadamente extenso para el producto en estudio.

En efecto, las firmas exitosas en sus productos están en capacidad de discriminar contractualmente a sus clientes, de tal forma que permitirán bajo ciertos términos y condiciones la compatibilidad con otros que le hagan más versátiles, creando una rosca de firmas que ayudan a dar contenido¹⁴⁵ a sus creaciones, zafando de la rosca a otros con los cuales no se quiere contratar. Como suele suceder, el que está fuera de la rosca buscará clonar la tecnología, después de haber intentado entrar en ella, y he ahí cuando inician los litigios que invocan a la par normas de propiedad intelectual y de protección a la libre competencia¹⁴⁶.

¹⁴³ (Unwired Planet v Huawei, 2017)

¹⁴⁴ (i) (Dam, 1995) Pág. 365: “*If the original innovator can enjoin the copying, then the property rights are clear, and the parties will have an incentive to deal with each other through contract*”, (ii) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1363.

¹⁴⁵ El modelo inicial de negocio, en cuanto a creación de *software* concierne, no contemplaba la provisión de contenido a plataformas, por lo que la competencia entre las firmas innovadores, sucesoras (“*follow-on*”) y competidoras (“*me too*”) consistía únicamente en crear productos sustitutos, y no complementarios. (i) (Dam, 1995) Pág. 328, (ii) (Nguyen, Gomulkiewicz, & Conway, 2012) Pág. 509.

¹⁴⁶ Rehusarse a contratar con terceros licencias sobre materia protegida por un derecho de propiedad intelectual es, en términos generales, una conducta legítima por parte del titular del derecho. El problema se presenta cuando de por medio se da un estándar *de facto* o una externalidad de redes que bloquea el acceso a la plataforma. Sobre el particular: (i) (Burtis & Kobayashi, 2001) Págs. 153 y 157, (ii) (Waelbroeck & Frignani, 1998) Págs. 1069 y 1072, (iii) (Graef, 2014) Pág. 10: “[C]ase law had established that a refusal to grant a license in itself cannot lead to liability, since the possibility of the owner of the protected property preventing third parties from using the underlying invention forms is the very subject matter of the exclusive right. However, the exercise of the exclusive right by the right holder may, in exceptional circumstances, involve abusive conduct”. (iv) Sobre condiciones discriminatorias para prestaciones equivalentes: (Jiménez Valderrama, 2019) Pág. 55

Hay voces que propenden por una *estandarización* de ciertos aspectos relacionados con el *software* por vía legislativa¹⁴⁷, en favor de los usuarios profesionales y a costa de la firma innovadora, pues se privilegia la reducción de costos de reentrenamiento para usuario finales antes que el premio a la genialidad¹⁴⁸ del innovador; lo cual no es Pareto eficiente, ya que perjudica la posición anterior del innovador. Además, no debe perderse de vista que “la” *estandarización* es función gremial de interés público¹⁴⁹, por la cual se agotan márgenes para la creatividad¹⁵⁰ entre oferentes del mercado relevante y en la cual media una compensación económica entre los actores del sector, en mejor trámite de concertación que la verticalidad de las leyes¹⁵¹, expuesta a la captura del legislador¹⁵².

¹⁴⁷ (Swinson, 1991) Págs. 205 y 208.

¹⁴⁸ (Landes & Posner, 1989) Pág. 352: “*The mere fact that a particular set of symbols has become the industry standard is a tribute to the expressive skills of the particular manufacturer and should not be deemed to convert expression into idea*”.

¹⁴⁹ (i) (Stern, 2018) Pág. 116: “*Product interoperability ensures that products from a variety of suppliers will work together efficiently, thus reducing costs for consumers and producers, making products more valuable, and promoting innovation both in and around the standard*”, (ii) (Microsoft Corp. v. Motorola Inc, 2015): “*A RAND commitment must be construed in the public interest because it is crafted for the public interest*”.

¹⁵⁰ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2402.

¹⁵¹ (i) (Contreras, 2016) Pág. 857: “*Standards may be developed in a variety of settings. Some health, safety, and environmental standards are developed by governmental agencies. Most interoperability standards, however, are developed in the private sector. Individual firms may develop proprietary technologies that, through broad market adoption, become de facto standards (e.g., Adobe's ‘portable document format’ (PDF)). In several well-known cases competing firms have engaged in commercial ‘standards wars’ to determine which of their proprietary formats will prevail in the market. Over the past two decades, however, most interoperability standards have been developed by groups of market participants that collaborate within voluntary associations known as standards-development organizations (SDOs). The standards produced within these organizations are often referred to as ‘voluntary consensus standards’, as they are developed through consensus-based collaborative processes, and there is no requirement that participants use the resulting standards*”, (ii) Sobre las peripecias de la jurisprudencia para hacer coercibles las programáticas declaraciones de quienes participan en “la” estandarización: (Stern, 2018) Pág. 133: “*The US courts have held that FRAND commitments are legally binding third-party beneficiary contracts enforceable in favor of firms that implement the standard to which the commitment relates*”.

¹⁵² Con todo, y sin desconocer la competencia que tiene el sistema judicial para dirimir controversias planteadas como incumplimientos contractuales, abusos de posición dominante o similares, consideramos que decisiones como el resuelto caso (Unwired Planet v Huawei, 2017) a efectos de concretar una licencia FRAND, o la disputa entre Apple y Qualcomm (<https://www.ft.com/content/70c5acfa-7c83-11e9-81d2-f785092ab560>; <https://www.theverge.com/2019/4/16/18410985/apple-qualcomm-settle-royalty-dispute-patent-licensing-terms-high-fees>), ciertamente ponen de relieve que la respuesta al precio “justo” por el cual remunerar al titular de una patente esencial que ha sido estandarizada (“SEP” por su siglas en inglés), o si son los consumidores quienes deben recibir todo el beneficio de una estandarización, merece un debate parlamentario que resulte con una norma supletiva, antes que decisiones judiciales “al ojo” o *ex aequo et bono* como parece la adoptada en el caso citado.

Sin desconocer las presiones que sobre la libre competencia genera la *estandarización* y la protagonista externalidad por redes en la industria del *software*¹⁵³, al desembocar en instituciones bien averiguadas por el derecho de la competencia¹⁵⁴, consideramos que en esos cuerpos normativos se encuentran las respuestas apropiadas, sin necesidad de proponer soluciones *ad hoc* que desenfocan el objeto de la propiedad intelectual¹⁵⁵.

Esto es, la protección de la propiedad intelectual vertida en el *software* debe incentivar la innovación en dicha industria, sin consideración al estímulo adicional que provoca su *estandarización* concertada¹⁵⁶, *jure et de jure* o de *facto*, y en este mismo marco propondremos al final del trabajo una mejor protección para el *software*; la cual, en todo caso, responde a la barrera de entrada generada por la *estandarización* que patrocina un término de protección exageradamente largo, con una combinación de derechos exclusivos y de remuneración, dentro de un acortado término de protección para el *software*, si se le compara con el ofrecido por el Derecho de Autor, el Sistema de Patentes y la Protección a Secretos Empresariales.

¹⁵³ (i) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1342, (ii) Advirtiendo sobre los agudizantes efectos sobre la externalidad de redes, si se conceden fuertes derechos de propiedad intelectual al *software*: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2375.

¹⁵⁴ Prácticamente reduciendo este tema, en la “nueva economía”, al tratamiento que reciben los monopolios, y sugiriendo que la carrera entre firma por alcanzar el estado de monopolistas, genera competencia en innovación y en consecuencia, no es muy perjudicial que digamos: (Posner, Antitrust law, 2001) Págs. 248 – 249.

¹⁵⁵ (i) (Dam, 1995) Pág. 351 y 359, (ii) (Waelbroeck & Frignani, 1998) Págs. 951 – 952, (iii) (Swinson, 1991) Pág. 170, (iv) (Burtis & Kobayashi, 2001) Pág. 156, (v) (Stern, 2018) Pág. 215: “*That is not a patent law problem, however, but rather a competition law problem, and one to be addressed separately*”.

¹⁵⁶ (Stern, 2018) Pág. 206: “[S]tandardization causes interoperability, which causes network effects, and the network then generates value that increases greatly with the number of users joining the interoperable network. That value is vastly greater than the sum of the values that would exist if the individual elements in the network were isolated from one another in their original form instead of being interconnected or interconnectable so that they can interoperate. That value is reflected in the surplus between the ex-ante value and ex-post value of a SEP”.

Capítulo II. Reaccionando al déficit de protección para el *software*: Mitos y realidades sobre las patentes de *software*

Si tenemos en cuenta que todo *software* es una combinación de procesos algorítmicos ejecutados por un computador¹⁵⁷, el modelo de protección del *software* mediante las patentes de invención o las patentes de procedimiento provoca grandes interrogantes respecto al cumplimiento de los requisitos clásicos de novedad, nivel inventivo y aplicación industrial que deben cumplir todas las patentes.

En efecto, las dudas sobre el cumplimiento de la novedad, para conceder una patente de *software*, derivan del irrefutable hecho de ser el *software* el producto, como ningún otro bien protegible por la propiedad intelectual, de la agregación explícita del estado de la técnica anterior¹⁵⁸; lo cual, nos conecta directamente con la obviedad y/o inclusión de la idea *en sí*, proscrita por el nivel inventivo¹⁵⁹.

Igualmente difícil es conciliar la aplicación industrial que deben cumplir las patentes de *software*, ya que si el efecto técnico que provoca la ejecución del *software* se reduce a los impulsos eléctricos¹⁶⁰ que se transmiten al interior del *hardware*, entonces el *software* no tiene aplicación industrial¹⁶¹.

¹⁵⁷ Definiendo lo que es un algoritmo, y cómo opera: (i) (Chisum, 1986) Págs. 975 y 976, (ii) (Swinson, 1991) Págs. 150 y 153, (iii) (Baylos Corroza, 2009) Pág. 761.

¹⁵⁸ (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pag. 302: “*The nature of software is one of the reason why defining derivative works for software is difficult. Most software is made up of numerous individual programs and files that work together to produce the results the end-user experiences. Software programs also make use of ‘libraries’. A library is a collection of subprograms used to develop software. ‘Libraries’ are distinguished from executables in that they are not independent program; rather, they are ‘helper’ code that provides services to some other independent program*”.

¹⁵⁹ En este sentido: (i) (Lexmark International Inc. v. Static Control Components Inc., 2004), (ii) (Parker v. Flook, 1978), (iii) (Gottschalk v. Benson, 1972), (iv) (Cohen, Reverse engineering and the rise of electronic vigilantism: Intellectual property implications of "lock-out" programs, 1994-1995) Pag. 1169, (v) (Peña Valenzuela, 2013) Pág. 43, (vi) (Preinfalk Lavagni, 2010) Pág. 149.

¹⁶⁰ (Botana Agra, 2001) Págs. 167 – 168.

¹⁶¹ (i) (Gómez Segade, 2001) Pág. 901, (ii) (Guerrero Gaitán, 2006-2007) Págs. 123, 141 y 143, (iii) (Botana Agra, 2001) Pág. 165: “[P]ara no romper con la exigencia del efecto técnico en la invención, se atribuye este efecto a los

Con todo, patentar el *software* como un procedimiento¹⁶², el *firmware*¹⁶³ como un invento, o el *software* como tal - pues el *software* está compuesto de algoritmos para solucionar un problema técnico¹⁶⁴ -, es una realidad en las oficina de patentes de Estados Unidos de América¹⁶⁵, en Europa y Japón (reticentes al principio), con lo cual se concede un nivel fuerte de protección¹⁶⁶; pues conceder patentes de invención o procedimiento al *software*, implicaría el encarecimiento subsiguiente de todo avance en la industria, si se tiene por cierta la evidencia empírica que denuncia la reticencia de los titulares de las patentes de *software* para conceder licencias a sus competidores, con el fin de que estos elaboren productos basados en aquél¹⁶⁷.

programas cuya función inmediata es hacer posible que la máquina computadora realice alguna actividad práctica”, (iv) (Chisum, 1986) Pág. 967: “[T]he appropriate inquiry with ‘software’ inventions should be on whether the claimed subject matter produces specific, active, useful results and not simply communication with humans”, (v) (Charfoos, 2002) Pag. 5: “There were two schools of thought on whether computer software could be patented. The first set of nations provided that computer software could only be patented if it produced some kind of real world mechanical result. The second set of countries allowed software to be patented per se as long as it satisfied the traditional definition and requirements for patents. The Paris Convention did nothing to clarify this dispute, thus leaving the issue unresolved”, (vi) Considerando que la dificultad en el patentamiento del *software* deriva de la falta de “manipulación material”, como elemento fundamental de un objeto patentable: (Peña Valenzuela, 2013) Pág. 29, (vii) En el mismo sentido anterior: (Baylos Corroza, 2009) Pág. 762, (viii) (Hong & Cheng-ssu, 1999) Págs. 257 y 267.

¹⁶² (Botana Agra, 2001) Pág. 174: “[Refiriéndose al criterio aplicado por la oficina de patentes de EE.UU. sobre el patentamiento de métodos de hacer negocios] [T]ambién en las Directivas para el Examen de las Computer Related Inventions (edición 1996) se incluyó el siguiente párrafo: «el personal de la Oficina ha tenido dificultades en valorar correctamente las reivindicaciones relativas a métodos de hacer negocios. Tales reivindicaciones no serán catalogadas como methods of doing business. En su lugar, tales reivindicaciones han de ser tratadas como cualesquiera otras reivindicaciones de procedimiento”.

¹⁶³ Conocido como el *software* que se encuentra insertado inescindiblemente al *hardware* que controla.

¹⁶⁴ (i) (Chisum, 1986) Pág. 976, (ii) (Swinson, 1991) Pág. 151, (iii) (Calle D'Alleman, 2012) Pág. 263, (iv) (Carranza Torres, 2008) Pág. 212, (v) (Botana Agra, 2001) Pág. 165, (vi) (Salas Pasuy, 2016) Pág. 470 - 472, (vii) (Aguirre Soriano, 2014) Pág. 96, (viii) (Baylos Corroza, 2009) Pág. 1105: “Reduciéndola a su núcleo esencial, cabría decir que la invención es la solución de un problema técnico aplicable a la industria, y que proporciona la posibilidad de obtener un cierto resultado útil”.

¹⁶⁵ Particularmente, a partir del caso (Diamond v Diehr, 1981). (i) (Guerrero Gaitán, 2006-2007) Pág. 135 y 136, (ii) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 4 y 8, (iii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2345, (iv) (Bessen & Hunt, The software patent experiment, 2004) Págs. 6 y 7. En Colombia no se podría patentar el *software* como tal: (v) (Salas Pasuy, 2016) Pág. 480, (vi) (Guía para examen de solicitudes de patente y modelo de utilidad, 2012) Pág. 49.

¹⁶⁶ (Swinson, 1991) Pág. 146

¹⁶⁷ Algo similar se debatió en el sonado caso por monopolio adelantado por la Comisión Europea contra *Microsoft Inc.*, pues esta empresa, abusando de su posición dominante, evitaba la complementación de su sistema operativo con reproductores de multimedia distintos al producido por ella misma, mediante el cobro prohibitivo de licencias a desarrolladores independientes. Para consultar al respecto: http://ec.europa.eu/index_es.htm. Sobre la reticencia de los titulares de patentes en conceder licencias, puede leerse: (i) (McGee, 1966), (ii) (de la Cruz Camargo, 2008), (iii) Acerca de la negativa a contratar con proveedores de servicios *post-venta* de productos patentados: (Burtis & Kobayashi, 2001).

Pero la realidad de las patentes de *software* supera la ficción con la cual se basan muchos textos académicos, motivo por el que ahondaremos en las vicisitudes aparejadas a la protección del *software* vía el Sistema de Patentes, primeramente por el soporte jurídico, beneficios generales y particulares que las sostienen (1), y luego considerando las consecuencias de concederlas (2).

1. ¿Por qué hay patentes de *software*?

Iniciando por una precisión terminológica, y en aislamiento doctrinal con mis compatriotas, consideramos que se comprende mejor la problemática en torno al modelo adecuado de protección para el *software* si a las invenciones que lo contienen se les llama con franqueza “patentes de *software*”, pues son las honduras¹⁶⁸ que implica la concesión de patentes sobre estos bienes las que se camuflan con la expresión “invenciones implementadas por computador”, o similares.

En efecto, detrás de la eufemística expresión “invenciones implementadas por computador¹⁶⁹” - o similares - se esconden algoritmos, considerados teóricamente¹⁷⁰ “materia no patentable”¹⁷¹, pero cuya protección hoy en día se concede, al menos de forma indirecta, gracias

¹⁶⁸ (Salas Pasuy, 2016) Pág. 470: “[E]n cuanto al software o programa de ordenador, estableció que las ‘invenciones que involucren un software pueden ser protegidas como invenciones implementadas por computador. Entonces, tendrá dicho carácter las reivindicaciones que involucran computadores, redes de computadores u otros aparatos que son ejecutables con el mismo, pueden adoptar la forma de un método de operación como un aparato, o pueden versar sobre la configuración o el medio físico que almacena el programa. En consecuencia, reivindicaciones de producto del software como ‘datos de almacenamiento’ ‘medios de almacenamiento’, ‘dispositivos de lectura’ o ‘señales’ pueden ser patentable por tratarse de invenciones implementadas por ordenador”.

¹⁶⁹ (i) Llamando “jerigonza”, al lenguaje empleado en las reivindicaciones hecha para la patente “click” concedida a Amazon: (Uribe Arbeláez, 2005) Pág. 199, (ii) Sobre los alcances de estos términos: (Salas Pasuy, 2016) Pág. 442.

¹⁷⁰ (i) (Chisum, 1986) Pag. 960 – 961, (ii) (Botana Agra, 2001) Pág. 161.

¹⁷¹ (i) Vale la pena mencionar que hubo un connato de modificación al numeral 2 del artículo 52 del Tratado sobre la

a la, en nuestro concepto, insubordinada, oportunista y nada rigurosa interpretación del artículo 27 - 1 de ADPIC.

Esto es así, porque por mal enfocado que haya sido desde el principio la protección del *software*¹⁷², la redacción del artículo 10 – 1 de ADPIC¹⁷³ no permite asomo de duda respecto del régimen asignado a la protección del *software*, luego la interpretación que se le ha dado al artículo 27 - 1 de ADPIC es asistemática, si bien soporta jurídicamente las patentes de *software*, en alivio parcial a su déficit de protección en cuanto al *cloning* se refiere. Lo anterior se ha conjugado con argumentos convergentes tales como: (i) El Sistema de Patentes solamente excluye el patentamiento del *software* “como tal”¹⁷⁴, (ii) Las patentes deben estar disponibles

Patente Europea, por el cual se eliminaba al *software* de la materia no-patentable, pero tal modificación, aunque defendida por algunos críticos, no fue finalmente introducida. Al respecto: (Castán Pérez-Gómez, 1999) Pág. 120, (ii) Hoy se admiten los algoritmos “como tal”, mientras cumpla los otros criterios de patentabilidad: (Salas Pasuy, 2016) Pág. 462, (iii) El caso (Alice Corp. v. CLS Bank International, 2014) hizo “resucitar” la revisión del presupuesto de “materia patentable” para el caso de las patentes de *software*: (Gugliuzza & Lemley, 2018) Pág. 769.

¹⁷² (Carranza Torres, 2008) Pág. 223: “*Con la mayoría de las otras obras del intelecto humano el software tiene tan pocas cosas en común que apenas se saludarían si se cruzaran en las calles del barrio que comparten. El software es ‘rarito’ dentro del barrio de los derechos de autor, y el derecho necesita ubicarlo en otro lado*”. En este libro se le critica enormemente a la ubicación del *software* dentro del Derecho de Autor, lo cual se considera en repetidas veces como un “*accidente histórico*” (El capítulo VIII de este texto, que inicia en la página 205, se titula: “*El accidente histórico de la protección del software*”).

¹⁷³ (i) “*Artículo 10: Programas de ordenador y compilaciones de datos. 1. Los programas de ordenador, sean programas fuente o programas objeto, serán protegidos como obras literarias en virtud del Convenio de Berna (1971)*”, (ii) Como lo narra (Charfoos, 2002) Pág. 8, tal parece que la preocupación de los redactores de esta disposición era impedir un monopolio sobre los algoritmos.

¹⁷⁴ (i) (Dam, 1995) Pag. 368, (ii) (Botana Agra, 2001) Pág. 164, (iii) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 20. Con todo, Australia y Japón han expresado que sí patentarían *software* como tales: (iv) (Charfoos, 2002) Pág. 16, (v) (Salas Pasuy, 2016) Pág. 459. Estos autores también indican que Estados Unidos de América ha manifestado que patentará *software* “como tal” (Entiéndase: “líneas de código”), pero aún están por verse los efectos sobre esta manifestación, después de la decisión en (Alice Corp. v. CLS Bank International, 2014), en el que se revisa el concepto de “materia patentable”, recogiendo posturas del emblemático caso (Diamond v Diehr, 1981), y de los trascendentes y recientes asuntos (Bilski v. Kappos, 2010) y (Mayo v. Prometheus, 2012). (vi) Sobre la disminución en la concesión de patentes de *software*, a raíz del caso (Alice Corp. v. CLS Bank International, 2014), aunque en realidad tal providencia aún no tiene característica de cosa juzgada: (Gugliuzza & Lemley, 2018) Pág. 809.

para todas las creaciones en todos los campos de la tecnología¹⁷⁵, y; (iii) Los artículos 27 – 2 y 27- 3 de ADPIC NO excluyen expresamente las patentes para el *software*¹⁷⁶.

En este contexto se han abierto paso¹⁷⁷ las reivindicaciones en patentes de invención o de procedimiento que implícitamente contienen un *software*, las cuales no abarcan todos los componentes y algoritmos que lo componen¹⁷⁸; pero gracias a la mejor protección que ofrecen en comparación al Derecho de Autor¹⁷⁹, han traído consigo beneficios sociales de consideración, que a continuación se destacan.

¹⁷⁵ (Córdoba Marentes, 2006) Pág. 139.

¹⁷⁶ (i) (Botana Agra, 2001) Pág. 169, (ii) (Charfoos, 2002) Pág. 14, (iii) (Carranza Torres, 2008) Pág. 216. (iv) Es indicativo también cómo en su momento la OMPI dejó a la interpretación (Entiéndase: “a la judicatura”) la viabilidad de la protección concurrente del Derecho de Autor y el Sistema de Patentes para el *software*: (Committee of Experts on the Legal Protection of Computer Software Second Session, 1983) Pár. 11.

¹⁷⁷ El camino hacia el registro de patentes de *software* no fue expedito: (i) (Botana Agra, 2001) Pág. 164, (ii) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 7, (iii) (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 139, (iv) En Colombia se registra en la doctrina una posición en contra de las patentes de *software*: (Córdoba Marentes, 2006) Pág. 141: “*Tradicionalmente los autores colombianos han rechazado la posibilidad de que un programa de computador pueda ser patentado. Uno de los primeros en pronunciarse sobre el tema, Manuel Pachón, se refería a la necesidad de que el producto a patentar fuera tangible, lo que en el caso del software, por supuesto, no era posible. Ni siquiera en el evento en que tal programa estuviera fijado en un soporte físico podía considerarse como un producto patentable, por cuanto el soporte sería un objeto conocido y, por lo tanto, no patentable*”. (v) Pero, contemporáneamente hay una apertura a estas, debido al giro que ha hecho la Superintendencia de Industria y Comercio al respecto (Quizá siguiendo la “penúltima” moda): (Salas Pasuy, 2016) Pág. 480, (vi) (Guía para examen de solicitudes de patente y modelo de utilidad, 2012) Pág. 49: “*El examinador deberá tener en cuenta que los programas de computador o de soporte lógico son las instrucciones que necesita una máquina para conseguir un resultado. En principio, no se consideran invenciones porque no tienen carácter técnico al ser considerados en sí mismos, como ocurre con las obras literarias. No obstante, cuando la solicitud no verse sobre un programa de ordenador en sí mismo o como tal, el examinador deberá proceder a adelantar el estudio de patentabilidad según los requisitos de novedad, nivel inventivo y aplicación industrial en la forma prevista en esta Guía*”.

¹⁷⁸ (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 26.

¹⁷⁹ (i) (Calle D'Alleman, 2012) Pág 69, (ii) (Botana Agra, 2001) Pág. 169, (iii) (Salas Pasuy, 2016) Pág. 439.

1.1. Beneficios generales de conceder patentes de *software*.

El beneficio general e inmediato de conceder patentes de *software* es, como en todas las demás patentes, el acceso a información útil, debidamente clasificada e incorporada al estado de la técnica. Sin duda, el hecho de que muchas patentes no expongan fielmente todo lo que el invento trae consigo reduce en parte el noble propósito de dar a conocer las creaciones de otros, pero en el caso del *software* podemos expresar, sin pecar de inocentes, que su estructura algorítmica permite conocer muy de cerca todo lo que el invento es, y esto reduce el desgaste innecesario de recursos en investigaciones pares, ocultas bajo la Protección a Secretos Empresariales¹⁸⁰.

Adicionalmente, el beneficio general derivado del cumplimiento estricto del necesario nivel inventivo en las patentes, aunque mal o mediocrementemente evaluado según críticas registradas en el siguiente punto de este capítulo, supone un elevado *standard* en comparación con la protección actual bajo el sistema del Derecho de Autor, que es indiferente al mérito de la obra protegida¹⁸¹.

Como efecto transversal a toda la economía, la asignación precisa de un derecho sobre la invención, en clásica aplicación del teorema de Coase¹⁸², reduce costos de transacción para

¹⁸⁰ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2384.

¹⁸¹ (i) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1350, (ii) (Nimmer & Krauthaus, 1987) Pág. 742.

¹⁸² (i) (Posner, Economic analysis of Law, 2011) Pág. 10: “[I]f transactions are costless, the initial assignment of a property right will not affect the ultimate use of the property”, (ii) (Aguirre Soriano, 2014) Pág. 103. No estamos de acuerdo con lo indicado por el autor de este artículo en las páginas 109 a 112, pues consideramos que la patentabilidad del *software* DISMINUYE costos de transacción, al definirse un mejor contorno de los derechos concedidos al titular, y NO eleva la especificación de los bienes, en términos de la infraestructura y recurso humano requeridos para el

ceder¹⁸³, licenciar y hacer respetar la titularidad del invento¹⁸⁴; cuestión que no es característica del modelo actual vía el Derechos de Autor, en el que derechos morales, titularidad originaria en persona naturales, limitaciones y excepciones diseñadas para obras no-utilitarias elevan enormemente los costos de transacción¹⁸⁵. Esto es particularmente crítico para emprendedores en busca de capital¹⁸⁶, pues podrían atraer inversiones u obtener créditos, apalancados por el *software* como activo intangible que es; pero la falta de contornos claros a los derechos sobre su desarrollo les limita las fuentes de recursos, en razón a la aversión de los acreedores al riesgo que representa para su garantía una titularidad difusa en estos productos.

En fin, aunque extremadamente largo para el *software*, una protección por 20 años es preferible a una protección mínima, en el mundo, de 50 años *post mortem auctoris*¹⁸⁷ - De 80 años *post mortem auctoris* según nuestra legislación -, pues tales términos de protección

desarrollo de *software*, pues la evidencia empírica enseña que el primero de estos factores es relativamente abundante, el segundo no tanto (aún a escala mundial) pero accesible. Respetuosamente considero que el autor presenta su tesis en contra del patentamiento del *software* con base en los costos de transacción asociados a los secretos empresariales, sin empatía por los procesos empresariales antecedentes a las patentes, ni los beneficios generales de estas. Con nosotros: (iii) (Smith & Mann, 2004) Pág. 256, (iv) (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 192.

¹⁸³ (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 139.

¹⁸⁴ (Posner, Economic analysis of Law, 2011) Pág. 53.

¹⁸⁵ (Dam, 1995) Pag. 368.

¹⁸⁶ (i) (Smith & Mann, 2004) Pag. 262, (ii) (Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia, 2014) Págs. 10 y 49, (iii) (Conpes 3582, 2009) Pág. 14, (iv) Calzan muy bien con esto, los próximos bonos naranja: <https://id.presidencia.gov.co/Paginas/prensa/2018/180818-En-noviembre-se-lanzaran-los-Bonos-Naranja-para-impulsar-a-emprendedores-anuncio-el-Ministro-de-Comercio.aspx>; <https://www.larepublica.co/economia/ministerio-de-comercio-inyectara-400000-millones-por-medio-de-bonos-naranjas-2769779>. Los artículos 162 y 165 de la Ley 1955/19 (Plan Nacional de Desarrollo 2018 – 2022: “Pacto por Colombia, pacto por la equidad”) establecen programáticamente la emisión de títulos a riesgo o crédito para pequeñas y medianas empresas, por un valor de 584.000 UVT, (v) (García, 2019) “*En Páramo hemos venido trabajando en el acceso a crédito de la línea ‘exprime la naranja’ y al principio fue muy difícil por las razones expuestas pero afortunadamente en las últimas semanas hemos contado con acompañamiento del banco y esperamos sacarlo adelante. Sin embargo, me parece importante resaltar que hemos tenido estas dificultades a pesar de ser una empresa con un área financiera robusta y con una experiencia sólida en temas de acceso a capital, así que podemos imaginar lo difícil que es para emprendedores nuevos y de poca experiencia*”.

¹⁸⁷ (i) (Smith & Mann, 2004) Págs. 257 – 258, (ii) (Pérez Gómez-Tétré, Perspectivas económicas, 2012) Pág. 409: “*El costo social es directamente proporcional a la duración de la patente. Así, entre más largo sea el tiempo de protección de la patente, mayor será el costo social y viceversa*”.

propician abusos de posición dominante, cuando no verdaderos monopolios, en el mercado de *software* de sistema operativo o *software* aplicativo¹⁸⁸, y en mercados paralelos como los relacionados con servicios de mantenimiento, actualización y capacitación de usuarios¹⁸⁹.

1.2. Beneficios particulares de conceder patentes de *software*.

Natural y obvio beneficio particular para quien obtiene una patente de *software* es que acaba en buena medida con los *free riders* bajo apremio de infracción de derechos y/o simple respeto de la propiedad ajena. En otras palabras, las patentes de *software* provocan un “freno en seco” al problema de *cloning*, gracias a que las reivindicaciones abarcan procedimientos y resultados similares¹⁹⁰, y esto es la razón principal por la cual se procura el patentamiento de *software*¹⁹¹ entre quienes eligen esta forma de protección.

¹⁸⁸ (i) Una descripción de las funcionalidades de cada *software*, ensayando una definición para cada uno, en: (Nimmer & Krauthaus, 1987) Pág. 734, (ii) También: <https://en.wikipedia.org/wiki/Software>, (iii) (Uribe Arbeláez, 2005) Pág. 197, (iv) (Peña Valenzuela, 2013) Págs. 60 – 73, (iv) (Soler Matute, 2004) Págs. 31 – 36.

¹⁸⁹ (i) (Zuleta Londoño, 2005) Pág. 541, (ii) (Uribe Arbeláez, 2005) Pág. 207, (iii) (Klein, 1993) Pág. 49.

¹⁹⁰ (i) (Menell, Tailoring legal protection for computer software, 1987) Págs. 1347 y 1362, (ii) (Charfoos, 2002) Págs. 14 y 19, (iii) (Botana Agra, 2001) Pág. 169, (iv) Basados en jurisprudencia dictada sobre el derecho de patentes, pero con muchas semejanzas a lo descrito aquí: (Hong & Cheng-ssu, 1999) Pág. 293: “*In this case [In regard Air Force Hospital 87333 v. A Technology Co.], the court of first instance used the doctrine of equivalence to decide the patent infringement. The court correctly pointed out that, when applying the doctrine of equivalence, the comparison was not between the alleged infringing product and the patented product, but between the technological features of the alleged infringing product and the essential technological features of the patented technological solution. Therefore, even though the defendant had used a different model of hardware and increased certain functions of the software, the defendant’s acts still constituted infringement*”. (v) No compartimos el criterio según el cual las patentes de *software* solo cubrirían el código fuente, dejando desprotegido el código objeto, expersado por: (Palacio Eastman, 1999) Pág. 212. Lo anterior, desconoce los derechos concedidos al titular sobre el invento y el resultado del procedimiento, en su caso, según los artículos 28 de ADPIC y 52 de la Decisión 486 de la Comisión de la Comunidad Andina de Naciones. En este sentido: (Baylos Corroza, 2009) Pág. 1109: “[E]l inventor se encuentra protegido frente a todo tipo de equivalentes verbales, al formular su invento; y frente a cualquier clase de equivalentes técnicos, al materializarlo en una determinada combinación de elementos sensibles”.

¹⁹¹ (López, 2009) Pág. 16.

Si bien lo anterior puede hacer caer en el facilismo de considerar que así se cohesionan monopolios, como se afirma sin rigor para todos los derechos de propiedad intelectual¹⁹², *prima facie* los requisitos de novedad y nivel inventivo exigidos para obtener una patente incentivan una competitividad creativa¹⁹³, que no solo remunera financiera sino emocionalmente al innovador¹⁹⁴ por la explotación de su irreplicable tecnología, lo cual motiva la proliferación de productos innovadores y comparables, bajo el amparo simultáneo de patentes; luego, no es cierto que las patentes generan monopolios, pese a que se asigna un derecho en el espectro de las reivindicaciones, en clave de funcionalidad como corresponde a la creación, todo lo cual el Derecho de Autor ni es capaz de garantizar, ni logra impulsar.

Por eso, se ha confirmado que las patentes de *software* es la más atractiva forma de atraer crédito o capital de riesgo a las micro y pequeñas empresas de esta industria¹⁹⁵, siendo muchas

¹⁹² (i) (Uribe Arbeláez, 2005) Pág. 202: “*Dadas las especiales características de los soportes lógicos, el problema que surge al patentarlos es que privatiza los algoritmos, por lo que nadie podría elaborar un software basado en los conjuntos y operaciones que conforman el programa patentado. Algo así como si el alfabeto fuera monopolio exclusivo de un puñado de escritores. Habría entonces una ‘jungla de patentes’ que emergerían para calificar de usurpadores a los que por sus propios medios y de forma independiente, desarrollen un programa distinto pero con la misma función del patentado*”. Más adelante también: Pág. 205, (ii) Corrigiendo una mala aproximación a los derechos de propiedad intelectual, cuando se les considera monopolios: (Dam, 1995) Págs. 336 – 337: “[I]t makes more sense, and better illuminates intellectual property law, to emphasize the rent seeking rather than the monopoly problem”, (iii) En un sentido similar: (Baylos Corroza, 2009) Pág. 589, (iv) Anotando el error común en considerar los derechos exclusivos otorgado por el Derecho de Autor, como características de un monopolio económico (Pérez Gómez-Tétré, Perspectivas económicas, 2012) Pág. 406: “*Al analizar tanto la definición como las características del monopolio económico, a la luz del derecho de autor, es posible afirmar que no existe un monopolio en el sentido económico, sino que se trata simplemente del ejercicio de un derecho exclusivo en términos jurídicos. Cabe preguntarse: ¿si el derecho de autor no se ejerce dentro de un mercado monopolístico, entonces dentro de qué mercado se ejerce? Profundizando en la teoría económica se llega a la conclusión de que el derecho de autor se desarrolla dentro de la competencia monopolística... La competencia monopolística se caracteriza primero por poseer una pluralidad tanto de productores como de compradores; segundo, existe una oferta de productos diferenciados con la cual surge la posibilidad de substituir los diferentes productos en el mercado; tercero, no existe verdaderamente limitaciones a la entrada al mercado; y cuarto, los productores tienen un cierto grado de control de precios en el mercado*”.

¹⁹³ (i) (Menell, Tailoring legal protection for computer software, 1987) Págs. 1337 – 1338, (ii) Considerando que las patentes de *software* “asfixian” la innovación: (Uribe Arbeláez, 2005) Págs. 214 – 215.

¹⁹⁴ (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 5.

¹⁹⁵ (i) (López, 2009) Pág. 16, (ii) (Aguirre Soriano, 2014) Pág. 93, (iii) Los bonos naranja caben perfectamente dentro de este mercado. (iv) El artículo 162 de la Ley 1955/19 (Plan Nacional de Desarrollo 2018 – 2022: “Pacto por

veces el único activo con el que cuentan, y en ocasiones lo único que les permite competir con las grandes empresas del sector¹⁹⁶.

Sabiendo empíricamente que la creatividad en la industria NO merma con ocasión de las patentes¹⁹⁷, o de un cúmulo de patentes (“*patent-trolls*”), las patentes de *software* en términos generales empujan cualitativa y cuantitativamente adelante a la industria, ya que provocan una competencia en cuanto a la ingeniosidad en la creación de *software* y la atracción de capital a las firmas de la industria.

2. Crítica al otorgamiento de patentes de *software*

Probablemente la mayor crítica que reciben las patentes de *software* tiene que ver con la ligereza con la cual se pueden haber concedido¹⁹⁸, en buena medida debido a la falta de un repositorio de algoritmos que permita establecer el estado del arte para el momento en que fueron concedidas¹⁹⁹, ya que los examinadores de patentes no tienen ninguna experiencia anterior²⁰⁰, su capacitación es insuficiente para negar reivindicaciones muy amplias²⁰¹ o lo es también para identificar el cumplimiento de los requisitos de novedad y nivel inventivo en las reivindicaciones formuladas en las respectivas solicitudes de patente²⁰².

Colombia, pacto por la equidad”) establece programáticamente la emisión de títulos a riesgo o crédito para pequeñas y medianas empresas, por un valor de 584.000 UVT

¹⁹⁶ (Dam, 1995) Pág. 371.

¹⁹⁷ (i) (Swinson, 1991) Pág. 165, (ii) (Paterson, 2013) Pág. 189.

¹⁹⁸ (i) (Uribe Arbeláez, 2005) Pág. 208, (ii) (Dukrok & Hwang, 2010) Pág. 377.

¹⁹⁹ (i) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 42, (ii) (Swinson, 1991) Pág. 161.

²⁰⁰ (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Págs. 12 – 13.

²⁰¹ (i) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 41, (ii) (Swinson, 1991) Pág. 153, (iii) (Bessen, The patent troll crisis is really a software patent crisis, 2013) Pág. 5.

²⁰² (i) (Merges, Menell, & Lemley, 2010) Pág. 169, (ii) (Smith & Mann, 2004). Pág. 258

Sin embargo, por evidente que lo anterior ha sido para las patentes de *software*, en nada difiere de lo que siempre se expresa como mal endémico en las oficinas de patentes del mundo y por ende, no nos detendremos a repasar puntos comunes de patentes en otras industrias²⁰³ sino en lo que particularmente reviste enorme preocupación a propios y extraños en la industria del *software*, y en particular a los seguidores de los movimientos *free software*²⁰⁴ y *open source*²⁰⁵; esto es, cómo asegurar el acceso a la información que arroja la codificación del *software*, a partir de lo cual pasmosamente se crean *clones* u obras derivadas²⁰⁶ (en términos del Derecho de Autor), y qué beneficio trae consigo el otorgamiento de derechos exclusivos sobre algoritmos considerados obvios, por haber sido previamente explotados²⁰⁷.

²⁰³ Pasaremos por alto también lo atinente a la vaguedad y/o excesiva amplitud de las reivindicaciones para las patentes, pues también esto es crítica ordinaria a las solicitudes de patentes para otras industrias. (Dam, 1995) Pág. 369.

²⁰⁴ (i) (Nguyen, Gomulkiewicz, & Conway, 2012) Pág. 523: “*Free software is developed based on certain key principles espoused by the founder of the Free Software Foundation, Richard Stallman. In the context of free software ‘free’ refers to freedom, not price; programmers often charge for free software. Free software means software that comes with certain freedoms, namely, the right to study the source code; run the software for any purpose; change the software in any manner; and distribute the software and any changes. The most prominent license that embodies free software principles is the general public license (GPL)*”, (ii) También: (Carraza Torres, 2004) Págs. 74 -120, (iii) (Uribe Arbeláez, 2005) Págs. 211 – 214.

²⁰⁵ (i) (Nguyen, Gomulkiewicz, & Conway, 2012) Pág. 523: “*The term ‘open source software’ was created to avoid perceived anticommercial connotation of the ‘free software’ label. Open source software is based on several key principles, which are embodied in the Open Source Definition published by an organization known as the Open Source Initiative (OSI)*”, (ii) También: (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 299: “*The open source movement grew out of the free software movement and shares many values and concepts of the free software movement, although open source developers tend to have a more accepting attitude toward developers of proprietary software*”, (iii) En contraste con el “*open source*”: (Peña Valenzuela, 2013) Pág. 66: “[E]l *software* privativo se refiere al programa de computador en donde los usuarios, las empresas y la comunidad virtual que hace uso de él tienen restringidas las facultades de usar, modificar o distribuir el acceso a su código fuente, y no es accesible al público si no se cuenta con una licencia o una autorización previa para ello”, (iv) (Uribe Arbeláez, 2005) Pág. 213: “*A diferencia del software libre, las modificaciones que se le hacen al programa son consideradas como un valor agregado y dan lugar a derechos reservados... Las mejoras quedan en cabeza de quien las efectuó, que por lo mismo, se las apropia*”.

²⁰⁶ (i) (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 302, (ii) (Byungkwon, 1989) Pág. 175, (iii) (Swinson, 1991) Pág. 192, (iv) (Blind Knut & Firedewald, 2005) Págs. 13 – 14, (v) Es muy importante resaltar el caso (Computer Associates Int’l v. Quest Software Inc., 2004), porque en este asunto el tribunal determinó que no es una obra derivada cubierta por la GPL el resultado del empleo de un *software parser*, empleado como analizador de la correcta compilación hecha al momento de crear un *software*.

²⁰⁷ Sobre el particular, haremos referencia al asunto (Jacobsen v. Katzer, 2008).

2.1. Reservas sobre la “materia patentable”, novedad, nivel inventivo y aplicación industrial del *software*.

Al menos desde hace unos veinte años se ha criticado, sin cesar, la falta de fuentes fiables y exhaustivas del estado del arte previo, como “talón de Aquiles” en la concesión de patentes de *software*²⁰⁸, lo cual tiene que ver con el requisito de novedad; pero quizás las críticas más ácidas tienen que ver con la identificación del *software* con el presupuesto de “materia patentable”²⁰⁹ y el rigor en la aplicación del requisito de nivel inventivo²¹⁰; pues, nuevamente resaltamos, el *software* se compone de algoritmos, los cuales pueden describirse como meros pasos mentales para la obtención de resultados²¹¹, máxime cuando la solicitud de invención se ha presentado como de procedimiento.

²⁰⁸ (Cohen, Reverse engineering and the rise of electronic vigilantism: Intellectual property implications of "lock-out" programs, 1994-1995) Pág. 1178.

²⁰⁹ (Alice Corp. v. CLS Bank International, 2014) Págs. 2 – 3: “*Alice’s remaining claims were held non-patentable, although for different, and inconsistent, reasons. As a result, the legal standards that govern whether computer-implemented inventions are eligible for patent protection under section 101 remain entirely unclear and utterly panel dependent*”. Más adelante en esta providencia se especifican las discordantes razones expresadas por los diferentes jueces que asintieron en que las reivindicaciones no eran patentables, unos considerando que eran meras “ideas abstractas”, otras razones más asociadas a la obviedad o falta de nivel inventivo. Es por eso que la Corte Suprema de los Estados Unidos de América se involucra, anunciando una próxima decisión sobre cuál es el contorno de las patentes de *software* que cumplen el pre-requisito de ser “materia patentable”, y los demás clásicos requisitos para su patentamiento.

²¹⁰ (i) (Palacio Eastman, 1999) Pág. 212, (ii) (Cohen, Reverse engineering and the rise of electronic vigilantism: Intellectual property implications of "lock-out" programs, 1994-1995) Pág. 1169, (iii) (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 139, (iv) (Preinfalk Lavagni, 2010) Pág. 149, (v) (Aguirre Soriano, 2014) Pág. 93, (vi) (Measures to enhance international cooperation in the field of legal protection of computer software, 1979) Pág. 2.

²¹¹ (i) (Nimmer & Krauthaus, 1987) Pag. 744: “[I]n *Schlumberger Canada Ltd. v. Commission of Patents a Canadian court denied protection to a system for measuring and analyzing data from oil well bore holes, analogizing this computer system to a mental process. In contrast, the Paris Court of Appeals allowed patent protection to a statistics analysis system applied to geological survey data*”, (ii) (Paterson, 2013) Pág. 176, (iii) (Chisum, 1986) Págs. 968, 980 – 981, (iv) (Alice Corp. v. CLS Bank International, 2014), (v) Señalando que en China los lenguajes de programación y las reglas de computación no son patentables, por ser consideradas pasos mentales: (Hong & Cheng-ssu, 1999) Pág. 257, (vi) Lo mismo en Japón: (Teruo, 1992) Pág. 107.

En nuestro concepto, es equivocado persistir judicialmente en que el *software* es “materia patentable”²¹², según se explicó al inicio de este capítulo, porque corresponde a una incorrecta aplicación de ADPIC y contribuye al galimatías provocado por la concesión de estas. Además, es inevitable el recelo por conceder el título, desde el mismo nombre como patente de “invención”²¹³ o “patente” de procedimiento²¹⁴, a lo que no es un artículo, ni es visible²¹⁵; máxime si sabemos que se integra por algoritmos, generalmente matemáticos²¹⁶, que ahora son hallados, escritos y aplicados directamente por las máquinas²¹⁷.

Ahora bien, el problema de la levedad con la cual se evalúa el requisito de novedad para las patentes de *software*, y probablemente para las patentes en todos los campos industriales, tiene sus días contados con el advenimiento de la tecnología *blockchain*²¹⁸; pues aplicar esta

²¹² Debe advertirse que la providencia citada en el caso (Alice Corp. v. CLS Bank International, 2014) NO define aún el asunto de fondo, pero es previsible que se dicte una compleja providencia con una lista de argumentos jurídicos para sostener a futuro las actuales patentes de *software*, y no perjudicar el mercado.

²¹³ (i) “*apparatus*”: (Dam, 1995) Pág. 366, (ii) Indicando lo que exige una descripción del invento que contiene un *software*: (Hong & Cheng-ssu, 1999) Pág. 263.

²¹⁴ Indicando lo que exige una descripción del procedimiento que contiene un *software*: (Hong & Cheng-ssu, 1999).

²¹⁵ (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág 25: “[S]*oftware patentees generally do not disclose much, if any, detail about their programs, and therefore there is no easy way to figure out what a software patent owner has built except to reverse engineer the program*”.

²¹⁶ (i) (Dam, 1995) Págs. 367 – 368, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2384, (iii) Opinando sobre la decisión en el caso (Gottschalk v. Benson, 1972): (Chisum, 1986) Pág. 977, (iv) Narrando la evolución jurisprudencial que responde a este argumento: (Carranza Torres, 2008) Pág. 215.

²¹⁷ (i) (Gómez Segade, 2001) Pág. 839, (ii) Aunque no es el sentido en que empleamos la palabra “máquina”, algunos consideran que el *software* ES una máquina: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2315 – 2316.

²¹⁸ (i) Una descripción de la tecnología *blockchain* en: (Rocha & Ducasse, 2018) Pág. 52: “*The blockchain database is managed by a peer-to-peer network where all peers store a complete copy of the database. Each record in this database is a block that is linked to the previous one forming a sequence. The blocks are immutable, which promote trust because the records cannot be altered or deleted. Even though blockchain technology is mostly recognized for its cryptocurrencies, it has been used for other applications as well. A very prominent application of blockchain is managing smart contracts. Smart contracts are programs written in Turing-complete languages that run on the blockchain platform. If we follow the analogy that blockchains are like databases, then smart contracts are like stored procedures since they execute procedural programming in the blockchain data. However, a better analogy is to see smart contracts as classes, because they are composed of data attributes and functions. Moreover, a contract can extend another through inheritance just like classes in object-oriented programming*”, (ii) Se han logrado obtener patentes de invención con base en tecnología *blockchain*: <https://es.cointelegraph.com/news/xerox-wins-patent-for-blockchain-system-that-tracks-revisions-to-electronic-documents>.

tecnología para verificar si hubo un previo uso de un algoritmo en una codificación es algo factible, hoy por hoy, con resultados tan confiables, como que ya una corte judicial en China la empleó para determinar si había ocurrido un plagio para un asunto en Derechos de Autor²¹⁹.

A partir de esa mejora en las consultas, naturalmente mejorará la evaluación de los demás criterios para el patentamiento de *software* y con ello, esta crítica se hará menos fuerte. No hay que perder de vista que cada tecnología disruptiva trae consigo esta recurrente protesta, y que igualmente se calla a medida que mejoran las librerías especializadas y se gana experiencia a partir de los casos judiciales.

Con todo, consideramos inapropiadas las nociones detrás de los requisitos de novedad, nivel inventivo y aplicación industrial²²⁰ para esta creación informática, tal como se comprenden hoy para las patentes, mucho más si es que seguirán concediéndose patentes de *software* a

²¹⁹ (i) (Hangzhou Huatai Yimei Culture Media Co., Ltd. v. Shenzhen Daotong Technology Development Co., Ltd. , 2018): Accesible en: <http://blog.dennemeyer.com/chinese-court-is-first-to-accept-blockchain-as-means-of-evidence>. (ii) Imaginando la ejecución de contratos y resolución de conflictos mediante el uso de *blockchain*: (Guzmán Caballero, 2018), accesible en: <https://www.ambitojuridico.com/noticias/tecnologia/tic/blockchain-el-cambio-de-los-paradigmas-juridicos>.

²²⁰ (i) Al respecto, nos alineamos más con: (Samuelson, Davis, Mitchell D., & Reichman, 1994), y no con: (Swinson, 1991). (ii) En una posición crítica al primer artículo: (Leith, 2007) Págs. 161 – 169. (iii) Sobre una descripción del por qué se considera que las patentes de *software* no resuelven un problema técnico: (Salas Pasuy, 2016) Pág. 441. (iv) Llamando a una reflexión sobre lo inapropiado de los criterios de patentabilidad aplicados a las patentes de *software*: (Bessen, The patent troll crisis is really a software patent crisis, 2013) Pág. 5.

futuro²²¹, debido a la naturaleza cumulativa de la codificación²²², los retos que plantea la actual automatización de la codificación y la inteligencia artificial.

Estas realidades, la primera muy presente desde que se empezaron a codificar los primeros *software*, y la segunda y tercera producto del vertiginoso adelanto de la industria, simplemente desdibujan la primicia y el salto cualitativo exigidos por la novedad y el nivel inventivo²²³, respectivamente, y de contera impiden describir de forma anticipada el uso previsible asumido por la aplicación industrial.

Como mucho de los algoritmos en un *software* viene dictado por los contornos del lenguaje de programación y búsqueda de eficiencia, el criterio de las patentes según el cual el invento o proceso debe tener un nivel inventivo parece ser un *standard* utópico para la tecnología bajo análisis, si se hace un estricto descarte de equivalentes conocidos o intercambio de

²²¹ Tal parece que las patentes de *software* vienen en retroceso, luego del caso (Alice Corp. v. CLS Bank International, 2014): (i) (Gugliuzza & Lemley, 2018) Pág. 782: “*The Federal Circuit has issued 33 precedential opinions on patentable subject matter since the Supreme Court decided Alice in June 2014. As Figure 4 shows, 26 of those 33 decisions (78,8%) found the patent invalid, while seven (21.2%) found the patent valid*”. En el mismo sentido: (ii) (Samuelson, Software patents are falling down, 2015) Pág. 27, (iii) (Salas Pasuy, 2016) Pág. 458. (iv) En contraste, la tendencia actual en la Superintendencia de Industria y Comercio indica una mejor y mayor propensión a conceder patentes en las áreas de electricidad, electrónica y telecomunicaciones así: Mientras en el período 2006 – 2009 se negaban el 93% de las solicitudes de patentes en estas áreas, en el período 2015 – 2016 se concedieron el 53% de las solicitudes, lo cual equivale a 146 patentes concedidas (Fuente: División de Nuevas Creaciones, SIC).

²²² (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2346, (ii) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 41.

²²³ (Gugliuzza & Lemley, 2018) Págs. 773 y 774: “*The Court in Alice explained that this second step of the analysis is ‘a search for an ‘inventive concept’-i.e., an element or combination of elements that is ‘sufficient to ensure that the patent in practice amounts to significantly more than a patent upon the [ineligible concept] itself.’ Applying that test to the facts of the case, the Court first determined that using intermediaries to mitigate risk was an abstract idea, in part because, like risk hedging in Bilski, ‘intermediated settlement is a fundamental economic practice long prevalent in our system of commerce’. On the second step, the Court concluded that merely performing that abstract idea on a general-purpose computer, as the patent claimed, did not represent the ‘inventive concept’ required for patent eligibility*”.

soluciones análogas²²⁴, así haya sido un criterio de patentabilidad superado por las actuales patentes de *software*.

Debido a lo anterior es que se han caído los títulos sobre las patentes de *software* cuestionadas²²⁵ y con ello, se evidencia que no es rigurosa la aplicación de los criterios de patentabilidad para el *software*²²⁶ o, visto de otra forma, la protección del *software* merece unos criterios de patentabilidad adaptados a la funcionalidad de los algoritmos expresados en el código fuente, pues es preferible que se abra el debate legislativo a una alternativa de protección para el *software*, que adaptar el Sistema de Patentes a este.

Por eso, postulamos en los Capítulos IV y V de este trabajo un sistema de protección *sui generis*, en el que se arranque por admitir una protección para el *software* separada del Sistema de Patentes, si bien se inspira en él, para de esta forma evitar la discusión de si es o no “materia patentable”; la novedad se califica según la explotación anterior de los algoritmos en Colombia²²⁷, el nivel inventivo pueda encontrarse aún en presencia de una yuxtaposición de líneas de códigos conocidas y la aplicación industrial se da por sentada.

²²⁴ (i) (Guía para examen de solicitudes de patente y modelo de utilidad, 2012) Pág. 100, (ii) (Merges, Menell, & Lemley, 2010) Pág. 327: “*Courts have long recognized that [c]arried to an extreme, the doctrine of equivalents could undermine the entire patent system*”.

²²⁵ (i) (Córdoba Marentes, 2006) Pág. 133 – 134, (ii) Sobre el decaimiento de la patente “click” otorgada a *Amazon*: (Uribe Arbeláez, 2005) Pág. 199, (iii) Un llamado por facilitar el decaimiento de las patentes de *software* se registra en: (Bessen, The patent troll crisis is really a software patent crisis, 2013) Pág. 7, (iv) Ahora, por cuestionamientos sobre si el *software* cumple el presupuesto de ser “materia patentable”: (Gugliuzza & Lemley, 2018) Pág. 789: “*Information technology accounts for the vast majority of decisions on our dataset: 98 out of 104. As Table 1 illustrate, The Federal Circuit has found the patent to be valid in only 7 of those 98 decisions (7,1%)*”.

²²⁶ (Bessen, The patent troll crisis is really a software patent crisis, 2013) Págs. 4 – 5.

²²⁷ Al no discriminar, el artículo 16 de la Decisión 486 de la Comisión de la Comunidad Andina de Naciones, contempla un criterio de novedad según el estado de la técnica a nivel internacional.

2.2. Las patentes de *software* no generan beneficios transversales para la industria del *software*.

Para quienes nos familiarizamos con el *software* de sistema operativo o aplicativo en un ordenador o un teléfono celular, nos pueden hacer sensibilizar las pequeñas y medianas empresas a las que una grande empresa de *software* se niega a licenciar su código fuente para que las pequeñas creen contenidos, hagan comunicar un *hardware* periférico o adhieran un *software* sustituto y/o complementario²²⁸; pero por grave que suena y es lo anterior, tales situaciones ocupan la mayor atención de la literatura referenciada sin advertir la realidad empírica del aprovechamiento de las patentes de *software* en la industria manufacturera²²⁹.

En realidad, el sonado incentivo provocado por las patentes de *software* a las firmas de esta industria es cuestionable, si tenemos en cuenta dos hechos de difícil análisis: Las patentes de *software* se conceden en mayor proporción a las firmas fuera de la industria²³⁰, y el presupuesto de investigación, desarrollo e innovación (I+D+i) se reduce en las firmas de la industria del *software*, al obtener un título de patente²³¹.

²²⁸ (i) (Swinson, 1991) Pág. 170, (ii) (Aguirre Soriano, 2014) Pág. 89, (iii) (The software industry and developing countries, 2012) Pág. 99. A otros también les conmueve cuando enfrenta a grandes empresas: (iv) Entre empresas manufactureras y del sector de *software*: (Stern, 2018) Pág. 193, (v) Entre empresas del sector del *software*: ASIA IP. Vol. 4. November 2012. Pág. 15: "[Pamela Jones, editor of award-winning patent blog Groklaw] says the Apple v. Samsung is just the latest example of a patent system that continues to sabotage innovation. 'software patents are, in my view, destroying innovation' she says. Software is algorithms, and algorithms are mathematics; and under US law, math cannot be patented, Jones explains".

²²⁹ (i) (Michael & Schankerman, 2006) Pág. 377: "Firms in the software industry show a positive effect on software copyright protection and a negative effect on software patent production, whereas firms in the manufacturing industry show the opposite effect", (ii) Este fenómeno puede deberse, en buena medida, a los efectos de la estandarización concertada para interoperabilidad de aparatos, ya que esta exige la divulgación de patentes: (Contreras, 2016) Pág. 862 y 875.

²³⁰ (Bessen & Hunt, The software patent experiment, 2004) Págs. 7 – 8.

²³¹ (i) (Bessen & Hunt, The software patent experiment, 2004) Pág. 3, (ii) (Sakakibara & Branstetter, 2001-2002) Pág. 165.

De hecho, obtener una patente NO es una variable en consideración para la creación de *software* en las firmas de la industria²³², en cambio sí es considerado por las firmas que más obtienen patentes de *software*, es decir las que producen y comercializan otro tipo de productos²³³. En nuestro concepto, esta evidencia empírica traiciona la expectativa de beneficios generales por la concesión de las patentes de *software*.

Se supone que incentivar el progreso tecnológico exige conceder derechos de exclusividad a ciertos sujetos de derecho para que estos y otros tomen el riesgo de innovar en bienes económicamente “públicos”²³⁴; es decir, que la propiedad intelectual se justifica, políticamente, en el temor de ver disminuida la creatividad, si no se les premia a los innovadores con un derecho, para que recuperen sus costos directos, indirectos y de oportunidad²³⁵.

²³² (i) (Dukrok & Hwang, 2010) Pág. 377 y 382, (ii) (Swinson, 1991) Pág. 165, (iii) (López, 2009) Págs. 9 y 13, (iv) (Caracterización del sector de teleinformática, software y TI en Colombia. Año 2015. , 2016) Pág 224: “*De acuerdo con las respuestas, las empresas encuestadas muestran su familiaridad con los procesos de innovación y de renovación tecnológica; sería interesante profundizar sobre las circunstancias que ocasionan la baja propensión del empresario por obtener patentes sobre sus innovaciones, por cuanto estas son una salvaguarda del patrimonio de las empresas representado en su propiedad intelectual y un indicador importante sobre la dinámica de innovación del país*”, (v) De las 146 patentes concedidas en Colombia, en las áreas de electricidad, electrónica y telecomunicaciones, apenas una tercera parte (48) corresponde a firmas dedicadas a la creación de *software*, o directamente involucradas en el sector de telecomunicaciones (Fuente: División de Nuevas Creaciones, SIC).

²³³ (Mansfield, 1984-2002) Pág. 33.

²³⁴ (i) (Márquez Escobar, Introducción a la relación entre el derecho y la economía, 2004) Pág. 61: “*¿es eficiente producir de modo privado un bien público? La respuesta es no. No es eficiente puesto que se presenta el fenómeno del free rider. Un free rider es un agente oportunista que observa que otro agente adquiere el bien público y debido a que dicho bien es no excluyente y no rival, no asume ningún costo para adquirirlo. La solución a la ineficiencia que genera la externalidad positiva de los bienes públicos es determinada por un subsidio. Como el productor sabe que no va a ser demandado del mismo modo que un bien privado el bien público, la mejor forma de incentivar su producción es dándole al productor un subsidio que compense las pérdidas de ingresos debidas a la naturaleza del bien*”, (ii) (Arenas Correa, 2013) Pág. 197, (iii) (Conpes 3582, 2009): Págs. 9 – 10: “*La literatura económica reconoce tres fallas de mercado que pueden conducir a niveles de inversión en ACTI inferiores a lo óptimo: 1. El conocimiento como bien público. El hecho de que un empresario no se pueda apropiar de todos los resultados de la inversión en generación de conocimiento hace que no tenga suficientes incentivos para invertir. 2. La tecnología es claramente un bien afectado por problemas de información imperfecta – y de barreras a la entrada – dado que los costos e infraestructura necesarios no siempre están al alcance de todos. 3. La incertidumbre acerca de los costos y beneficios de la investigación y sus resultados. Esta incertidumbre reduce y en algunos casos elimina del todo la inversión en ACTI*”.

²³⁵ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2382, (ii) (Dam, 1995) Pág. 333, (iii) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1337 – 1338: “*The basic linkage between the scope of*

Pero si en el caso de las patentes de *software*, las firmas que son titulares entran en la zona de confort que aparece cobrar regalías por veinte años, procrastinando la investigación y desarrollo de nuevos productos²³⁶, y además resulta que las firmas que mejor aprovechan el sistema de patente son las que menos empleos generan en la industria del *software*²³⁷; entonces conceder patentes de *software* es un mal negocio para los Estados, porque además de no incentivar el progreso²³⁸, imponen un alto precio en el mercado relevante, sin un beneficio general apreciable²³⁹.

Más allá de generalizar en la mala idea de conceder veinte años de exclusiva a los titulares de las patentes, el argumento político en contra del amplio término de protección de las patentes para el caso del *software* debe fundarse en el desincentivo que producen a la innovación²⁴⁰, y por ello el poco beneficio social que generan.

intellectual property protection and the public welfare involves three steps. First, enhancing the scope of intellectual property protection (e.g., by increasing the term of legal protection or expanding the breadth of legal protection) increases the expected reward to the creator by enhancing the opportunity for monopolistic exploitation of any works created. Second, increased rewards encourage inventive activity. Moreover, the disclosure of new discoveries that is encouraged by protection further spurs inventive activity. Third, greater investment in inventive activity results in the discovery of more ideas and faster advancement of technology, thereby increasing the range of products and reducing the cost of products to society”.

²³⁶ (i) (Dukrok & Hwang, 2010) Pág. 382, (ii) (Dam, 1995) Pág. 369.

²³⁷ En proporción al capital invertido. (i) (Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia, 2014) Pág. 63, (ii) (Conpes 3834, 2015) Pág. 15 y 20, (iii) (Buitrago Restrepo & DuqueMárquez, 2013) Las economías relacionadas con los derechos de autor emplean a 1’160 personas y producen valor por USD\$11.000’, lo que equivale a un 3,3 % PIB Pags. 127 – 133.

²³⁸ O como se puso de relieve en la investigación hecha a Microsoft, la patente le sirve para aplazar el lanzamiento de adelantos en el mercado: (de la Cruz Camargo, 2008) Pág. 130.

²³⁹ (i) Probablemente con la excepción de los protocolos concertados gremialmente, para el desarrollo y explotación de la Internet: (Contreras, 2016) Pág. 882, (ii) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Págs. 46 – 47.

²⁴⁰ (Dam, 1995) Pág. 369.

2.3. El problema de las *patent trolls* es la interoperabilidad y los costos de reentrenamiento.

Aunque el Derecho de Autor recibe grandes críticas como modelo de protección para el *software*, muchos le abonan la excepción contemplada para la interoperabilidad²⁴¹, ausente en el Sistema de Patentes²⁴², seguramente porque el Derecho de Autor ha debido lidiar con los problemas aparejados al desarrollo de la industria y en cambio, como el Sistema de Patentes le queda muy grande al *software*, la detección de los defectos es complicada y demorada²⁴³.

Y es que, como en toda industria, las compañías establecidas tendrán un fuerte incentivo en procurar el patentamiento de *software* a fin de controlar la nueva entrada de oferentes al mercado, si estos emplean un algoritmo innovador o uno conocido empleado para un uso innovador, a fin de recuperar sus costos en la cortísima ventana que tienen; de esta forma, asegurarán la renta derivada de contratar con firmas proveedoras de contenido o generadoras de redes²⁴⁴, dejando apartadas de cualquier contratación a firmas sucesoras (“*follow-on*”) y competidoras (“*me-too*”)²⁴⁵.

²⁴¹ Empleamos el término “interoperabilidad” como evolución de “compatibilidad”, pues al inicio de la industria lo importante era que el *hardware* periférico pudiera controlarse por un sistema operativo, y que las piezas de un computador fueran reemplazables con productos de diferentes firmas productoras. Hoy, la preocupación no solo se concentra en el *hardware* periférico o interno, sino también en el *driver* que los controla y las plataformas sobre las cuales se integran. (i) (Gómez Segade, 2001) Págs. 885 – 886, (ii) (Dam, 1995) Págs. 342 y 344, (iii) Resaltando que de no existir esta excepción, el Derecho de Autor podría estar consagrando derechos de patentes *de facto*: (Samuelson & Scotchmer, The law and economics of reverse engineering, 2002) Págs. 1610 – 1611.

²⁴² (i) (Uribe Arbeláez, 2005) Pág. 206, (ii) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 6 y 18, (iii) Reseñando una derrotada propuesta de Directiva en ese sentido: (Graef, 2014) Págs. 15 - 16.

²⁴³ (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 7.

²⁴⁴ (i) (Calle D'Alleman, 2012) Pág. 193, (ii) (Dam, 1995) Pág. 337, (iii) (Stern, 2018) Pág. 210: “*Even when an anointed technology is superior, its market value (i.e., ability to command payment) is always increased above its pre-standardization value because standardization excludes any competitive alternatives. The surplus is an artifact of the standardization process*”.

²⁴⁵ (Charfoos, 2002) Pág. 19.

Por esto, acumular las patentes de *software* a fin de controlar el acceso de firmas sucesoras (“*follow-on*”) o competidoras (“*me too*”) es la motivación de las empresas establecidas en la industria cuando se convierten, de *facto* o *jure et de jure*, en el *standard* seguido por los ofertantes del mercado relevante, debido a que el tiempo transcurrido ha familiarizado los usuarios con la *interfaz* o funcionalidades del *software*²⁴⁶. Esto, como toda barrera de entrada, es misión de las autoridades de competencia supervisar²⁴⁷, si llega a suceder que retrasa la aparición de mayor oferta de productos²⁴⁸ o se convierte en un abuso en la explotación de los derechos concedidos por la propiedad industrial²⁴⁹.

Nada debe impedir el ascenso de una empresa exitosa hasta lograr posición dominante²⁵⁰ en un mercado relevante, pues su eficiencia no debe ser sancionada en nombre del interés público; pero lo que sí puede regularse es el acceso a la plataforma que alcanza el nivel de importancia social y económica, que caracteriza a las empresas con posición dominante, a fin de

²⁴⁶ (Dam, 1995) Págs. 349 – 351.

²⁴⁷ (Contreras J. L., 2013) Pág. 58.

²⁴⁸ Señalando que uno de los objetivos del Derecho de la Competencia es promover la innovación y el progreso tecnológico: (Jiménez Valderrama, 2019) Pág. 12.

²⁴⁹ (i) Artículo 40 ADPIC, (ii) (Swinson, 1991) Pág. 171, (iii) (Burtis & Kobayashi, 2001) Pág. 152 - 153, (iv) (Samuelson & Scotchmer, The law and economics of reverse engineering, 2002) Pág. 1661, (v) (de la Cruz Camargo, 2008) Pág. 124. En este artículo se analiza la investigación hecha a Microsoft por la Comisión Europea de la Competencia, por abuso de posición dominante, al limitar el desarrollo técnico en perjuicio de los consumidores y ventas atadas. (vi) También: (Zuleta Londoño, 2005) Págs. 541 y 545, (vii) Reseñando varios casos en el sector del *software* supervisados por la Comisión Europea: (Graef, 2014) Págs. 9 - 11 (viii) Describiendo el abuso de los derechos de patentes y las ventas atadas que caracterizan el modelo de negocios de la Firma Qualcomm, en el mercado de *chips* para teléfonos celulares: (Stern, 2018) Págs. 135 – 142, (ix) (Contreras J. L., 2013) Pág. 50: “*Aware of the possibility of patent hold-up and stacking, many SDOs have promulgated internal policies designed to mitigate these risks. Perhaps the most prevalent of these is a requirement that SDO participants’ license their patents to all potential vendors of technologies implementing those standards on terms that are ‘fair’, ‘reasonable’, and ‘non-discriminatory’ (FRAND)*”.

²⁵⁰ (i) (Klein, 1993) Pág. 74: “*The idea is based upon using the economic models of perfect competition and monopoly as the two extremes or end points on a continuum that defines a firm’s pricing power. Perfectly competitive firms have no pricing power (they are price-takers) while monopolist have maximum pricing power (they are price-setters). However, this framework fails to recognize that firms may have negatively sloped demands and individual pricing power for reasons that have nothing to do with monopoly*”, (ii) Con unos cargos similares, Apple y la FTC han acusado a Qualcomm: (Stern, 2018) Pág. 144.

asegurar interoperabilidad entre *hardware* y *software*²⁵¹, lo mismo que promover cierto grado de competencia - Y con esta más innovación²⁵² - .

Este es el enfoque que consideramos acertado para lidiar con la externalidad de redes, en los pocos casos en los cuales alcanza una firma en la industria del *software* tener posición dominante en el mercado²⁵³, porque además de ser una externalidad consabida por el derecho de competencia, se debe en mucho al excesivo término de protección concedido para el *software*; luego, no estamos a favor de una normativa especial de regulación económica para el caso del *software*, pues se corre el riesgo de hacer incoherente a ambos cuerpos normativos (el de la propiedad intelectual y el de la competencia)²⁵⁴ y además, es posible que quede superada prontamente por una realidad cambiante²⁵⁵.

²⁵¹ (i) (Charfoos, 2002) Pág. 18, (ii) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 22, (iii) (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1625. (iv) Una aproximación a partir de la regulación para integraciones y el abuso de posición dominante en: (Graef, 2014) Pág. 11.

²⁵² (i) Anotando que el argumento pro-competencia que se lanza para fundamentar excepciones de interoperabilidad, pasa por alto que basta la observación de las firmas sucesoras (“*follow-on*”), respecto del desempeño del producto innovador, para que estas se encuentren beneficiadas socialmente: (Dam, 1995) Pág. 359, (ii) En tono escéptico hacia la comprensión de los procesos de innovación, y dejando una estela a favor de mercados concentrados: (Burtis & Kobayashi, 2001) Pág. 158: “*Allowing antitrust scrutiny of the patent holder's statutory right to exclude would increase applications of antitrust law to potentially pro-competitive conduct. Because economists and courts do not fully understand the innovation process, they are unlikely to be able reliably to differentiate between pro-competitive and anti-competitive effects of conduct*”. En sentido contrario, pero debido al enfoque de la estandarización: (Contreras J. L., 2013) Pág. 47: “*According to the U.S. Department of Justice and Federal Trade Commission, ‘Standards can make products less costly for firms to produce and more valuable to consumers. They can increase innovation, efficiency, and consumer choice; foster public health and safety; and serve as a ‘fundamental building block for international trade’*”.

²⁵³ (Dam, 1995) Pág. 350.

²⁵⁴ (i) En este sentido: (Dam, 1995) Pág. 359, (ii) Haciendo un recuento histórico de las vicisitudes que la Corte Suprema de los Estados Unidos de América ha debido sortear, para conservar coherente y en sus respectivos contornos diferentes materias normativas, cuando se cruzan con derechos de propiedad intelectual (Entre ellos las normas sobre prácticas restrictivas de la competencia): (Duffy & Hynes, 2016) Págs. 22 – 27.

²⁵⁵ Señalando que las autoridades de competencia podrían incluso respaldar mañana, lo que hoy consideramos externalidades negativas: (Maggiolino, 2011) Pág. 85.

**Capítulo III. Feudalismo informático: Esto pasa cuando se juntan el *software* y los
secretos empresariales**

Si el titular de un *software*, ingenuamente, considera que puede conservar secreto su código fuente²⁵⁶, por ejemplo porque sobredimensiona los contratos de confidencialidad que suscribe con los colaboradores del desarrollo²⁵⁷ y/o el comercializador del *software*²⁵⁸, entonces confiará sus derechos a las normas sobre represión de la competencia desleal²⁵⁹, específicamente en Colombia en el artículo 16 de la Ley 256 de 1996, que sanciona el acto de violación de secretos²⁶⁰.

El problema con este proceder radica en la naturaleza misma de la Protección a Secretos Empresariales²⁶¹, pues esta reglamentación, como en el caso del Derecho de Autor, estimula la investigación independiente²⁶² y en consecuencia, no concede protección al secreto que es

²⁵⁶ (i) Al respecto de la noción de “código fuente”: Circular externa No. 5, Dirección Nacional de Derecho de Autor. 9 de octubre de 2001, (ii) Dejando claro qué es lo oculto en el *software*: (Rengifo García, 2003) Págs. 215 - 216: “*La estructura externa es la parte del programa que puede verse en la pantalla, los elementos ostensibles que se expresan por pantalla o mediante la presentación gráfica o textual como las instrucciones o el manual y la estructura interna está compuesta por los elementos ocultos de la obra, es decir, aquellos que no pueden ser perceptibles por un usuario normal*”.

²⁵⁷ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2333: “*Computer programs are an unusual kind of industrial product because the bulk of the know-how required to create them is accessible on or near the face of the product distributed in the marketplace. There are several reasons why this is so: (1) Software products need little specialized mass-production know-how; (2) they are particularly rich in design know-how; and (3) they are more susceptible in some respects to reverse engineering than traditional industrial products*”.

²⁵⁸ (As Colombia Ltda. v. Informática y Gestión S.A., 2006) Pág. 38 – 40.

²⁵⁹ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2389.

²⁶⁰ El caso (Computer Associates International Inc. v. Altai Inc., 1992) abordó también el concepto de “*misappropriation*”, equivalente al de “violación de secretos”.

²⁶¹ (i) Artículo 260, Decisión 486 de la Comisión de la Comunidad Andina de Naciones: “*Se considerará como secreto empresarial cualquier información no divulgada que una persona natural o jurídica legítimamente posea, que pueda usarse en alguna actividad productiva, industrial o comercial, y que sea susceptible de transmitirse a un tercero, en la medida que dicha información sea: a) secreta, en el sentido que como conjunto o en la configuración y reunión precisa de sus componentes, no sea generalmente conocida ni fácilmente accesible por quienes se encuentran en los círculos que normalmente manejan la información respectiva; b) tenga un valor comercial por ser secreta; y c) haya sido objeto de medidas razonables tomadas por su legítimo poseedor para mantenerla secreta. La información de un secreto empresarial podrá estar referida a la naturaleza, características o finalidades de los productos; a los métodos o procesos de producción; o, a los medios o formas de distribución o comercialización de productos o prestación de servicios*”, (ii) (Galán Corona, 2011) Pág. 356, (iii) (Drexl, 2011) Págs 182 – 183, (iv) (Carranza Torres, 2008) Pág. 223.

²⁶² (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2380.

revelado sin apelar a la infracción de deberes de confidencialidad, o que se descubre sin la ejecución de actividades de espionaje industrial²⁶³.

En consecuencia, el innovador es conducido a creer que la protección del *software* a través de las normas que protegen el secreto empresarial le prohíbe a todos el uso de la información²⁶⁴, y le permitirá contar con suficiente tiempo para recuperar sus costos en (I+D+i) del *software* antes de que un competidor ofrezca en el mercado un *software* sustituto; pero, en realidad, la protección del *software* a través de las normas sobre secreto empresarial no resguardan al innovador de la descompilación²⁶⁵ del código fuente, por ser esta, en términos generales, una forma de ingeniería reversa permitida dentro del marco de la investigación independiente²⁶⁶.

²⁶³ (i) (Galán Corona, 2011) Pág. 368, (ii) (Jiménez Valderrama, 2019) Págs. 98 – 99.

²⁶⁴ (i) (Gómez Segade, 2001) Pág. 906, (ii) (Drexl, 2011) Pág. 180, (iii) (Ascarelli, 1970) Pág. 260.

²⁶⁵ La “descompilación” consiste en traducir el código objeto, hasta llegar lo más cerca posible al código fuente del *software*. La “compilación”, por el contrario, es el procedimiento *standard*, dentro de los pasos que caracterizan la elaboración de un *software*, por medio del cual se lleva el código fuente a un código objeto legible únicamente por la CPU de un computador.

²⁶⁶ (i) (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Págs. 1583 y 1660. (ii) Advirtiendo la legalidad de la ingeniería reversa, pero anotando la validez de cláusulas contractuales que la prohíben en Estados Unidos de América, y la invalidez de esas cláusulas en Europa, si es para interoperabilidad: (Nguyen , Gomulkiewicz , & Conway , 2012) Pág. 545. Con todo, el alcance de las cláusulas que prohíben la ingeniería reversa claramente es limitada, teniendo en cuenta que se dan entre quienes han celebrado el contrato respectivo, sin efectos frente a los terceros que pudieran aprovecharse ilegítimamente del código fuente. (iii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2342 – 2343 y 2387, (iv) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 17.

Como resultado, la firma sucesora (“*follow-on*”) o competidora (“*me too*”) está en capacidad de clonar el *software* innovador a un mucho menor costo²⁶⁷ que lo invertido por el innovador²⁶⁸, quien percatado de esto disminuirá o acabará su presupuesto para (I+D+i)²⁶⁹.

Ahora, una firma que eficientemente pueda controlar la divulgación de su código fuente, o parte del mismo, a través de un sistema de contratos ejecutables, será capaz de controlar grandemente las condiciones del mercado relevante²⁷⁰ en cuanto lanzamiento de productos innovadores y complementarios se refiere, pues hará una integración económica, sino jurídica, con sus proveedores de partes y contenidos²⁷¹.

Lo anterior, pues teniendo claro que la eficiencia dicta la programación de cualquier *software*, conservar eficientemente en secreto el algoritmo disuadirá potenciales competidores de lanzar productos aislados y extraños, por el costo asociado a la introducción de nuevos productos en el mercado y la barrera del conocimiento adquirido por los usuarios anteriores que debería flanquear, creando así un feudo informático para sí²⁷².

²⁶⁷ Es bueno aclarar que cuando ciertos autores mencionan que el costo marginal de hacer una copia es \$0, no se están refiriendo al costo de clonar: (i) (Posner, *Economic analysis of Law*, 2011) Pág. 412, (ii) (Posner, *Antitrust law*, 2001) Pág. 246, (iii) Sobre lo que implica clonar: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2336.

²⁶⁸ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2318: “*A skilled programmer can, in other words, copy the behavior of a program exactly, without appropriating any of its text*”.

²⁶⁹ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2333 y 2388, (ii) (Samuelson & Scotchmer, *The law an economics of reverse engineering*, 2002) Pág. 1582.

²⁷⁰ (i) (Carranza Torres, 2008) Pág. 247, (ii) (Blind Knut & Firedewald, 2005) Pág. 13, (iii) (Monroy Rodríguez, 2012) Pág. 111, (iv) (Aguirre Soriano, 2014) Pág. 110, (v) (Samuelson & Scotchmer, *The law an economics of reverse engineering*, 2002) Pág. 1624: “*‘Tipping’ means that a single interface succeeds in becoming the standard in the market, creating a monopoly*”.

²⁷¹ Narrando la ingeniosa complementación de derechos de patentes y estipulaciones contractuales concadenadas que practica Qualcomm Inc.: (Duffy & Hynes, 2016), Pág. 59.

²⁷² (i) (Drexler, 2011) Pág. 183: “*It has to be noted that trade secrecy protection involves a high risk of monopolization of markets. Wherever the law protects against access to information, there is a risk that markets are automatically monopolized if competitors depend on access to that information for entering the relevant market*”, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2376.

Apoyado particularmente en el contexto económico de la industria y de algunos casos judiciales, el presente capítulo hace un balance entre ventajas (1) y desventajas (2) de la protección del *software* mediante secretos empresariales.

1. Ventajas de la protección del *software* mediante secretos empresariales

Para una firma emprendedora (“*start-up*”) o no-estratégica (“*spin-off*”), con bajo presupuesto para invertir en la protección jurídica de su propiedad intelectual y sin saber realmente la respuesta del mercado que tendrá el producto que forja, el código del silencio entre sus partes interesadas (“*stakeholders*”) es su máxima moral, mientras definen qué hacer con el producto creado. Las firmas bien establecidas en la industria, en cambio, usualmente definen una unidad de negocios para (I+D+i)²⁷³, con gerentes, objetivos y cronogramas determinados, cubiertos bajo estrictos contratos de confidencialidad.

Unas y otras firmas aprovechan de esta manera la versatilidad de la Protección a Secretos Empresariales, que también es un modelo de protección internacional²⁷⁴, por lo cual su asunción en la industria ha sido espontánea²⁷⁵ y generalizada²⁷⁶, potencializado por la facilidad de entregar al usuario final únicamente el código objeto²⁷⁷.

²⁷³ (Lema, Quadros, & Schmitz, 2012) Pág. 38.

²⁷⁴ (i) Artículo 39 ADPIC, (ii) Artículo 10 *bis* Convenio de París.

²⁷⁵ El aprovechamiento de las normas sobre Protección a Secretos Empresariales parece haber sido estado del arte en una época incipiente de la industria, en la cual todavía *hardware* y *software* permanecían casados sacramentalmente. (Smith & Mann, 2004) Pag. 243.

²⁷⁶ (i) (López, 2009) Pág. 14, (ii) (Nguyen , Gomulkiewicz , & Conway , 2012) Pág. 515.

²⁷⁷ (Nguyen , Gomulkiewicz , & Conway , 2012) Pág. 509.

Esto ha hecho que los secretos empresariales se acomoden muy bien a los intereses de los involucrados en la industria, pues resultan con un mayor alcance que el Derecho de Autor y reducen la litigiosidad, como a continuación se explica:

1.1. Modelo de protección automático y ampliado.

Tal como en el Derecho de Autor, la protección del *software* vía secretos empresariales es un modelo de protección “a la mano” para muchas firmas dedicadas a la creación de *software*, básicamente porque les permite recibir una protección inmediata, desde la muy temprana etapa de mera ideación²⁷⁸; es decir, los poseedores de secretos empresariales no deben acudir a registros oficiales para recibir tutela jurídica a sus avances, ni están restringidos por supuestos de expresión, originalidad, “materia patentable”, o requisitos de novedad y nivel inventivo.

Además, a diferencia del Derecho de Autor y del Sistema de Patentes, la Protección a Secretos Empresariales sirve para resguardar la dialéctica del *software*, que es la combinación de funcionalidad y experiencia del usuario²⁷⁹, en ampliada defensa de la propiedad intelectual respecto del *software*, más allá de lo que abarcaría el Derecho de Autor²⁸⁰ y las reivindicaciones de una patente, en particular prohibiendo el *cloning*²⁸¹.

²⁷⁸ (Calle D'Alleman, 2012) Pág. 177.

²⁷⁹ Incluido el “*look-n-feel*”: (Bouza López, 1997) Pág. 344.

²⁸⁰ (i) (Buitrago, 2007) Pág. 159: “*Algunos países reconocen abiertamente la posibilidad de una protección como secreto comercial, tal es el caso de los EE.U. La opción por este modo de protección tiene la ventaja de que el ámbito de protección es mayor que el del copyright, ya que puede comprender las ideas y no solo la expresión de las mismas; la desventaja es que dicha protección desaparece cuando el secreto se hace público*”. (ii) Reseñando el caso (Lasercomb America Inc. v. Job Reynolds, 1990), donde el creador de un *software* demandó protección a su Derecho de Autor, perdiendo la demanda contra el licenciatarario que había creado un *software* sustituto: (Rodríguez Yong, 2012) Pág. 335.

²⁸¹ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2385, 2392 y 2396, (ii) (Nguyen , Gomulkiewicz , & Conway , 2012) Págs. 515 – 516, (iii) (Hong & Cheng-ssu, 1999) Pág. 374: “*In China, trade secret protection is*

Si a ello se suma que la violación a secretos empresariales se sanciona en los Tratados internacionales más importantes sobre propiedad industrial del planeta, sin contemplar derechos morales de autor, ni eventos de licencias obligatorias, es virtualmente perpetuo, mientras no sea de fácil acceso²⁸², y hasta algunas normas laborales sancionan su violación²⁸³, es fácil entender por qué la industria del *software* acoge extendidamente a los secretos empresariales como la forma obvia de protección para los códigos fuente que crea.

1.2. Menor propensión hacia el litigio.

Teniendo claro que la eficiencia dicta la programación de cualquier *software*, que son muy rápidos los cambios en las técnicas de programación y cortos los períodos de tiempo para los productos innovadores antes de que se enfrenten a productos sustitutos y/o complementarios; contratar el licenciamiento de porciones del código fuente arbitra entre las firmas involucradas los costos de producción entre ellas²⁸⁴, a fin de introducir a tiempo sus productos en el mercado, montados sobre una plataforma habitual para los usuarios finales.

very important for computer software, because software may not be granted patent protection without being combined with hardware, and copyright law can only protect the expression of software, rather than the idea”.

²⁸² (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2386, (ii) Vale la pena resaltar el artículo 18.78 del Tratado de Cooperación Trans-Pacífico (“TPP” por sus siglas en inglés), pues contempla la obligación para las partes signatarias de contemplar sanciones penales a quienes violen un secreto empresarial contenido en un sistema informático. Sabemos que el Tratado no ha entrado vigor, pero es relevante rescatar la negociación multilateral de una obligación de esta naturaleza, pues podría servir de base para futuros acuerdos multilaterales, como por ejemplo una actualización de ADPIC.

²⁸³ En Colombia se permite la terminación con justa causa de la relación laboral con el trabajador, entre otras sanciones, en virtud del artículo 62-a-8 del Código Sustantivo del Trabajo.

²⁸⁴ (i) (Samuelson & Scotchmer, *The law and economics of reverse engineering*, 2002) Pág. 1650, (ii) (Dam, 1995) Pág. 365.

Más aún, la contratación extendida de porciones de código fuente de *software* no solo genera la red con proveedores de contenido a la que nos hemos referido, sino que crea condiciones para mercados secundarios²⁸⁵ para firmas dedicadas al mantenimiento, soporte y capacitación de técnicos y usuarios finales²⁸⁶, bajo el abrigo e inversión en publicidad de la firma titular del secreto empresarial.

Todo esto se produce gracias a la *estandarización* de contratos entre el titular del secreto empresarial y la red de aliados que construya²⁸⁷, lo cual disminuye ostensiblemente la propensión hacia el litigio; porque la predisposición de los contratos alinea a los vinculados con unos parámetros y valores determinados por el titular del secreto, lo cual disuade jurídica y moralmente la revelación del secreto²⁸⁸ y además, las interacciones entre el titular del derecho y sus vinculados hacen evolucionar tales valores y parámetros, de forma más o menos armónica, a través de las concesiones recíprocas que de tanto en tanto hacen²⁸⁹.

²⁸⁵ (i) (Dam, 1995) Pág. 361, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2385.

²⁸⁶ (i) (Menell, Tailoring legal protection for computer software, 1987) Págs. 1367 – 1368, (ii) (Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia, 2014) Pág. 54: “Se solicitó a las empresas que asignaron el porcentaje de su facturación, en el 2012, por línea de negocios; dando como resultado que el 20% de las empresas encuestadas considera el “Licenciamiento de Software Nacional-Producto Propio” como su actividad principal, mientras que el 18% se dedica al “Desarrollo a la Medida” y el 17% a Servicios de Software, Soporte y Mantenimiento”.

²⁸⁷ (i) (Dam, 1995) Pág. 353, (ii) (Drexler, 2011) Pág. 180.

²⁸⁸ (Nguyen, Gomulkiewicz, & Conway, 2012) Pág. 519.

²⁸⁹ Nos referimos a la posibilidad de negociar licencias exclusivas a ciertos proveedores de contenido, o hacer crecer mancomunadamente la plataforma: (Samuelson & Scotchmer, The law and economics of reverse engineering, 2002) Págs. 1617 – 1618, y 1623.

2. Desventajas de la protección del *software* mediante la Protección a Secretos

Empresariales

Así como es una ventaja recibir una protección inmediata por las normas de competencia desleal, los secretos empresariales imponen un sobre costo para todas las firmas dedicadas a la creación de *software*, pues si desean lanzar productos sustitutos a previos productos innovadores en el mercado, deberán internarse en la ingeniería reversa²⁹⁰ para estudiar y “clonarlo”; o si han dedicado tiempo y recursos al desarrollo y expresión de una idea en forma de *software*, sin saberlo podrían estar atravesando el mismo camino de “ensayo y error” transitado previamente por otros²⁹¹.

El resultado para la sociedad no es positivo en ningún caso, pues la duplicación en investigaciones o desarrollos pares resta tiempo y recursos en el avance cualitativo de la tecnología, lo cual retrasa la introducción de mejores productos informáticos en el mercado²⁹². Lo anterior es igualmente perjudicial en el evento en que se permita la ingeniería reversa para la interoperabilidad de productos informáticos, pues ese procedimiento consume tiempo y dinero²⁹³, y aunque ningún estudio ha confirmado cuán caro ha sido, podría evitarse a través de la consulta de un repositorio de algoritmos previamente empleados para *software*, en el marco de un derecho de remuneración.

²⁹⁰ Bien sea a través del proceso de descompilación o de “*Black Box*”, que consiste en insertar información al *software* para estudiar los resultados que arroja el procesamiento de datos, y de esta forma concluir cuál habría sido el algoritmo explotado. (i) (Lemley, Menell, Merges, Samuelson, & Carver, 2011). Pág. 26, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2393, (iii) (Gómez Segade, 2001) Pág. 883, (iv) (Graef, 2014) Pág. 14.

²⁹¹ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2387.

²⁹² (i) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1343 – 1344, (ii) (Dam, 1995) Pág. 359.

²⁹³ Intrigada por la razón que justifica la ingeniería reversa como *standard* de la industria para todo proceso, siendo un trabajo tan arduo: (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1614.

Sin duda los titulares de secretos empresariales, que exitosamente hubieren conformado una red de proveedores de contenido, soporte y capacitación, no comulgarán con un cambio en la legislación que les rete el orden en sus feudos informáticos²⁹⁴, y por supuesto que no estarán incentivadas a registrar sus algoritmos, pero aquí mencionaremos por qué es importante desaprobar esta vía de protección para el *software*.

2.1. Ingeniería reversa, ¿para replicar productos?...

El seductor argumento político por el cual se permite la descompilación y/o la ingeniería reversa con el cándido propósito de interoperabilidad, no compensa el desperdicio de tiempo y dinero que supone el proceso de descompilación – compilación – reescritura y de contera, crea el riesgo moral para el que descompila o hace ingeniería reversa de crear un producto sustituto y/o complementario, o vilmente una copia pirata, en vez de uno compatible con el *software* innovador²⁹⁵; luego, permitir la ingeniería reversa es estimular firmas que no procuran innovar sino repetir o aprovecharse (“*free ride*”) de la infraestructura y *marketing* sin haber aportado para

²⁹⁴ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2421, (ii) Describiendo todos los flancos que resguarda el innovador, para aprovechar al máximo la explotación de su secreto empresarial, desde tempranas etapas de formación: (Menell, Tailoring legal protection for computer software, 1987) Pág. 1339.

²⁹⁵ Por esto, no compartimos la enumeración de excepciones que permitan la ingeniería reversa, según el propósito del beneficiado: (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1656.

su creación y sostenimiento²⁹⁶, en detrimento del espacio de tiempo que requieren las firmas innovadoras para recuperar sus costos de innovación²⁹⁷.

Lo mismo ocurre con el que desea lanzar un nuevo *hardware* al mercado, o un sistema operativo, pues el “urbanismo” (Application Programming Interfaces “API’s”) se licencia entre quienes hacen aplicaciones a fin de difuminar los costos asociados al *hardware* en el precio cobrado por las licencias del *driver*²⁹⁸; si la arquitectura del *driver* es de código abierto, entonces no se puede hacer esta jugada, a menos que se desconozcan términos de la licencia de código abierto. El propietario del *hardware* puede hacer directamente el *software* aplicativo, pero en este caso y el anterior, los innovadores del *hardware* o de la plataforma de *software* corren el riesgo del éxito o fracaso de estos, si el *software* aplicativo no es favorecido por el mercado²⁹⁹.

Puede creerse que la (I+D+i) incluye algo de descompilación y/o ingeniería reversa, y sin duda en la actual protección del *software* esto es lo que ocurre³⁰⁰, pero otro sería el resultado si pensáramos en un sistema que permitiera el acceso a las funcionalidades del *software*, con el

²⁹⁶ (i) (Samuelson & Scotchmer, *The law an economics of reverse engineering*, 2002) Pág. 1623: “*The platform developer's ability to attract developers to develop applications for the platform and to recoup subsidies incurred to attract application developers may be negatively affected to some degree by a rule favoring reverse engineering. If reverse engineering is lawful, licensed developers may worry about their recoupment of R&D expenses if unlicensed entrants can now offer competing applications for the platform-and can do so without paying royalties to the platform developer for the right to make applications for the platform*”, (ii) Comentando la sentencia dictada en el asunto (Sega Enterprises Ltd. v. Accolade, 1992), en el que salió favorecida la parte demandada no obstante la descompilación del código fuente de la demandante: (Carranza Torres, 2008) Págs. 253 – 255.

²⁹⁷ (i) (Dam, 1995) Pág. 358 – 359, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2341.

²⁹⁸ (i) (Menell, *Tailoring legal protection for computer software*, 1987) Pág. 1362, (ii) Sobre los riesgos que corren las API, protegidas por secretos empresariales: (Samuelson & Scotchmer, *The law an economics of reverse engineering*, 2002) Pág. 1620.

²⁹⁹ (Samuelson & Scotchmer, *The law an economics of reverse engineering*, 2002) Pág. 1616.

³⁰⁰ Mencionando la segunda mirada que hacen las firmas sucesoras (“*follow-on*”) y competidoras (“*me too*”) al producto innovador, lo cual debería permitirles competir mediante la creación de productos innovadores: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2367.

explícito propósito de crear productos innovadores³⁰¹, en vez de réplicas de *software* previamente exitosas³⁰².

Emular lo que en el mercado ocurre es una apropiada justificación política, económica y jurídica para actualizar los regímenes, pero es la política pública la que soluciona atrasos debidos a movimientos en la industria que no pueden ser resueltos mediante negociación privada³⁰³. Por esto, la motivación política actual en torno a la economía naranja³⁰⁴, debe aprovecharse para

³⁰¹ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2379: “[T]he smaller the taking, the less dependent the creation, the less similar the results, and the less proximate the markets, the less likely it is that a second comer’s borrowing will undermine the first comer’s incentives to invest in innovation”, (ii) (Athreye, 2007) Pág. 8: “Over two-thirds of all the software development effort is spent in maintaining and enhancing the existing software code, rather than in producing new software”, (iii) Proponer que se pueda estudiar, pero no se pueda usar lo que ya se estudió, mínimamente, es una propuesta ingenua: (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1658.

³⁰² (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2378: “Market failure is likely if the quantum and significance of the entity taken is substantial, the second comer’s development effort is rapid, easy, and highly dependent on the first comer’s product, the degree of similarity in the resulting products approaches identity, and the second comer’s market is proximate to that in which the first comer operates”.

³⁰³ (i) (Calabresi, Transaction Costs, Resource Allocation and Liability Rules-A Comment, 1968) Págs. 68 – 69: “The primary implication is that problems of misallocation of resources and externalities are not theoretical but empirical ones. The resource allocation aim is to approximate, both closely and cheaply, the result the market would bring about if bargaining actually where costless”, (ii) Esto mismo sucede cuando la estructura normativa del derecho no puede adaptarse a los cambios (Calabresi, A Common Law for the Age of Statutes, 2009) Pág. 91: “[I]f resolution of a dispute is eased by government intervention through payments, taxes, or detailed regulations, common law resolution becomes much less likely”, (iii) (Cooter & Ulen, 2012) Pág. 89: “There is an extensive literature on bargaining games, including a large number of carefully constructed experiments testing the Coase Theorem. One of the most robust conclusions of these experiments is that bargainers are more likely to cooperate when their rights are clear and less likely to agree when their rights are ambiguous”, (iv) (Aguirre Soriano, 2014) Pág. 103.

³⁰⁴ (i) (Bases del plan nacional de desarrollo 2018 - 2022, 2019) Pág. 756: “El DNP y los demás miembros de la CIPI actualizarán la Política de Propiedad Intelectual”, (ii) Corte Constitucional, sentencia C-16-2016, Págs. 17 – 18: “[E]l Plan de Desarrollo es desde la perspectiva del artículo 334 de la Carta, una de las máximas manifestaciones de la dirección de la economía por parte del Estado o, como lo ha indicado este Tribunal, una ‘expresión suprema de la función de planeación’. Erigiéndose en el ‘esfuerzo del Estado por estructurar una política económica razonada y armónica durante un cierto período’ constituye ‘un presupuesto indispensable para el logro de los objetivos básicos del sistema constitucional’ y, en consecuencia, se trata de ‘un mecanismo ordenador y aglutinador de la política estatal’. La importancia económica de este instrumento se traduce, a su vez, en una especial posición en el sistema de fuentes y en una particular eficacia normativa. En primer lugar, la Carta Política señala que el Plan de Inversiones Públicas debe ser aprobado por una ley que tendrá prelación sobre las demás leyes. Por tal razón esta Corporación ha indicado que ‘una norma que desarrolle materias contenidas en la Ley del Plan Nacional de Desarrollo, no podrá contener disposiciones contrarias a la misma, aunque se trate de disposiciones legales equivalentes, pues de manera expresa la Constitución le ha otorgado prevalencia a Ley del Plan Nacional de Desarrollo’. Incluso este Tribunal ha advertido que ‘una ley que contenga disposiciones contrarias a las contenidas en el Plan Nacional, será una norma contraria a la Constitución’”.

legislar sobre el déficit de protección para el *software*, en el doble propósito de diseñar un nuevo modelo de protección, con normas al mejor estado del arte, y propender por una asignación de derechos mediante la aguda observación de los participantes en el mercado.

De esta forma, los involucrados en la industria (“*stakeholders*”) aprovecharán la certidumbre en los derechos asignados, beneficiando a la sociedad vía el estímulo a la competencia de tipo horizontal³⁰⁵ (“*dynamic competition for the standard*”) y/o vertical (“*dynamic competition within the standard*”) ³⁰⁶, en el mercado de aplicativos, sistemas operativos o plataformas de contenido³⁰⁷. Además, si se concibiera un sistema que organice reglas generales de acceso a la información, tal como sucede con las licencias del *software* de código abierto³⁰⁸, el fortalecimiento en la competencia no se obtendría mediante el “pirata” expediente de la descompilación y/o ingeniería reversa con fines de interoperabilidad³⁰⁹.

Por ende, las nuevas normas jurídicas deberían emular reglas que faciliten el acceso a la información³¹⁰, pero con todo el derecho para que el innovador cobre por su esfuerzo y recupere

³⁰⁵ (Dam, 1995) Pág. 354.

³⁰⁶ (i) Tomamos prestados los términos acuñados por (Drexl, 2011) Pág. 174, (ii) Semejante a tales términos, en el campo de la distribución selectiva, está la llamada competencia “*intra-brand*” e “*inter-brand*”, (Waelbroeck & Frignani, 1998) Pág. 858.

³⁰⁷ Señalando que en el mercado de *software* aplicativo, el incentivo para crearlos no se ve muy afectado por la naturaleza abierta o cerrada del código de la plataforma: (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1622.

³⁰⁸ (i) (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Págs. 1622 – 1623, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2402: “*There are three principal ways that firms presently achieve interoperability. First, some developers publish interface specifications to enable third party developers to construct programs that will interoperate with their programs. Firms have incentives to do this when they think they can increase sales of their software or hardware systems, which will be more in demand if there are many programs that interoperate with their system*”.

³⁰⁹ (Calle D'Alleman, 2012) Pág. 208.

³¹⁰ (Graef, 2014) Pág. 15: “*Because of the uncertainty that reverse engineering involves, more structural measures may be necessary to ensure that interoperability information is available*”.

sus costos antes de que un producto sustituto y/o complementario salga a menor precio en el mercado, como sucede hoy con los clonadores, quienes no incurren en los costos iniciales que sí tienen los innovadores³¹¹.

Si se promulgan normas de este estilo, la (I+D+i) - que es el presupuesto para el desarrollo de la industria - se haría sobre la base explícita del estado del arte anterior, seguramente a la medida de los requerimientos de los usuarios y para una base más amplia de firmas; las cuales estarán en la capacidad de crear productos sustitutos y/o complementarios innovadores los unos de los otros, y no copias baratas del *software* innovador o licencias de código fuente por debajo del precio del mercado³¹².

2.2. Costosa protección del secreto.

No es “secreto” que la protección de éste implica tomar medidas *de facto y jure et de jure* para proteger el *software*, consistentes las primeras en programar medidas de seguridad contra la

³¹¹ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2382 y 2420: “*A reason to distinguish between them is that imitations generally only increase the supply of products embodying the innovation and contribute to price competition. Improvements, on the other hand, benefit the market because they elevate to some degree the overall level of skill in the technical community. As indicated above, giving consideration to improvements would probably be desirable if a substantial similarity standard, rather than a substantial identity standard, was selected as the standard by which to judge when a second comer had unfairly interfered with the market opportunities of an innovative software developer. However, although it is often easy to tell that a second program improves on the original, in some instances it can be difficult to say whether differences from the original are improvements or merely attempts to avoid liability. Of course, the market will generally determine which differences are improvements: consumers will favor an improved version. This measure, however, does not easily distinguish between substantive improvements and the price improvement that all clones bring to the market. However, the fact that improvements are not readily distinguishable from mere differences suggests that this standard could be somewhat troublesome*”.

³¹² Cobrar un precio por licencias del código fuente, por debajo de lo que el mercado puede pagar, podría ser una estrategia elegida por el innovador, a fin de recuperar sus costos de una base extendida de usuarios (Samuelson & Scotchmer, *The law an economics of reverse engineering*, 2002) Pág. 1589.

copia del código fuente³¹³ y en la inversión de recursos en la instalación de severas medidas de seguridad en sus sitios de operación, por decir lo menos; las segundas, celebrando acuerdos de confidencialidad, complicados de elaborar y hacer cumplir³¹⁴. Todos estos sobrecostos son trasladados a los usuarios finales del *software* vía precio de las licencias, cuando no trasladados a los propios creadores de contenidos.

Aparte, el titular del secreto debe emplear recursos para la administración de los contratos de confidencialidad y licencia³¹⁵, desde las etapas de negociación hasta las de seguimiento y eventual exigencia de cumplimiento, más allá de lo normalmente aceptado en otro tipo de relaciones contractuales; pues la contratación a la que nos referimos se da con potenciales competidores³¹⁶, sindicatos de proveedores o entes gubernamentales.

Realmente los costos de transacción que implica conservar secreto el código fuente deben ser muy tenidos en cuenta al momento de repensar un modelo de protección para el *software*, pues sin importar el modelo de negocio empleado (*software* propietario o *software* de código abierto), el innovador gasta importantes fondos en administrar y conservar secreto el código

³¹³ Respecto del alcance del artículo 11 del WCT, y una caracterización de las medidas tecnológicas (Lipszyc, 2004) Págs. 146 – 166.

³¹⁴ (i) (Menell, Tailoring legal protection for computer software, 1987) Págs. 1351 – 1352, (ii) (López, 2009) Pág. 12: “Paallysaho and Kuusisto (2006) studied a sample of Finland and UK firms in three knowledge intensive service sectors: software consultancy and supply, business and management consultancy services and advertising. Most firms were small and medium-sized and their sales came mostly from tailor-made services. As expected, patents were used very little (software firms had a relatively higher rate of use), while trademarks and copyright dominated in the field of formal IPRs. However, by far the most used appropriability mechanism was restrictive contracts (85 per cent of the surveyed firms used contracts, against 36 per cent in the case of trademarks). These contracts included requiring employees to sign non-disclosure agreements or non-competition clauses. The use of legal instruments was often complemented by informal means, such as secrecy (which was widely used by the firms sampled in this study), publishing, restrictions on access to information, enhancing personnel commitment and implementing schemes of fragmentation and rotation of duties”.

³¹⁵ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2373.

³¹⁶ (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1587.

fuente, siendo este algo relativamente fácil de conocer por los entendidos³¹⁷. Debe entonces incentivarse la redirección de recursos hacia la (I+D+i) de nuevos y mejores productos, pues ello resulta siendo más apreciable en términos económicos y sociales³¹⁸, que dedicar fondos y personal para mantener el hermetismo que recubre el código fuente.

2.3. Fuga de talentos.

Por más que se tomen medidas administrativas para conservar secreto el proceso de elaboración de un *software*, lo mismo que su código fuente, por más que se tomen medidas jurídicas para sancionar la violación a un deber de confidencialidad nacido de una relación profesional, y por más que todas estas medidas hayan sido efectivas en mantener oculto “el” algoritmo del *software* protegido; ciertamente no es realista confiar en que los colaboradores en la programación del *software* dejarán de aplicar sus conocimientos adquiridos en programación, para sí o para terceros³¹⁹, solucionando problemas y respondiendo los requerimientos de los usuarios, expresando en forma de *software* modelos de negocios y mercadearlos, si fueran luego desvinculados voluntaria o involuntariamente de la firma innovadora.

³¹⁷ Señalando el cálculo que hace el innovador para conservar inaccesible el secreto, en función de la facilidad en su descubrimiento: (Posner, *Economic analysis of Law*, 2011) Pág. 51.

³¹⁸ (Ascarelli, 1970) Pág. 261.

³¹⁹ (i) (Galán Corona, 2011) Pág. 365 y 370, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pag. 2342 - 2343: “*Although trade secrecy law has a long history of protecting industrial compilations of applied know-how, it cannot protect behavior or other know-how borne on the face of a mass-marketed software product because such know-how cannot be kept a secret. An experienced programmer who runs a program to study its component behaviors can often learn everything necessary to make a functionally indistinguishable program. Nor can trade secrecy law protect know-how borne near the surface of software products, because trade secrecy law has long regarded reverse engineering of products available in the marketplace as a fair means of acquiring trade secrets*”.

El tiempo y dinero empleados por las firmas innovadoras en la formación de equipos de trabajo para la creación del *software*, o sus derivaciones, generalmente NO es recuperado con la comercialización de los productos informáticos que hubieren creado y peor aún, el remplazo del personal programador puede llegar a ser imposible, condenando a las firmas innovadoras a constantes capacitaciones de nuevo personal, gracias a la movilidad y baja oferta laboral que caracteriza esta industria³²⁰.

No podría señalar cuán visionarios de estos problemas fueron quienes desde los albores de la industria, en las normas modelo para la protección del *software* preparadas a instancias de la OMPI, propusieron un reforzamiento de las normas de competencia desleal o secretos industriales que estuvieren disponibles; pero considero aún vigentes los motivos que se expresaron en favor de tales reglas de protección reforzadas, políticamente justificadas de formas resumidas en los comentarios a la sección 5 de las normas modelo³²¹ publicadas en 1978.

³²⁰ (i) El Tiempo. 19/12/15: *En el sector TI hay más empleo que profesionales. El país necesitará unos 93.000 desarrolladores de 'software' en el 2018. Empleabilidad es del 98%: “Hace 10 años me tomaba un mes encontrar a un programador; pero ahora, son unos cuatro meses”, contó Hans Christian Boehlke, líder de Aldeamo, una firma de soluciones para dispositivos móviles*”, (ii) La República. 18/12/17: *“De acuerdo con datos del Observatorio TI, la estimación de la brecha de talento TI en el país es de 45.520 profesionales para 2018; esto implica que Colombia necesita de este recurso humano altamente cualificado y con las competencias priorizadas por la industria TI”*, (iii) El País. 8/7/19: *“Popayán, ciudad donde escasean las oportunidades de empleo, tiene ahora en el Centro de Desarrollo e Innovación de Software, una plataforma de formación y trabajo para jóvenes de estratos 1, 2 y 3, gracias a una alianza entre el Sena y Everis, una compañía desarrolladora de sistemas de información. Juan Sebastián Escobar, socio y director de la unidad de negocio ITS&S de Everis Colombia, cuenta lo que los llevó a crear este centro en Popayán, fue el déficit enorme de profesionales asociados a la ingeniería de software”*. (iv) Apostando a un programa de compensaciones “*off set*” para la capacitación de personal en sectores “estratégicos”, como el del *software*: (Conpes 3582, 2009) Pág. 46, (v) Un estudio sobre las características de la contratación laboral en el sector IT: (Allen, Armstrong, Reid, & Riemenschneider, 2009) Pág. 98: *“Increasingly the IT field is having difficulty recruiting and retaining employees. Part of that difficulty may be due to a mismatch between expectations for a career in IT when compared to the reality of the IT workplace”*, (vi) (Informe Nacional de Competitividad 2018 - 2019, 2018) Pág. 393: *“Esto significa que el crecimiento del sector TIC en las últimas décadas no ha sido acompañado por una dinámica equivalente en la oferta de profesionales relacionados con el área de las TIC”*, (vii) (Bases del plan nacional de desarrollo 2018 - 2022, 2019) Pág. 646: *“El capital humano calificado es uno de los principales factores que un país requiere para insertarse en la economía digital. Colombia tiene grandes desafíos en esta área para que la población se empodere del entorno digital”*.

³²¹ (Model provisions on the protection of computer software, 1978) Pág. 17: *“[P]rotection they provide may already be available in a country adopting the model provisions, under its law on trade secrets, breach of confidence or data*

Nos inspiramos en ellas, a fin de hacer contemporáneos algunos remedios que proponemos en el capítulo V, los cuales suponen una política pública que incentiva la estabilidad laboral del personal especializado; pero dicha política pública deberá complementarse con otros instrumentos o incentivos para atender la agravante escasez de oferta laboral, mediante el diseño de planes y programas orientados a la formación profesional de la fuerza laboral requerida³²².

trespass. Provisions of this kind would be useful for countries whose laws do not provide sufficient protection against the acts covered; computer software is in particular need of such protection because of its vulnerability and the difficulty of detecting its use after misappropriation”.

³²² En este sentido: (Bases del plan nacional de desarrollo 2018 - 2022, 2019) Pág. 652: “MinEducación, en conjunto con CPE, impulsará la transformación de las prácticas de enseñanza a través del desarrollo de competencias tecnológicas, pedagógicas, comunicativas, investigativas y de gestión, por medio del desarrollo de competencias docentes para la enseñanza y el aprendizaje, de cara a la transformación digital y el fortalecimiento del uso pedagógico de los contenidos educativos digitales, especialmente aquellos relacionados con las áreas de matemáticas, ciencia y tecnología y para la programación de dispositivos electrónicos. Esto implica que previamente se consolide un programa de formación docente que genere las capacidades”.

**Capítulo IV. Hagamos lo siguiente: Marco normativo para un nuevo modelo de
protección**

La lectura de las iniciativas legislativas y casos jurisprudenciales antes reseñados, permiten concluir que la preocupación sobre el modelo adecuado de protección para el *software* llama la atención de la academia³²³, los gremios³²⁴ y las autoridades gubernamentales³²⁵, pese al actual conformismo normativo nacional e internacional.

Pero, todo el entorno anteriormente explicado hace prever que la dinámica propia de la industria del *software*, debido a su ritmo de crecimiento³²⁶, pese a caracterizarse por su actual atomismo³²⁷ y desarticulación gubernamental³²⁸, empezará a reclamar de las autoridades

³²³ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2364 – 2365, (ii) (Dam, 1995) Pág. 331, (iii) (Menell, The challenges of reforming intellectual property protection for computer software, 1994) Pág. 2644, (iv) (Nimmer & Krauthaus, 1987) Pág. 734, (v) (Aguirre Soriano, 2014) Pág. 113, (vi) (Carranza Torres, 2008) Pág. 206, (vii) (Castán Pérez-Gómez, 1999) Págs. 115 y 119, (viii) (Swinson, 1991) Pág. 145, (ix) (Salas Pasuy, 2016) Pág. 437, (x) (Leith, 2007) Pág. 159, (xi) (Calle D'Alleman, 2012) Pag. 99: “*Lo dicho hasta ahora revela la necesidad de una noción diversa y ajena a estas categorías tradicionales de la propiedad intelectual que ahora sobrepasan sus linderos a través del fenómeno informático y su inclusión en el Derecho desde una perspectiva diferente*”, (xii) (Palacio Eastman, 1999) Pag. 220: “*Además, el mismo decreto [refiriéndose al decreto 1360/89] permite que en Colombia se cree un régimen sui generis para el software, pues en su artículo 71 establece: ‘Artículo 71. La protección que otorga el derecho de autor al soporte lógico (software) no excluye otras formas de protección por el derecho común’*”.

³²⁴ (i) (Caracterización del sector de teleinformática, software y TI en Colombia. Año 2015. , 2016) Página 224, (ii) (Smith & Mann, 2004) Pág. 242, (iii) (Castán Pérez-Gómez, 1999) Página 115, (iv) (Informe Nacional de Competitividad 2018 - 2019, 2018) Pág. 390.

³²⁵ En Julio de 2015 se ha iniciado a discutir una modernización de la Decisión 486, tanto en el III Seminario Internacional de Propiedad Intelectual (Julio de 2015), como en la Conmemoración de los 15 años de la Decisión 486 (Agosto de 2015). Si bien no se han escuchado aún voces gubernamentales resueltas a renovar la actual protección del *software*, esta cíclica primavera de discusión política debe aprovecharse también para registrar e implementar todo lo que durante dos décadas se ha escrito sobre el particular, lo cual permite llegar a muchas e interesantes conclusiones sobre la ventaja de catapultar la industria del *software* para obtener resultados más elevados en productividad y empleo. En este sentido: (i) (The software industry and developing countries, 2012) Pag. 55 y 100, (ii) (Buitrago, 2007) Pág. 159, (iii) Anticipando una reforma al Tratado de la Patente Europea, que no se dio: (Botana Agra, 2001) Pág. 170. (iv) Una queja respecto del ilógico régimen de protección para el *software*: (Laboratorios California S.A. v. System Software Associates Inc. y S.S.A. Colombia S.A., 2001), (v) Describiendo una propuesta francesa: (Calle D'Alleman, 2012) Pág. 78, (vi) Mencionando una propuesta de la Comisión Europea, del año 2002: (Uribe Arbeláez, 2005) Pág. 209.

³²⁶ Esta es una industria que en muchos mercados crece de forma sostenida, año a año, dos dígitos porcentuales. En este sentido: (i) (Caracterización del sector de teleinformática, software y TI en Colombia. Año 2015. , 2016) (ii) (Castán Pérez-Gómez, 1999) Página 116.

³²⁷ (i) (Heshusius Rodríguez, 2009) Página 139: “*La industria de software en Colombia se encuentra bastante desarticulada*”, (ii) (Conpes 3582, 2009) Pág. 31: “*En el país, la falta de focalización ha generado dispersión de recursos*”.

³²⁸ Documento Proexport de marzo de 2004, accesible en:

<http://www.proexport.gov.co/vbecontent/NewsDetail.asp?ID=1458&IDCompany=16>. De igual forma, los documentos del Ministerio de Comercio, Industria y Turismo de agosto de 2007, accesibles en:

<http://www.mincomercio.gov.co/eContent/documentos/Eventoshome/2007/TallerConvalidacionEstudioPrevision1.p>

colombianas una protección más agresiva y efectiva para el *software*³²⁹; ya que el marco normativo de predilección es el Derechos de Autor y la Protección a Secretos Empresariales, los cuales son de poca utilidad práctica para su desarrollo a gran escala, pues absuelven a los “clonadores” bajo el argumento de los trabajos independientes³³⁰.

¿Será que dejamos así?... El innovador inconforme con el grado de protección que recibe su creación, los manufactureros aprovechando el Sistema de Patentes y la seguridad jurídica a merced de interpretaciones³³¹, en procura de sostener jurídicamente un tripartito³³² e ineficiente sistema de protección para el *software*³³³?...

df. y

<http://www.mincomercio.gov.co/eContent/documentos/Eventoshome/2007/TallerConvalidacionEstudioSoftware.pdf>

³²⁹ (The software industry and developing countries, 2012) Pág. 99: “*The relevance of intellectual property policies and their enforcement grows as the capabilities of local software developers and enterprises improve (UNCTAD, 2002). In the early stages of development, when most indigenous software firms are engaged in relatively simple software services for the local market, the protection of intellectual property may be of limited concern*”.

³³⁰ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2380, (ii) (Calle D'Alleman, 2012) Página 307: “[C]uando el Estado colombiano toma la rienda de conducir la sociedad hacia la informatización, es posible constatar que los intereses alrededor del software son diversos y más amplios de los que suponen proteger una industria que en Colombia apenas podría comenzar a surgir, pero sobre parámetros jurídicos sólidos, distintos a los que lograron surgir en años anteriores”.

³³¹ (i) (Rengifo García, 2003) Pág. 216, (ii) (As Colombia Ltda. v. Informática y Gestión S.A, 2006) Pág. 32, (iii) (Dam, 1995) Pág. 364, (iv) (Teruo, 1992) Pág. 97, (v) (Aguirre Soriano, 2014) Pág. 199, (vi) (Samuelson & Scotchmer, The law and economics of reverse engineering, 2002) Pág. 1611, (vii) Opinando sobre la aplicación de la doctrina del “*fair use*” en casos judiciales: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2391: “*Two federal courts of appeals have decided that making intermediate copies of programs is fair use when done for a legitimate purpose, such as to gain access to information necessary to develop a program that can interoperate with other programs or with hardware. While these rulings are unexceptionable as a matter of copyright and trade secrecy law, they may leave software internals more vulnerable to market-destructive appropriations than is desirable*”

³³² Algunos llaman “*acumulada*” (Castán Pérez-Gómez, 1999) Página 121.

³³³ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2393 - 2394: “*Reimplementing another program's functionality is not, in our view, a copyright, patent, or trade secret misappropriation. However, it can have market-destructive effects because of the relative ease and low cost with which a skilled engineer can appropriate the know-how required to produce certain behavior by watching another firm's program in action. If the engineer thereafter makes a functionally indistinguishable program that arrives in the market very quickly, it will undercut the ability of the first firm to recoup its considerable research and development expenses*”.

No consideramos conveniente, ni para Colombia ni para el mundo, la actual disparidad de criterios frente a la protección concurrente del *software* vía Derechos de Autor, Sistema de Patentes y Protección a Secretos Empresariales³³⁴, pues esto ha generado la compartimentación de mercados, a escala nacional, regional y global³³⁵, y peor aún, disminuye el grado y velocidad con que se innova en este campo³³⁶.

En efecto, conservar un amparo para el *software* vía Derechos de Autor, Sistema de Patentes y Protección a Secretos Empresariales, a elección del titular de la tecnología, es aumentar costos de transacción para innovadores como se ha explicado en los capítulos anteriores, lo cual es aprovechado para discriminar las prestaciones habilitadas entre los usuarios del *software*, y fundamentalmente en el supuesto de acciones judiciales por infracciones a bienes jurídicos diferentes, con términos de protección acumulables³³⁷; esto es, la posibilidad que tiene el titular de la tecnología en poder demandar simultáneamente por infracción a derechos de autor, infracción a derechos de patente y competencia desleal por acto de violación de secretos empresariales, cuyas reglamentaciones sustanciales y procesales difieren en bienes jurídicos protegidos y plazos de prescripción³³⁸.

Si el Derecho de Autor le queda muy chico al *software*³³⁹, el Sistema de Patentes grande³⁴⁰ y la Protección a Secretos Empresariales es la combinación de lo más malo de los

³³⁴ Concordamos con: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2421.

³³⁵ (Graef, 2014) Pág. 18.

³³⁶ (Menell, Tailoring legal protection for computer software, 1987) Pág. 1330.

³³⁷ (i) (Paterson, 2013) Pág. 181, (ii) (Castán Pérez-Gómez, 1999) Pág. 121.

³³⁸ Arts. 244 Decisión 486/00, 23 Ley 256/96 y 2536 del Código Civil.

³³⁹ (Palacio Eastman, 1999) Pág. 213.

³⁴⁰ Resaltando una propuesta europea que concebía al *software* como un modelo de utilidad, pero criticando su alcance: (Leith, 2007) Págs. 173 y 178 – 180.

anteriores³⁴¹, definitivamente debe procurarse una mejor³⁴² reglamentación para el *software* que combine lo rescatable de cada uno de los cuerpos normativos corrientes³⁴³, sin desconocer el conjunto de Tratados internacionales pertinentes.

1. ¿Por qué no despegaba una protección *sui generis* para el *software*?

La idea de una protección *sui generis*³⁴⁴ para el *software* está sobre la mesa desde los 70's, cuando se empezó a responder a las primeras preguntas que llegaban sobre el *software*. Así, la propuesta de protección *sui generis* del comité nombrado por el Ministerio de Industria y Comercio Internacional del Japón, que textualmente propuso la inaplicación del sistema de *copyright* para el *software*³⁴⁵; aunque finalmente dicho comité acordó desistir de la propuesta con la Agencia de Asuntos Culturales del Japón, la cual presentó en 1985 una modificación a la Ley de Derechos de Autor de 1970, básicamente ajustando la legislación vigente a las realidades litigiosas del momento, provocadas por los empresarios del *software* y los video juegos³⁴⁶.

³⁴¹ Analizando el resultado de otro estudio, se pone de presente la insatisfacción de la propia industria respecto de los modelos de protección actuales para el *software*: (López, 2009) Pág. 10.

³⁴² (Sen, 2006) Pág. 223.

³⁴³ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2365: [E]xisting legal regimes are not well-suited to protecting software innovation. They either cannot protect the principal source of value in programs behavior or cannot protect the incremental innovation typical in the field. Trade secrecy cannot protect what is not secret; copyright protects only text and text is largely independent of behavior; and incremental innovation cannot meet patent standards. Attempts to stretch the bounds of existing regimes to protect the incremental innovation in software will result in either too much or too little legal protection”.

³⁴⁴ “*industrial copyright*”: (Nguyen , Gomulkiewicz , & Conway , 2012) Pág. 33: “The *sui generis* protection was called ‘*industrial copyright*’, and should be established at the intersecting part of copyright and industrial property”

³⁴⁵ (i) (Nimmer & Krauthaus, 1987) Pág. 752, (ii) (Palacio Eastman, 1999) Pág. 215: “La OMPI en 1978 creó unas disposiciones tipo para que sirviera de modelo a las legislaciones nacionales. En 1973 el Ministerio de Comercio Exterior e Industrias del Japón, MITI, propuso una legislación *sui generis* con un plazo de protección de 15 años. A pesar de lo anterior, actualmente se protege en Japón vía derecho de autor”.

³⁴⁶ En el caso de *Namco v. Suishin Kogyo*, el demandado instaló fraudulentamente unas máquinas del juego PAC-MAN en sus tiendas de café, provocando que el demandante alegara una infracción al derecho de comunicación pública de la obra audiovisual bajo el reglamento de la Ley de Derecho de Autor japonés, que ampara la exposición visual o audiovisual de la obra original que hubiere sido fijada en un soporte material, requerimientos cumplidos en el caso concreto y en consecuencia susceptible de protección por las reglas del Derecho de Autor. De esta forma, el

Lamentablemente esa visión futurista no pudo rescatarse para el momento en que se celebraron las conferencias diplomáticas que resultaron con los Tratados OMPI³⁴⁷ y la Convención de Washington sobre la Propiedad Industrial respecto de los Circuitos Integrados³⁴⁸, a efectos de legislar sobre la protección del *software*, pues sucedieron precisamente cuando empezaron a darse en los Estados Unidos de América las sentencias más discutidas sobre protección vía Derechos de Autor y el Sistema de Patentes, y ya que las propuestas académicas demoraron en consolidarse³⁴⁹. Siendo un principio respetado y apropiado del legislador esperar a que la jurisprudencia se dicte antes de legislar³⁵⁰, la discusión de un modelo de protección *sui generis* para el *software* más que aplazada, quedó entonces descartada.

software inmerso en el videojuego se protegió por la vía de la obra audiovisual... Curioso, no? (i) Al respecto de la protección de los videojuegos como obras audiovisuales: (Carranza Torres, 2008) Pág. 264, (ii) Poniendo de presente una nueva iniciativa japonesa para reconciliar los intereses de la industria con las normas jurídicas: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2421 – 2422.

³⁴⁷ (i) Nos referimos al “WCT”, no al borrador de Tratado de Protección para el Software o Normas Modelo de Protección para el *software* de 1983, como se habla en la doctrina (Nimmer & Krauthaus, 1987) Págs. 750 – 751, (ii) (Lipszyc, 2004) Pág. 118 – 122.

³⁴⁸ Esta Convención no ha entrado en vigor, pero sus principios y algunas reglas, se han incorporado en ADPIC y en la Decisión 486 de la Comisión de la Comunidad Andina de Naciones. (i) Recomendando seguir tal Convención, como base para una protección *sui generis* del *software*: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2400, (ii) Explicando que esta Convención no ha trascendido, debido a que permite una ingeniería reversa para empresas sucesoras y en consecuencia NO debería servir de base para una protección *sui generis*: (Dam, 1995) Pág. 373 y 375.

³⁴⁹ En general, el análisis académico de la protección del *software* se da después de analizados los efectos de las patentes de *software*, en contraste con la protección vía Derechos de Autor. Por ejemplo: (i) (Menell, Tailoring legal protection for computer software, 1987) Págs. 1364 – 1365, (ii) (Leith, 2007) Repitiendo un objetivo de mejorar el acceso a la información, y proteger el adelanto que supone el *software* Pág. 160 y 178, (iii) Proponiendo incluso un derecho, de confusos e ilimitados contornos, para el creador y el usuario: (Calle D'Alleman, 2012) Págs. 270, 279, 300 y 301, (iv) Poniendo sobre la mesa un listado de circunstancias consideradas admisibles para la ingeniería reversa: (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1651, (v) Lanzando la idea de encontrar un régimen entre el Derecho de Autor y el Sistema de Patentes: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2399.

³⁵⁰ Aunque algunos jueces consideran que las Cortes no tienen la capacidad para procesar la información que procure los cambios normativos requeridos, a la velocidad adecuada, (Calabresi, A Common Law for the Age of Statutes, 2009) Pág. 163.

Si a la falta de *momentum*³⁵¹ se suma que los modelos de protección *sui generis* para el *software* se concibieron teniendo en mente el *software* propietario, al cual le sobrevino en paralelo un modelo de negocio basado en *software* de código abierto, que aceleró la rapidez y obviedad con la cual se codifica, simplemente las ideas de un modelo de protección *sui generis* quedaron más que descontextualizadas.

Consideramos que el ambiente histórico jugó un papel muy importante en la postergación de la decisión sobre el mejor modelo de protección para el *software*, por razones que precisaremos a continuación, pero superadas esas circunstancias creemos que el tema ha vuelto a escalar posiciones en el orden del día local y mundial, pues tenemos a disposición la información necesaria para corregir el rumbo³⁵².

1.1. Las leyes especiales de protección para el *software* y el *cloning*.

No importa que el propósito constitucional sea incentivar la producción intelectual (Como en Estados Unidos de América)³⁵³ y/o el reconocimiento de derechos a favor de los creadores (Como en Colombia)³⁵⁴, las propiedades del teorema de Coase se ven más que confirmadas cada vez que se definen con mayor precisión los derechos asignados a los sujetos de derecho, como es el caso de las leyes especiales para la protección del *software* que han sido

³⁵¹ (Menell, The challenges of reforming intellectual property protection for computer software, 1994) Pág. 2653.

³⁵² (Menell, The challenges of reforming intellectual property protection for computer software, 1994) Pág. 2648.

³⁵³ (i) Art. 1-8-8 de la Constitución de los Estados Unidos de América, (ii) (Samuelson & Scotchmer, The law and economics of reverse engineering, 2002) Pág. 1581,

³⁵⁴ (i) Artículos 61 y 71 de la Constitución Política Nacional, (ii) (Plata López, 2010) Pág. 190, (iii) Considerando que las reglas de la propiedad intelectual se ajustan mejor a un sistema de asignación de derechos, que a reglas de responsabilidad civil: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2369.

promulgadas a lo largo y ancho del planeta³⁵⁵; esto es, como lo recogen los estudios especializados, que el refinamiento de un derecho impulsa las industrias correspondientes, lo cual para el caso serían las dedicadas a la elaboración de *software*³⁵⁶.

No obstante, las leyes especiales para la protección del *software* vinieron inoculadas del patógeno Derecho de Autor³⁵⁷, o quien sabe si al Derecho de Autor se le agravó la infección de *software* que venía combatiendo, pues si las leyes especiales para la protección del *software* hubieran detenido las patentes de *software*, entonces tales leyes habrían justificado la infección del Derecho de Autor, porque lo habrían convertido en el súper huésped que detuvo un mal peor; pero ni lo uno, ni lo otro.

Originalidad, dicotomía entre idea y expresión de la idea, derechos morales de autor, término de protección y permisión de trabajos independientes fueron perezosamente inabordados por estas leyes, básicamente de finales de los 80's y principios de los 90's³⁵⁸, aunque reseñamos la excepción de interoperabilidad y la inclusión de registros; pero como hemos denunciado previamente, ambos puntos a favor, quedaron a “medio hacer”. El primero, porque no conjuga con los derechos morales del autor, y crea el riesgo moral de aprovechar la excepción para fines

³⁵⁵ (Dukrok & Hwang, 2010) Pág. 381.

³⁵⁶ (Dukrok & Hwang, 2010) Págs. 381 y 385.

³⁵⁷ (i) Señalando que en un primer momento las leyes especiales parecían más una reglamentación *sui generis* debido a la cantidad de excepciones alejadas del Derecho de Autor: (Gómez Segade, 2001) Pág. 848, (ii) (Nguyen , Gomulkiewicz , & Conway , 2012) Pág. 33: “[T]he international situation in software legislation changed greatly during this period. The drafters heard that Brazil and South Korea, which had at one time been regarded as models for establishing independent software protection outside the copyright system, had included software as copyright subject-matter around 1987. In early 1988, the former Soviet Union, the sole country which had declared it was preparing an independent software protection law, also followed the copyright trail. If China persisted with drafting a specific software protection system, it might find itself isolated in this field”.

³⁵⁸ Una crítica de los defectos de la CPPL, por venir inspirada en la ley de Derechos de Autor: (Byungkwon, 1989) Pág. 192.

diferentes al diseñado; el segundo, porque se trata de registros con meros fines de oponibilidad³⁵⁹ y no constitutivos de derechos³⁶⁰, como es natural en el marco del Derecho de Autor.

En otras palabras, las leyes especiales del *software* pasaron por alto el fenómeno del *cloning*³⁶¹, también como es natural, porque teóricamente el Derecho de Autor no se ocupa de las creaciones utilitarias, y convulsiona con los problemas que le vienen con el *software*, como la externalidad de redes, la deseada compatibilidad y los peligrosos *standard de facto*.

Luego, legislar una reglamentación de derechos sobre el *software* en el marco del Derecho de Autor es una opción diplomáticamente impecable a los ojos del mundo³⁶², pero intrascendente en la industria y política económica; todo lo contrario a lo que fue legislar, a nivel nacional y mundial, contra el *piracy*³⁶³.

³⁵⁹ (Plata López, 2010) Pág. 192.

³⁶⁰ (Nguyen , Gomulkiewicz , & Conway , 2012) Pág. 37.

³⁶¹ (Nguyen , Gomulkiewicz , & Conway , 2012) Pág. 33.

³⁶² Señalando que bajo ADPIC la protección por Derechos de Autor es la máxima protección a la “dialéctica” del *software*: (Charfoos, 2002) Pág. 12.

³⁶³ (i) Asegurando que la óptima protección dispensada a un tipo de obra se debe aumentar, cuando se amplía la demanda de las obras amparadas y se reducen los costos para realizar copias: (Landes & Posner, 1989) Pág. 344. Consideramos que este razonamiento es extensible al combate del *cloning*. (ii) (The software industry and developing countries, 2012) Pág. 100: “*In India, stricter enforcement of intellectual property protection had an impact on software piracy and helped the country to attract more offshore software product development. Changes to the Indian Copy Right Law in 1994 made it illegal to make or distribute copies of copyrighted software and therefore punishable. In addition, the Government, in cooperation with NASSCOM, conducted regular anti-piracy raids to discourage software piracy. In Brazil, increased enforcement of intellectual property protections also made the market more attractive for software product development (UNCTAD, 2002)*”.

1.2. La realidad del *software* de código abierto.

El *software* de código abierto ha dejado perplejos a juristas, economistas y políticos³⁶⁴, y silenciado a los que deberían ser directamente afectados por su incursión, ya que ninguno de los primeros formula explicaciones convincentes del por qué³⁶⁵ prospera la economía colaborativa aplicada en la industria del *software*, en el sector del *software* de código abierto, por qué sería bueno o malo incentivarlo; mientras los directamente interesados se adaptan y sacan provecho de un mercado que les exige tanto productos de *software* propietario, como de *software* con código abierto.

Es un hecho que las firmas que crean *software* tienen un pie en la elaboración de *software* propietario y otro en el *software* de código abierto, por una razón absolutamente comprensible: El cliente les pide productos en ambas presentaciones³⁶⁶; lo que asombra es la incoherencia que predomina en las firmas de esta industria, al exigir un derecho de exclusiva para ambas

³⁶⁴ (Posner, Economic analysis of Law, 2011) Pág. 110: “*Why would a skilled programmer contribute his time, free of charge, to creating open-source code when he could sell his time to a commercial software producer, such as Microsoft? The answer is not entirely clear, but there do appear to be a economic motivations for contributing to open-source code... What the open-source movement shows is not the inadequacy of economics to explain behavior, but the variety of economic incentives that determine behavior*”.

³⁶⁵ (i) Sugiriendo, tímidamente, que el comportamiento económicamente “irracional”, que exponen los movimientos de código abierto, es debido a la mera satisfacción proveniente de la creatividad (Paterson, 2013) Pág. 191: “*Within the programming community there has been a strong movement to promote free or ‘open source’ software. These organizations subscribe to an ideal that questions whether the motivation for programmers to create software is that the software will be afforded intellectual property protection (especially copyright). In turn, these organization believe that ‘programmers are motivated by other factors and will produce the most (and best) software if all software is under a free [license]’*”, (ii) Opinando que el incentivo clásico de la propiedad intelectual aplica para las firmas dedicadas a la creación de *software*: (Dam, 1995) Pág. 335.

³⁶⁶ (Peña Valenzuela, 2013) Pág. 82: “*La Asociación Colombiana de Ingenieros de Sistemas (ACIS) realizó en noviembre de 2004 una investigación circunscrita a la participación del software libre en el país, donde se estableció que el 75% de las empresas encuestadas utiliza software libre, de las cuales el 74% aproximadamente, afirma tener una experiencia positiva con este tipo de software*”.

presentaciones³⁶⁷ - Esta conducta explica, en buena medida, por qué el secreto empresarial es el método de protección más empleado en la industria - .

Claramente el oportunismo de las firmas es censurable para el caso del *software* de código abierto³⁶⁸, pero vale decir que la concurrencia de modelos de protección es lo que les conduce a esas posiciones³⁶⁹, y ha sido el propio Estado el que miopemente hizo crecer el problema. Es decir, como buena parte de la materia prima del creador de *software* bajo licencia de código abierto es gratis³⁷⁰, no es raro que la mayor facturación de estos emprendedores en ciertos países provenga de la creación de *software* a la medida en lenguajes de programación de código abierto³⁷¹ y en consecuencia, el creador de *software* de este mercado no se ve directamente interesado por una respuesta al déficit de protección para el *software*; hasta el momento en que se percata que puede patentar el *software* para detener las firmas que crean productos sustitutos y/o complementarios, en clara traición a las bases fundamentales del movimiento de *software* de código abierto³⁷² que le donó el lenguaje de programación y líneas de código previamente explotadas.

³⁶⁷ Como trabajo de campo del presente trabajo se hizo una encuesta pública entre programadores y firmas de la industria del *software*, en la cual se les preguntó si permitían el libre acceso al código fuente del *software* que elaboran en lenguaje de programación de código abierto. El 57% de los encuestados respondió que no.

³⁶⁸ (i) (Uribe Arbeláez, 2005) Pág. 214, (ii) Aunque en el caso del *software* propietario, protegido por el Derecho de Autor, el innovador también es oportunista al tomar de adelantos precedentes pero exigir máxima protección a su creación: (Landes & Posner, 1989) Pág. 333.

³⁶⁹ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2310 – 2311.

³⁷⁰ Advirtiendo que una licencia tipo BSD no permite la creación de obras derivadas: (Nguyen , Gomulkiewicz , & Conway , 2012) Pág. 530.

³⁷¹ (i) (The software industry and developing countries, 2012) Pág. 68, (ii) (Caracterización del sector de teleinformática, software y TI en Colombia. Año 2015. , 2016) Pág. 45.

³⁷² (i) Este caso trata la reserva de derechos por parte del autor de un *software* escrito en lenguaje de código abierto: (Jcobson v. Katzer, 2008), (ii) (Nguyen , Gomulkiewicz , & Conway , 2012) Págs. 529 – 530, (iii) Poniendo de presente que algunas patentes de *software* concedidas podrían usarse en contra de la base esencial de Linux: (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 302, y además: Pág. 304: “[I]n *Computer Associates v. Quest Software*, 333 F Supp. 2d 688 (N.D. Ill.2004). There the court held that the use of a open-source software development tool did not prevent the user from claiming rights in software written with the tool. Only modified versions of the development tool itself, not the output of the tool, were subject to the GPL”.

Si a lo anterior se añade que algunas organizaciones gubernamentales se casaron con el *software* de código abierto³⁷³, prácticamente creando un jugoso monopsonio para firmas del mercado relevante³⁷⁴, el reclamo al déficit de protección jurídica del *software* se hace en los estrados judiciales, pero no en la prensa. El resultado de la contratación pública y el *software* creado bajo licencias de código abierto es una lenta velocidad para la innovación en esta industria³⁷⁵, pues las firmas del sector no tienen incentivo para tomar el riesgo de crear nuevos sistemas operativos o programas aplicativos, ya que basta replicar lo conocido para ganar buenos ingresos; con el contradictorio agravante impuesto a estas firmas, a las cuales normalmente el contrato público les exige una cesión de derechos – Seguramente porque se pensará en forma errónea que no hay acceso al código fuente de un *software* creado en lenguaje de programación de código abierto - .

Aparte, “pescando en río revuelto” está la industria manufacturera, que ha capitalizado los avances en la codificación abierta, fondeada por dineros públicos, mediante patentes de invención de *hardware* que en muchos casos incluyen este tipo de *software* de código abierto³⁷⁶,

³⁷³ (i) (Uribe Arbeláez, 2005) Págs. 219 y 221, (ii) (Junqueira Botelho, Stefanuto, & Veloso, 2007) Pag. 117: “Another area where the Brazilian government is a world leader is in the adoption of ‘free’ software such as Linux. At least eleven Brazilian cities have passed laws giving preference to or requiring the use of ‘software livre’ and a number of other municipalities, states, and the national government have considered similar legislation”, (iii) (The software industry and developing countries, 2012) Pág. 64. (iv) En este sentido los artículos 147-5 y 148 de la Ley 1955/19 (Plan Nacional de Desarrollo 2018- 2022 “Pacto por Colombia, pacto por la equidad).

³⁷⁴ (Peña Valenzuela, 2013) Págs. 81 - 82: “A pesar de no tener un índice significativo en el establecimiento de nuestro PIB, sí lo es para nuestra industria. Así, se encontró que el 54,4% de las soluciones incorporadas en el sector público corresponde a desarrollos locales, en tanto que en el sector productivo, con excepción de microestablecimientos, dicha participación está entre el 41,1% (servicios) y el 46,6% (industria)”.

³⁷⁵ Si bien la creación de *software* bajo términos de licencia GPL son elaborados con mayor rapidez, no es menos cierto que el salto cualitativo de la tecnología ocurre a paso de babosa, porque las licencias de código abierto implican una renuncia a los derechos de propiedad intelectual. Señalando que el valor social de la propiedad intelectual se evidencia por el efecto que provoca en la velocidad con la que se crean nuevos productos: (Menell, Tailoring legal protection for computer software, 1987) Pág. 1330.

³⁷⁶ (i) (Caracterización del sector de teleinformática, software y TI en Colombia. Año 2015. , 2016) Pág. 213, (ii) Señalando que no ocurre esto: (Menell, Tailoring legal protection for computer software, 1987) Pág. 1362.

accesible públicamente y obvio para el entendido en la materia.

Mejor dicho, era previsible que una errónea decisión política sobre el modelo de protección para el *software* trajera consigo reacciones igualmente erróneas en el campo jurídico, y resultados indeseados en lo económico; luego no hay por qué sorprenderse con el oportunismo de las firmas de la industria, lo que se debe decidir es si patrocinamos el *status quo*³⁷⁷ en vacilante equilibrio entre costos y beneficios, o corregimos hábil y contemporáneamente el déficit de protección jurídica del *software* – Nos apuntamos a lo segundo como a continuación se expone -.

2. Características de un modelo *sui generis* de protección para el *software*

El papel gubernamental no puede estar enteramente dictado por la industria correspondiente, pues la identificación de un déficit de protección para el *software* no se comprueba por el grado de alevosía con que se reclame por los directos interesados, ni por la definición contractual de los derechos que sus creadores disponen, sino por el fundamento político, económico y jurídico que recoge una norma jurídica cuando es debatida³⁷⁸.

³⁷⁷ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2364 – 2365.

³⁷⁸ (i) (Conpes 3582, 2009) Pág. 39: “La última estrategia es consolidar el sistema de propiedad intelectual mediante el desarrollo de los lineamientos formulados por el Conpes 3533 de ‘Bases de un plan de acción para la adecuación del sistema de propiedad intelectual a la competitividad y productividad nacional 2008-2010’, de tal forma que los empresarios encuentren recompensa para sus esfuerzos de innovación y aprovechen la propiedad intelectual para potenciar procesos de transferencia de tecnología. En este sentido, las acciones específicas apuntan, en primer lugar, a fortalecer la función de divulgación de derechos de propiedad intelectual de las entidades del gobierno mediante la realización de seminarios, talleres y cartillas sobre la institucionalidad y legislación de propiedad intelectual vigente nacional e internacionalmente. En segundo lugar, es necesario apoyar a los investigadores y sus instituciones en la protección y explotación de los resultados de investigación, lo cual incluye no solo financiamiento de los procesos ante las autoridades de propiedad intelectual sino también capacitación de investigadores para que

En otras palabras, nada tiene que ver con la respuesta al problema del déficit de protección para el *software* el que muchos creadores se encuentren silenciados por sus ingresos provenientes de la contratación pública, y silenciados también por aprovechar materia prima gratuita, si los beneficios generales y propósitos normativos de la propiedad intelectual³⁷⁹ se ven traicionados en la práctica. Es allí cuando la intervención legislativa, apoyados en la jurisprudencia nacional e internacional, lo mismo que por el aporte de la academia, debe hacer presencia con normas jurídicas³⁸⁰ que emulen ciertas dinámicas del mercado y aseguren los fines políticos pretendidos por la propiedad intelectual³⁸¹.

adquieran los conocimientos necesarios para gestionar patentes, contratos de licencia, gestión y comercialización de resultados de investigación. Estas acciones se complementan a través de la financiación de planes de negocio basados en explotación comercial de activos intangibles”, (ii) (Rawls, 2010) Pág. 326: “Una ley o un programa es suficientemente justo, o al menos no es injusto, si, cuando tratamos de imaginar cómo funcionaría el procedimiento ideal, decidimos que la mayoría de quienes toman parte en ese procedimiento y cumplen sus estipulaciones estarían en favor de esta ley o de este programa. En el procedimiento ideal, la decisión legislativa ha de concebirse no como una contienda de intereses, sino como un intento de conseguir el mejor programa político, definido por los principios de justicia”, (iii) (Sen, 2006) Pág. 229, (iv) (Calabresi, 1991) Pág. 1225, (v) (Ascarelli, 1970) Pág. 303: “No es, en efecto, la simple existencia de la creación intelectual la que justifica el derecho de utilización exclusiva (como acaba por postular, en su formulación corriente, la teoría de la propiedad) o el derecho absoluto a una compensación frente a quien la utilice (como acaba por postular la teoría que se apoya en el trabajo), del mismo modo que no existe un derecho absoluto sobre la clientela o el aviamiento; sólo cuando el derecho absoluto encuentre su última justificación en el interés público podrá ser reconocido”, (vi) (Carraza Torres, 2004) Pág. 163, (vii) (Duffy & Hynes, 2016) Pág. 33.

³⁷⁹ (i) (Menell, Tailoring legal protection for computer software, 1987) Págs. 1337 – 1338, y 1364, (ii) (Ascarelli, 1970) Pág. 277, (iii) Artículo 7 ADPIC: “Objetivos. La protección y la observancia de los derechos de propiedad intelectual deberán contribuir a la promoción de la innovación tecnológica y a la transferencia y difusión de la tecnología, en beneficio recíproco de los productores y de los usuarios de conocimientos tecnológicos y de modo que favorezcan el bienestar social y económico y el equilibrio de derechos y obligaciones”.

³⁸⁰ (i) (Plata López, 2010) Pág. 190: “Es aquí donde se encuentra la real aplicación de los principios del análisis económico, buscando mecanismos eficientes de asignación de los derechos de propiedad exclusiva durante un tiempo limitado, pero que sea suficiente para remunerar el trabajo de los creadores, pero que no sea tan amplio que aumente el costo social y limite el acceso público a los descubrimientos y adelantos científicos y tecnológicos”, (ii) (Sen, 2006) Pág. 228: “Public deliberation can be important and useful both in pursuing institutional reform (even when the totality of the institutional reforms needed for perfection cannot be carried out) and in examining what can be done to reduce injustice – a basic question in the comparative approach to justice- even when not all the right institutions are in place”.

³⁸¹ (Dam, 1995) Pág. 335: “To say that a property right in software is necessary to deal with the appropriability problem is not to say that no economic problems are created by the use of copyright. But these economic concerns can be better assessed once we accept that the appropriability problem must be solved by some form of property rights if the necessary economic incentives are to be created for software innovation”.

Naturalmente deben leerse con agudeza los movimientos actuales de la industria del *software*³⁸², a nivel nacional e internacional³⁸³, los modelos de negocios totalmente diferentes para el momento en que se dieron los icónicos casos estudiados, prestar atención a lo que ya viene en materia de computación³⁸⁴ y admitir un grado de especulación sobre lo que sucedería con un nuevo modelo de protección para el *software*³⁸⁵, a fin de informar a los organismos competentes cómo hacerle seguimiento a los resultados de las normas, y ajustarlas en lo necesario³⁸⁶.

³⁸² (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág 2378: “*The desire for interoperability means that there are natural incentives for many companies to share information (and for some, to monopolize such information) and that consumers have an interest in seeing that information shared. Any legal regime should both take advantage of the existing incentive and facilitate interoperation. The large number of small companies illustrates that the initial creation of a new software product is relatively low in capital expense and affirms the folk wisdom that innovation in this industry is often accomplished by small teams. This, in turn, suggests that any legal regime should be wary of its impact on small companies, with their limited resources*”.

³⁸³ (Sen, 2006) Pág. 234: “*Despite the undoubted importance of local knowledge, global knowledge has some value too, and can contribute to the debates on parochial values and practices*”.

³⁸⁴ (i) Vaticanando 7 cambios fundamentales en la programación de *software* para la siguiente década (Murphy-Hill & Grossman, 2014) Pág. 145: “*In particular, we make the following predictions: 1. Over the next decade, the design of new programming languages will increasingly focus on leveraging supportive IDEs, as well as assuming powerful social networks. 2. Over the next decade, the source-code ecosystem will evolve away from an ASCII-as-ground-truth mindset, treating code as rich, structured data supporting many views. 3. Over the next decade, language designers will increasingly use data to drive the design of new languages and language features. 4. Over the next decade, the use of interactive proof assistants for co-developing robust programs and proofs of correctness will allow developers to prove more powerful properties of real programs, with the proof-engineering difficulties becoming a primary research focus. 5. Over the next decade, the passé argument over static versus dynamic typing will give way to languages supporting a continuum and a gradual-typing methodology that can be adapted to application needs. 6. Over the next decade, language innovations will shift from focusing on batch-oriented or single-user programs to distributed, concurrent, and parallel programming; large workflows of asynchronous computations; accessing massive amounts of rapidly changing data; and other modern computing challenges that will change the boundaries of a well-defined “program” or “code base.” 7. Over the next decade, functional programming will continue to see increased industry adoption, both in terms of developers adopting functional languages (Clojure, Erlang, F#, Haskell, OCaml, Racket, Scala, etc.) and in terms of language designers adopting functional features into other languages. The term “functional language” will continue to lose precise meaning, replaced by a split focus on immutable data and first-class functions*”.

³⁸⁵ (Dam, 1995) Pág. 372.

³⁸⁶ (Bases del plan nacional de desarrollo 2018 - 2022, 2019) Pág. 543: “*c. Objetivo 3: Mejorar la información para la toma de decisiones de política pública de CTeI. 1) Evaluar los impactos de las intervenciones para mejorar las decisiones de política El DNP, con el apoyo del DANE, cruzará la información de los beneficiarios de los programas vigentes con las bases de datos empresariales del DANE (EAM y EDIT), para crear una línea base y posteriormente aplicar evaluaciones de los programas. Esto requerirá realizar convenios de intercambio de información entre las entidades del Gobierno, atendiendo al buen uso y tratamiento de la información*”.

La organización del sistema de protección de forma coherente no es mero propósito a fin de satisfacer vanidades académicas o políticas, sino la respuesta responsable a la indeseable sobreposición de derechos de propiedad intelectual respecto a las funcionalidades del *software*³⁸⁷, que conduce a potenciar la incertidumbre de los participantes en el mercado, al verse avocados a sistemas contradictorios de protección en lo más valioso de este producto, cual es su dialéctica³⁸⁸, sin generar beneficios generales apreciables; lo anterior, porque el Derecho de Autor, el Sistema de Patentes y la Protección a Secretos Empresariales no compensan lo que socialmente se pierde con estos³⁸⁹.

Por supuesto que no se deben descuidar las obligaciones internacionales derivadas de los Tratados ratificados, pero se debe considerar que, mientras se logra el ambiente internacional propicio para *un nuevo modelo de protección para el software*³⁹⁰, proteger internamente a los innovadores contra el *cloning* reportará beneficios domésticos, con vocación internacional³⁹¹,

³⁸⁷ (i) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1340: “*From the perspective of the public interest, the optimal system for protecting intellectual work equates the marginal benefit of enhancing the scope of intellectual work protection with the marginal cost of greater protection. That is, it equates the benefits of the availability of more and better products with research costs, losses due to monopolistic exploitation, administrative costs, and inhibiting effects on inventive activity*”. (ii) (Baylos Corroza, 2009) Págs. 498 – 499

³⁸⁸ (i) “*behavior*” para: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2315 – 2319, (ii) “*idea subyacente*” para: (Calle D'Alleman, 2012) Pág. 263, (iii) “*flow of control (the dynamic-structure)*” para: (Swinson, 1991) Pág. 198, (iv) Indicando que también otras obras protegidas por el Derecho de Autor se “comportan”: (Ginsburg, 1994) Pág. 2567.

³⁸⁹ (Calabresi, 1991) Págs 1221 – 1222: “*Some may argue, however, that I have been using the wrong definition of efficiency. The appropriate definition, they would say, is that given by the Kaldor-Hicks test. This test roughly states that a move is efficient whenever the winners win more than the losers lose, in the sense that, if the winners compensated the losers to their satisfaction, the winners would still be better off than they were before the change. (This test is sometimes called potential Pareto superiority, because it has the potential of leading to a Pareto improvement. Achieving that potential, however, depends on successful compensation of losers, to their satisfaction. But actual compensation is not required by the Kaldor-Hicks test and may not even be feasible)*”.

³⁹⁰ Quizá reformando ADPIC: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2388.

³⁹¹ (i) De acuerdo con el informe (Caracterización del sector de telecomunicaciones, software y TI en Colombia. Año 2015. , 2016), los servicios de programación de *software* representan un 20% del total de la facturación al exterior de la industria (Pág. 32), (ii) La ventaja comparativa aquí indicada es advertida por: (Nimmer & Krauthaus, 1987) Pág. 737, (iii) Sobre el efecto sobre la innovación, derivada del cambio en las instituciones jurídicas en el *civil law* (Considerado un sistema “rígido” de promulgación de normas) y en el *common law* (Considerado un sistema “flexible” de promulgación de normas): (Anderlini, Felli, Immordino, & Riboni, 2013) Pág. 953: “[R]igid legal systems are

gracias a la ventaja comparativa que las firmas establecidas en Colombia aprovecharán por las reglas jurídicas favorables que acá encuentran, y que poco a poco irán permeando el resto del mundo³⁹².

Ya no estamos en ese momento incipiente de la industria³⁹³ ni de la jurisprudencia³⁹⁴ de la época en que se determinó el modelo actual de protección para el *software*, y al margen de la conformidad con las licencias de código abierto con que se haya creado un *software*³⁹⁵, lo cierto es que el *cloning* impide hoy una compensación financiera adecuada tanto para el creador de un *software* propietario, como para el creador - o creadores – de un *software* creado bajo licencia de código abierto. Lo anterior, pues ni uno ni otro son retribuidos por el producto que elabora el clonador, no son compensados por el menor esfuerzo que debe hacer éste al crear su producto – y

preferable (in terms of welfare and rate of output growth) in the early stages of technological development. Flexible legal systems are, instead, preferable at intermediate stages of development: output grows faster and welfare is greater. Finally, when technology is mature, the two legal systems are equivalent. The amount of innovation in the rigid regime may be either inefficiently low or, under some conditions, inefficiently high. The welfare comparison summarized above holds even when we assume that in the rigid legal system the statute (or regulation) can be changed ex post at a cost. In our view, the analysis above sheds light on how technological innovation is shaped by a system of common law, as opposed to a civil”.

³⁹² Este efecto comparativo ocurrió entre Estados Unidos de América y Europa, al comenzar los primeros a conceder patentes de *software*, y lo registra: (i) (Botana Agra, 2001) Págs. 168 – 169, (ii) (The software industry and developing countries, 2012) Pág. 55, (iii) Sobre el principio de ventaja comparativa (Samuelson & Nordhaus, 2006) Pág. 287.

³⁹³ En concordancia: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2369.

³⁹⁴ Esto, en respuesta a: (i) (Menell, The challenges of reforming intellectual property protection for computer software, 1994) Pág. 2651, (ii) (Measures to enhance international cooperation in the field of legal protection of computer software, 1979) Pág. 2. Adicionalmente, nunca se dio el viraje hacia unas decisiones jurisprudenciales concordantes con los que sucede en el mercado, como esperaba: (iii) (Ginsburg, 1994) Pág. 2571 y (iv) (Nimmer & Krauthaus, 1987) Pág. 746.

³⁹⁵ Seguidores de la FSF y de la OSI sí han levantado la voz, por considerar en múltiples casos que se han irrespetado los términos de la licencia GPL, en particular cuando los creadores de *software* previamente licenciado bajo versiones de la licencia GPL no permiten el acceso a los código fuente, ora mediante medidas de seguridad para el acceso, ora mediante su ocultamiento. (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 302: “*Many within the free software community now believe that software patents are the greatest threat to open source and free software. A report by open source Risk Management found that 238 granted patent could potentially be used against the Linux kernel. However none of these patents have been tested in court*”. Algunos artículos de prensa sobre este particular pueden consultarse en: <http://gpl-violations.org/>

por el cual es remunerado como si fuera innovador -; lo cual no debe tutelarse³⁹⁶, como no deben concederse más las patentes de *software* que surgieron en reacción al *cloning*.

Atender el déficit de protección denunciado es un llamado a corregir el rumbo y conducirnos hacia “una nueva tierra”³⁹⁷, es dar el paso adelante que el mundo ha demorado³⁹⁸, pues el propósito legislativo y jurisdiccional debe ser el de asignar apropiadamente los derechos³⁹⁹, a fin disminuir la incertidumbre derivada de la aplicación inercial de regímenes inapropiados, con lo cual los beneficios de la propiedad intelectual serán mejor apreciados en esta industria y en la economía en general.

En esta empresa, consideramos que la política pública del país debe “por fin” ponerse “manos a la obra”, actuando en la dirección que tantos documentos nacionales⁴⁰⁰ e internacionales indican, tantos estudios económicos ponen de presente y por sobre todo, la política pública nacional debe tomar conciencia del pendular retroceso jurídico que se evidencia hoy respecto del otorgamiento de patentes de *software*, y que se lee en tono angustiante desde el

³⁹⁶ (Carraza Torres, 2004) Pág. 164: “[D]e ninguna manera hay igualdad entre aquel que pone esfuerzo, tiempo, creatividad, etc., en la creación o desarrollo de un *software* y aquel que no pone nada”. Vale indicar también, con el mismo autor y obra (Pág. 171), que no se afecta la justicia si el autor renuncia a sus derechos vía adherencia al movimiento de código abierto.

³⁹⁷ (i) (Menell, The challenges of reforming intellectual property protection for computer software, 1994) Pág. 2654, (ii) (Sen, 2006) Pág 237: “A procedural requirement to consider nonlocal perspectives can help to avoid undue dominance of local interest as well as possible parochialism of local reasoning shaped by the influence of established conventions an limited informational frameworks”.

³⁹⁸ (Sen, 2006) Pág. 235 “The comparative approach does not requires ‘all or nothing’ extremism, and it allows the world to come to grips with intense issues of global injustice (such a famines, widespread hunger, rampant illiteracy, or needless deaths from preventable or manageable diseases), on which consensus may be easier to obtain, without waiting for a full agreement on more contentious evaluations. **A similar thing can be said about taking note of global perspectives in making local decisions that are not primarily matters of global justice**” (Negrilla y subraya fuera del texto original).

³⁹⁹ Considerando que es mejor dejar esta tarea al judicial, y no al legislativo: (Dam, 1995) Págs. 372 – 373.

⁴⁰⁰ ¿No es una pena lo que hemos resaltado en notas anteriores respecto de los siguientes estudios: (i) (Conpes 3582, 2009), (ii) (Heshusius Rodríguez, 2009), (iii) (Conpes 3678, 2010), (iv) (Conpes 3834, 2015), (v) (Informe Nacional de Competitividad 2018 - 2019, 2018)?

Extranjero⁴⁰¹; pues tal evolución jurídica, empieza a discontinuar el único remedio contra el *cloning*.

En vez de estar a la “penúltima moda”, concediendo ahora patentes de *software*, consideramos que los siguientes contornos generales responden a la misión de re-balancear los fundamentos políticos, económicos y jurídicos que soportan el reconocimiento de una propiedad intelectual para el *software*, por las razones que se expresan a continuación:

2.1. Protejamos los algoritmos que componen el *software*, para usos reivindicados y segundos usos innovadores.

La definición de un sistema de protección *sui generis* está condicionado a la existencia de un repositorio especial⁴⁰² de bienes de esta naturaleza, organizado mediante un sistema de registro constitutivo de derechos⁴⁰³. Un sistema de esta naturaleza evita el detrimento social que

⁴⁰¹ (i) (Gugliuzza & Lemley, 2018) Págs. 776 y 777: “*The current state of patentable subject matter doctrine has elicited vociferous complaints from lawyers, scholars, and even some judges. They criticize Supreme Court and Federal Circuit case law as confusing and unpredictable. They worry that the restriction of patent eligibility threatens innovation, particularly in the fields of biotechnology and medical diagnostics. And they lament that the patentable subject matter requirement serves no policy objective... Those concerns have led the major bar associations representing patent lawyers to call for legislation overturning the Supreme Court’s patentable subject matter case law*”, (ii) (Alice Corp. v. CLS Bank International, 2014) Págs. 28 – 30.

⁴⁰² (Model provisions on the protection of computer software, 1978): Pár. 15: “[C]ountries considering the question discussed above should first decide the basic approach to the system of protection to be established. If a patent law approach were adopted, it would be logical for a requirement for compliance with formalities to be included in legislation based on the model provisions, which, as a whole, would have to be examined in the light of such an approach”.

⁴⁰³ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2426: “*Although anti-cloning legislation will take care of the industry’s most pressing protection need, it is only a partial solution. A broader solution would include adoption of a registration system for innovative compilations of applied know-how embodied in software. This system would facilitate development of a repository of state-of-the-art software that would be of assistance in the evolution of software engineering, as well as potentially provide an exchange through which low-cost transactions about reuse of software innovations could occur. A registration system of this sort might also provide incentives for software developers to disclose high level descriptions of innovative algorithms and other internal design elements of software*

comporta la inversión de recursos en actividades legales e ilegales de descompilación y/o ingeniería reversa⁴⁰⁴, búsqueda de antecedentes⁴⁰⁵, espionaje, desorganización empresarial, o el socialmente costoso fenómeno de conservar secreta información.

Ahora, para no conceder derechos a lo que ya se venía explotando en el mercado⁴⁰⁶, y a fin de respetar los derechos de patentes de *software* que se hubieran concedido, deben recopilarse algoritmos de las librerías especializadas, del propio sistema de patente en lo que fuere conocido, y con la ayuda de la tecnología del momento, hoy *blockchain*⁴⁰⁷, contar con una bitácora inicial de códigos fuentes, clasificados según las diferentes combinaciones de bases de datos y lenguajes de programación actuales⁴⁰⁸; todo esto, antes de poner en servicio el registro público de algoritmos para *software*, para dar tiempo a que los funcionarios correspondientes se entrenen

that today are kept secret. Registration would provide a way for innovators to get compensation for disclosed innovations without undergoing the lengthy, expensive, and cumbersome process of patenting".

⁴⁰⁴ (i) (Sega Enterprises Ltd. v. Accolade, 1992), (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2387

⁴⁰⁵ (i) (Dam, 1995) Pág. 369 – 370, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2410, (iii) Así podría evitarse la “quiebra” de las pequeñas y medianas compañías informáticas, por el “hercúleo” trabajo de estudiar el desempeño del algoritmo y evitar infracciones: (Uribe Arbeláez, 2005) Págs 215 – 216.

⁴⁰⁶ A este respecto se ha resaltado la conveniencia del criterio de “originalidad” asociado al “estado del arte previo”, que encontramos en el artículo 3.2. del Tratado sobre la propiedad industrial respecto de los circuitos integrados (1989): (i) (Gómez Segade, 2001) Pág. 860, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2400, (iii) Líneas que consideramos aplicables a este punto, pero referidas a los efectos sobre las nuevas obras si se crea una protección de derechos retroactiva: (Landes & Posner, 1989) Pág. 362.

⁴⁰⁷ (i) Las propiedades inmutables, sucesivas y de comprobación simultánea en las operaciones que se computan mediante plataformas *blockchain* es lo que puede aprovecharse a fin de rastrear la explotación de ciertos algoritmos implementados por programas de ordenador, que presenten específicas funcionalidades. (ii) (Bases del plan nacional de desarrollo 2018 - 2022, 2019) Pág. 655: “Las entidades públicas, de todos los sectores, incorporarán en sus proyectos de inversión soluciones basadas en el uso de tecnologías emergentes y disruptivas, como los registros distribuidos (por ejemplo, *blockchain*), analítica de datos, inteligencia artificial, robótica e Internet de las cosas, que tengan potencial para transformar la vida de los ciudadanos”.

⁴⁰⁸ Mientras el algoritmo esté destinado a servir de fundamento a un *software*, la “idea en sí” de un modelo de negocio no estaría protegida, solamente su aplicación como *software*, en la configuración que sea descrita en las reivindicaciones, y según el tiempo dual de protección exclusiva y de remuneración. Así, nos pondríamos a tono con lo que actualmente sucede con las patentes de *software* en Estados Unidos de América, que se acercan cada vez a los modelos de negocios. Sobre esta evolución de las patentes de *software* en Estados Unidos de América, y el llamado para que Europa no se queda atrás: (i) (Botana Agra, 2001) Págs. 174 – 177, (ii) (Salas Pasuy, 2016) Pág. 459.

previo a la concesión de derechos de exclusiva⁴⁰⁹ y asegurar que sean concedidos derechos a algoritmos concretamente aplicados⁴¹⁰, y no a ideas abstractas⁴¹¹ o eufemísticas “*invenciones implementadas por programas de ordenador*”⁴¹².

No debe perderse de vista que la característica cumulativa de la industria sugiere una re-explotación del algoritmo para usos desconocidos inicialmente⁴¹³, bien sea por la velocidad con la cual cambian los modelos de negocios, paradigmas de codificación⁴¹⁴, bases de datos y lenguajes de programación; ora porque las aplicaciones para estos algoritmos solo se presentan tiempo después de estudiadas las propiedades de éstos⁴¹⁵, ora porque se descubrirán tiempo después de haber sido escritas por las máquinas⁴¹⁶. Por esto, el registro constitutivo debe basarse

⁴⁰⁹ (Menell, The challenges of reforming intellectual property protection for computer software, 1994) Pág. 2650: “*Patent or patent-type legislation (with shortened duration and improved examination) should do well in handling new techniques for improving the operation of software and hardware*”.

⁴¹⁰ (i) (Calle D'Alleman, 2012) Pág. 262, (ii) Debe evitarse la omni-comprensión de aplicaciones con base en un mismo algoritmo, debido a que los lenguajes de programación serán cada vez más abstractos: (Swinson, 1991) Pág. 174, (iii) Resaltando lo controversial que puede ser una protección a los algoritmos de un *software*: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2379 – 2380.

⁴¹¹ (i) (Swinson, 1991) Pág. 154 – 155, (ii) (Ginsburg, 1994) Pág. 2570, (iii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2395, (iv) (Gugliuzza & Lemley, 2018) Pág. 775.

⁴¹² (Dam, 1995) Pág. 366.

⁴¹³ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2361 – 2362.

⁴¹⁴ Una combinación diferente de lenguaje de programación y base de datos, aplicando un mismo algoritmo, puede traer consigo una mejora en eficiencia, no prevista anteriormente. En este sentido: (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 48.

⁴¹⁵ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2361 – 2361.

⁴¹⁶ (Dam, 1995) Pág. 324: “*Protecting software as a literary work seemed particularly galling to some critics since in some forms software can not be read by humans, since it is intended for ‘reading’ by machines, and since it may someday actually be written by machines*”.

en la descripción de reivindicaciones⁴¹⁷, que luego pueden derivar en segundos usos protegibles⁴¹⁸, sin necesidad de registrar el *software* por completo⁴¹⁹.

Mediante este sistema de registro, el secretismo alrededor del algoritmo se transforma en una forma menos costosa de administrarlo⁴²⁰, estudiarlo⁴²¹ y explotarlo⁴²², tanto para su titular como para los demás involucrados en la industria, lo que redundará en beneficio de una disciplina en el mercado de mayor innovación⁴²³ y menos desperdicio de recursos⁴²⁴.

⁴¹⁷ Nos imaginamos una descripción de la reivindicación de algoritmos, a partir de la explicada compaginación (“*indentation*”) en: (Paterson, 2013) Págs. 176 – 177.

⁴¹⁸ Esto evitaría el efecto retrospectivo que se presenta en las patentes, al ver todo obvio en el momento actual, sin comprender bien que el adelanto hecho por el innovador, la compañía sucesora (“*follow-on*”) o competidora (“*me too*”) es vital para más adelantos futuros, en una industria que re-usa constantemente, y para mejores productos, algoritmos conocidos. En este sentido: (i) (Cohen & Lemley, Patent scope and innovation in the software industry, 2001) Pág. 46, (ii) (Swinson, 1991) Pág. 159 – 160, (iii) (Lemley, Menell, Merges, Samuelson, & Carver, 2011) Pág. 169.

⁴¹⁹ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2418.

⁴²⁰ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2373.

⁴²¹ Permitir una “*explotación pasiva*” del secreto, a efectos de no repetir errores y con base en una generosa descripción de las reivindicaciones, es el fundamento explícito para que sean concedidos derechos al *software* a partir de un registro constitutivo. Al respecto: (i) (Galán Corona, 2011) Pág. 365, (ii) (Dam, 1995) Pág. 330, (iii) “*clean room*”: (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1651.

⁴²² Lo cual consideramos un incentivo para que las firmas de la industria elijan este modelo frente a los disponibles actualmente. En posición contraria: (Ginsburg, 1994) Pág. 2565.

⁴²³ (i) Señalando que el postulado clásico a favor de la propiedad intelectual, como respuesta a la externalidad de bienes públicos, debe entenderse hoy como el vehículo para comunicar y compartir eficientemente la información: (Maggiolino, 2011) Pág. 88, (ii) Advirtiendo que los consumidores se verán afectados por la menor propensión al lanzamiento de productos en el mercado: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2382 – 2383: “*Consumers might find this lower price attractive in the short run. If the situation is not corrected by the law or otherwise, consumers suffer in the long run when software products are under-produced because optimal investments for innovation cannot be economically justified*”, (iii) (Bessen & Hunt, The software patent experiment, 2004) Págs. 15 – 16: “*In contrast to the incentive theory already described, suppose instead that firms in an industry firms assemble large patent portfolios in order to extract royalties from competitors and to defend themselves from similar behavior by their rivals. Economists have come to describe such an environment as a patent thicket. In theory at least, extensive competition in patents, rather than inventions, may occur if firms rely on similar technologies and the cost of assembling large portfolios is not very high. In such an environment, firms may compete to tax each others’ inventions and in the process reduce their competitors’ incentive to engage in R&D. The outcome of patent litigation and licensing agreements often depends on the size of the firm’s patent portfolio. This creates an incentive to build larger patent portfolios, especially when the firm focuses on patents as a competitive strategy. In this account, firms choose to compete in court, rather than in the marketplace*”.

⁴²⁴ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2409, (ii) (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1625: “*Minimizing wasted costs is the fourth social welfare criterion. It too yields somewhat mixed policy prescriptions. Duplicated or wasted costs may arise in the software industry from at least three activities: the act of reverse engineering itself (costs wasted by the reverse engineer); the process of devising ways (e.g., technical protection measures) to make interfaces difficult or impossible to reverse-engineer (costs*

Lo primero, pues la determinación del derecho según las reivindicaciones descritas no estará expuesto a las discusiones sobre la funcionalidad del *software* debatidas en los casos de Derechos de Autor analizados⁴²⁵, o en los supuestos y criterios de patentabilidad cuestionados en las patentes de *software* concedidas. Lo segundo porque los titulares de derecho no podrán procrastinar su propia innovación, y las empresas sucesoras (“*follow-on*”) o competidoras (“*me too*”) podrán estudiar el producto innovador realmente advertidas de las consecuencias infractoras de intentar lanzar al mercado un producto sustituto y/o complementario⁴²⁶.

No dudamos que otros elementos del *software* pueden alcanzar grande valor⁴²⁷, sino lo son ya muy valiosos, pero consideramos que amparar tales aspectos con derechos exclusivos o

wasted by the platform developer); and the development of different applications for different interfaces rather than the same applications for all interfaces (costs wasted by application developers generally). A prohibition on reverse engineering would avoid the first two but may well encourage the third. A platform provider can, of course, avoid the first cost by licensing, and as in other industrial contexts, a legal rule in favor of reverse engineering may provide powerful incentives for firms to license to avoid having their products reverse-engineered”.

⁴²⁵ (i) (Bouza López, 1997) Pág. 344, (ii) (Menell, The challenges of reforming intellectual property protection for computer software, 1994) Págs. 2652 – 2653, (iii) Consideramos igualmente que se resuelven todos los problemas asociados al *cloning*, enumerados en: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2381.

⁴²⁶ Tengamos muy presente que las firmas (“*me too*”), en términos generales, no procuran innovar, y las firmas sucesoras (“*follow-on*”) reposan muy inercialmente en el adelanto del innovador; pero ambas tienen la aptitud para convertirse en innovadoras. Luego, deben ser incentivadas para eso.

⁴²⁷ Criticando propuestas para la protección de algoritmos, en consideración a que no es lo único que debería protegerse en el *software*: (Leith, 2007) Pág. 172: “*The assumption is that algorithms are what require protection. This appears to be a lawyer’s view of computer science when – as we have outlined earlier – software is really a virtual machine where algorithms are important but not the only element worth protecting”.*

de remuneración enquistaría las externalidades de *redes* y la *estandarización de facto*⁴²⁸ que, aunque no deben descuidarse⁴²⁹, distraerían realmente la solución al *cloning*⁴³⁰.

2.2. Concedamos un derecho de exclusiva, que se transforme en un derecho de remuneración al cabo de un tiempo y esté sujeto a un agotamiento nacional del derecho.

Si sabemos que un derecho de exclusiva por veinte años adormece la innovación⁴³¹, ni pensar en uno superior o indefinido⁴³², como en el Derecho de Autor y la Protección a Secretos Empresariales, respectivamente. No desconocemos que la definición de un término de protección siempre será una decisión política y económicamente controversial, pero jurídicamente debe especificarse en un número de años exacto⁴³³, que sugerimos fijarlo en máximo diez años, sin

⁴²⁸ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2385, (ii) (Drexel, 2011) Págs. 170 – 171: “[C]ompetitors need to know the interface specifications only for designing the interfaces of their own programs. Therefore, in the process of creating a compatible program, ... specification should not be ‘...reproduce, adapted, arranged or altered’; competitors should only be enabled to write their own specification-compliant interfaces”, (iii) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1345: “In designing legal protection for intellectual work with network externalities, policy makers should foster the realization of benefits from standardization-wider availability of products, lower search costs, vigorous competition-while minimizing the potentially adverse effects of universal access to new technologies on incentives to innovate better product standards”, (iv) (Samuelson & Scotchmer, The law and economics of reverse engineering, 2002) Págs. 1620.

⁴²⁹ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2398 – 2399.

⁴³⁰ (Calabresi, 1991) Pág. 1236: “Taking seriously Coase’s insight that everything, including existing markets and potential markets, has costs, means that we can concentrate on which kinds of costs we should realistically try to reduce and which we should not, for the moment, bother with”.

⁴³¹ (i) (Dukrook & Hwang, 2010) Pág. 382, (ii) (Dam, 1995) Pág. 369, (iii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2412: “They may be reluctant to accept a substantially lesser duration of protection than existing regimes provide, even if this is all that is necessary to repair market failure and promote consumer welfare. By emphasizing consumer welfare as a principle and goal of a market-oriented legal regime, however, it should be possible to overcome the temptation to overprotect”.

⁴³² (i) En aislamiento, proponiendo un término de protección más largo para la protección utilitaria de la codificación del *software*, después de afirmar que han habido propuestas de protección por tres (3) años: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2382 y 2423, (ii) Considerando que la obsolescencia del *software* se da a los dos años: (Gómez Segade, 2001) Pág. 897.

⁴³³ (i) (Ascarelli, 1970) Pág. 278, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2397: “Near-clones should be regulated for a market-preserving period of time just as clones would be”.

posibilidad de renovación, pero con la característica de que solo los primeros cinco conceden un derecho de exclusiva para el titular⁴³⁴.

Es decir, los primeros cinco años permitirán al titular impedir la explotación del algoritmo, premiándole con una ventaja de tiempo en la cual podrá recuperar sus costos directos, indirectos y de oportunidad, proveyendo sólo las prestaciones asociadas al *software*⁴³⁵ o si lo prefiere, construyendo su red de proveedores y usuarios, mediante el pago de regalías o como su modelo de negocio le recompense⁴³⁶; pero al cabo de los primeros cinco años el derecho concedido al titular solo será de remuneración⁴³⁷, por cinco años más⁴³⁸, lo cual le imposibilitará rehusarse exitosamente a contratar con firmas de la industria las licencias que le reclamen⁴³⁹,

⁴³⁴ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2414.

⁴³⁵ Como la firma innovadora está recuperando sus costos por la creación del *software* innovador, no es previsible que lo haga con precios predatorios como lo advierte, genéricamente: (Posner, Antitrust law, 2001) Págs. 255 – 256.

⁴³⁶ (i) Considerando que un modelo de protección para el *software* que no conceda derechos exclusivos es insulso para la industria: (Leith, 2007) Pág. 172, (ii) Recomendando la adopción de un nuevo sistema de protección para el *software*, que le permita a los innovadores disfrutar una ventana de tiempo para atender la demanda del producto: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2364.

⁴³⁷ (Baylos Corroza, 2009) Págs. 805 – 806: “[J]unto al «derecho de explotación del autor» (con sus poderes de autorizaren exclusiva la reproducción, transformación, distribución y comunicación pública de su obra), la categoría de los «derechos de remuneración equitativa». Entendemos que esa denominación -aceptada casi unánimemente por los legisladores y la doctrina- es la más apropiada para designar cualquier derecho que se reconozca al autor y que, en lugar de concederle el poder de autorizar o prohibir una determinada explotación de su obra, sólo le confiera el de percibir (del explotador o de un tercero) una suma de dinero, cuyo importe, cualquiera que sea la causa de la atribución de ese derecho, no sea fijado por la voluntad de ese titular”:

⁴³⁸ (i) Explicando que identificado el interés político detrás de la concesión de propiedad intelectual a una creación, el Derecho puede premiar y/o compensar al creador: (Ascarelli, 1970) Pág. 281, (ii) Como trabajo de campo se hizo una encuesta pública entre programadores y firmas de la industria del *software*, en la cual se les preguntó si estarían dispuestos a registrar el algoritmo del *software* que crearan y a aumentar sus presupuestos en investigación y desarrollo, si a cambio tuvieran un período de exclusividad en la explotación del algoritmo. El 86% de los encuestados respondió que sí. Preguntados igualmente sobre el período que consideraban apropiado para recuperar los costos por la creación del *software*, también el 86% de los encuestados respondieron que cinco (5) años serían suficientes. Lo anterior, concuerda con: (iii) (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1653: “Our advice to policymakers is this: Before banning a means of reverse engineering, require convincing evidence that this means has market destructive consequences. Realize that existing market participants may want a ban mainly because they wish to protect themselves against competitive entry. Any restriction on reverse engineering should be tailored so that it does not reach more than parasitic activities”; y con, (iv) (Landes & Posner, 1989) Pág. 336: “Optimal copyright protection depends not only on the number and the cost of works but also on the number and the cost of copies of each work”, (v) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2409.

⁴³⁹ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2420: “We also considered an approach that would differentiate between those who made improvements to an innovative program compilation derived from another

aunque se deja a salvo la posibilidad para que el titular facture las regalías que usualmente cobraría⁴⁴⁰.

Podría cuestionarse que un sistema *sui generis* de protección⁴⁴¹ diseñado para conceder al titular un tiempo corto para recuperar su inversión en (I+D+i) no se justifica, si el mercado concede al innovar esa misma ventana de tiempo en forma más eficiente; pero ciertamente tal ventana de tiempo no es clara⁴⁴², y en la práctica insuficiente, por lo que reservar normativamente un tiempo a los innovadores para que decidan si atienden solos la demanda es una medida apetitosa para las firmas de la industria⁴⁴³.

firm's product and those who imitated the compilation without making improvements. One might, for example, let an improver come to market sooner than one who copied an equal quantum but made no improvements. Alternatively, one could block the copier while allowing the improver to license for a standardized fee"

⁴⁴⁰ No congeniamos con la excepción de interoperabilidad y/o compatibilidad, tampoco con un listado ampliado de excepciones según el propósito del beneficiado, motivo por el cual no proponemos un derecho de exclusiva excepcionado o limitado, sujeto a interpretaciones, como es la preocupación que despierta esa permisión de ingeniería reversa. Sobre auscultar el propósito de la ingeniería reversa, para patrocinar una y repudiar otras: (i) (Gómez Segade, 2001) Pág. 887 – 889, (ii) (Dam, 1995) Pág. 349, (iii) (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Págs. 1608, 1650 – 1651 y 1657, (iv) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2390 y 2404.

⁴⁴¹ (López, 2009) Pág. 15.

⁴⁴² (i) (Dam, 1995) Pag. 334 y 356, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2364, (iii) (Samuelson & Scotchmer, The law an economics of reverse engineering, 2002) Pág. 1649: "*The right balance between facilitating improvements and protecting earlier innovators can be achieved by granting a kind of 'leading breadth' to each innovation, but not by prohibiting researchers from access to knowledge, as the DMCA does*", (iv) (Posner, Antitrust law, 2001) Pág. 254.

⁴⁴³ (i) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2371, 2422 – 2423, 2392, 2413, (ii) (López, 2009) Págs. 13 – 14, (iii) (Leith, 2007) Pág. 160, (iv) (López, 2009) Págs. 9 y 13. Consideramos también que el tiempo de protección así planteado no genera los efectos denunciados por quienes temen que refinar la protección para el *software* con un fuerte derecho de propiedad intelectual, podría empeorar las externalidades negativas que se presentan en la industria; tales como: (v) (Menell, The challenges of reforming intellectual property protection for computer software, 1994) Págs. 2646 y 2648, (vi) (Maggiolino, 2011) Págs. 82 – 83, (vii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2375 y 2390 (viii) Sobre los efectos generales de una protección débil o fuerte a obras protegidas por el Derecho de Autor, en función de los costos para su expresión y reproducción: (Landes & Posner, 1989) Pág. 335: "*Some copyright protection is necessary to generate the incentives to incur the costs of creating easily copied works, but too much protection can raise the costs of creation for subsequent authors to the point where those authors cannot cover them even though they have complete copyright protection for their own originality*".

Lo anterior tiene directa relación con los presupuestos de (I+D+i) que, para no estancarlos por el *cloning*⁴⁴⁴, mal dirigirlos por la excepción de interoperabilidad o que las compras estatales estimulen la repetición de productos, es recomendable dejar a las firmas de la industria asignar y orientar recursos para la (I+D+i) apremiadas por una competencia paulatina⁴⁴⁵, y que tales presupuestos se vean nutridos y no interferidos por el confort proporcionado por un cliente tan importante como el Estado⁴⁴⁶, determinando unos plazos perentorios y de fácil contabilización⁴⁴⁷.

Vale la pena señalar que una política estatal de incentivos tributarios, como la que actualmente se ha puesto en marcha en Colombia⁴⁴⁸, no se aprovecha igual sin estar acompañada

⁴⁴⁴ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2388: “*If the general rule that unpatented technical innovations embodied in mass-marketed products could be freely appropriated were to be applied to program code, it would result in market-destructive appropriations of program innovations and underinvestment in the development of commercially valuable programs. In such a legal environment, the developer of an innovative software product would have to recoup all of its research and development expenses and make a profit on the first (and likely only) sale of the product, a scenario unlikely to lead to a high degree of investment in software*”.

⁴⁴⁵ (i) (Samuelson & Scotchmer, The law and economics of reverse engineering, 2002) Pág. 1394 y 1650, (ii) (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs., 2380 – 2381, 2389.

⁴⁴⁶ (i) (Junqueira Botelho, Stefanuto, & Veloso, 2007) Pág. 100, (ii) (Adler, 1986) Págs. 18, 19 y 33, (iii) Resaltando el peso del sector público en la inversión en (I+D+i) y el bajo presupuesto privado en estas actividades: (Conpes 3834, 2015) Pág. 16, (iv) (Conpes 3582, 2009) Pág. 31: “*El reconocimiento del potencial que tienen las compras públicas como instrumento para promover la innovación hace preciso establecer la viabilidad de un instrumento que haga parte del portafolio de incentivos y que permita cambiar el enfoque de adquirir bienes y servicios detalladamente definidos por uno que haga uso de las convocatorias para encontrar soluciones innovadoras a problemas específicos que, posteriormente, puedan ser objeto de adquisición mediante cualquier modalidad de contratación*”. (v) (Bases del plan nacional de desarrollo 2018 - 2022, 2019) Pág. 536: “*Para aumentar el volumen total de inversión y alcanzar la meta del 1,5 % del PIB es necesario generar más incentivos para la atracción de recursos privados. Aunque la inversión privada en CTeI ha venido creciendo en los últimos años, se pasó de un crecimiento en 2015 del 8,5 % a uno del 2,1 % en 2017*”.

⁴⁴⁷ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2411

⁴⁴⁸ Nos referimos a: (i) Las exenciones, deducciones y descuentos tributarios por (I+D+i), contenidas en los artículos 57-2, 158-1, 256 y 256-1 del Estatuto Tributario (Con las modificaciones introducidas por los artículos 168, 170 y 171 de la Ley 1955/19), y (ii) La tarifa reducida del 3,5% por concepto de retención en la fuente, por el pago o abono en cuenta, por concepto de derecho de uso y/o elaboración de *software*, según el párrafo 1° del artículo 1.2.4.3.1 del Decreto 1625 de 2016. (iii) No se pudo hacer una medición sobre el efecto del derogado numeral 8 del artículo 207-2 (Fedesarrollo, 2005) Pág. 82. (iv) Considerándolo una financiación indirecta a la producción nacional de propiedad intelectual (Buitrago Díaz, 2012) Pág. 472: “*En Colombia, la normativa tributaria no incentiva la explotación de DPI [Derechos de Propiedad Intelectual] de no residentes, que está sujeta al pago de varios impuestos, entre ellos el impuesto sobre la renta*”. (v) Pidiendo a la OCDE no satanizar incentivos a la creación local de productos de propiedad intelectual: (Buitrago, Patent boxes and the erosion of trust in trade and in governance, 2019) Pág. 30.

de una mejor protección de la creación intelectual⁴⁴⁹ que, a no dudarlo, será más orientada hacia la exportación de servicios⁴⁵⁰. Esto último, como se expresó anteriormente, por cuanto una protección *sui generis* para el *software* no se ha ensayado previamente en otras latitudes – apenas se han perfeccionado las leyes de Derecho de Autor existentes, y por esto es que con propiedad es mejor llamar a esas novedades legislativas una *lex specialis*⁴⁵¹ -, con lo cual se crearía en Colombia una ventaja comparativa que fortalece el mercado interno, para actuar en el externo⁴⁵².

⁴⁴⁹ (Conpes 3834, 2015) Pág. 9. El mismo documento señala, como una de las dificultades para el aprovechamiento de las deducciones, deriva de la inseguridad jurídica para compartir información sin arriesgar la explotación de la propiedad intelectual forjada: Pág. 23: “Según su experiencia, las diferencias en el lenguaje, **las dificultades para alcanzar acuerdos en términos de propiedad intelectual y la articulación de las necesidades empresariales con los intereses académicos de los grupos de investigación**, demandan esfuerzos de medio y largo plazo, que no se ajustan a la ejecución de los proyectos. En términos generales, no existen todavía vínculos consolidados entre las partes de acuerdo con la percepción de empresarios, como lo refleja el número de proyectos presentados” (Negrilla y subraya fuera del texto original).

⁴⁵⁰ (The software industry and developing countries, 2012) Pág. 38: “Domestic use of software can be instrumental in improving the competitiveness of enterprises and the welfare of society. The social marginal benefit of a dollar worth of ICT (including software) consumed locally is likely to be higher than that of a dollar worth of ICT (including software) that is exported (Kumar and Joseph, 2005). The domestic market is also potentially an important base to develop relevant skills and innovative products. Focusing too heavily on exports implies a danger of transforming the software production into an enclave industry with little exchange with other domestic sectors”.

⁴⁵¹ (i) (Gómez Segade, 2001) Págs. 848 – 849, (ii) (Nimmer & Krauthaus, 1987) Pág. 746, (iii) (Measures to enhance international cooperation in the field of legal protection of computer software, 1979) Pág. 7. Vale decir que ninguna de las leyes especiales para la protección del *software* aplaudidas, transcribieron las reglas especiales contra el *cloning* que venían en el borrador de un Tratado Internacional para la Protección del *Software*, preparado a instancias de la OMPI en 1979, y que parafrasean el modelo de protección propuesto también por expertos ante la OMPI, en 1978. Nos referimos, en particular a la sección 5 del modelo de protección propuesto en 1978. La lectura de los reportes de 1979 y 1978 nos ha inspirado mucho a lanzar las propuestas presentadas en este trabajo. Para acceder a estos documentos: http://www.wipo.int/edocs/pubdocs/en/copyright/120/wipo_pub_120_1978_01.pdf; http://www.wipo.int/mdocsarchives/LPCS_I_79/LPCS_I_2_E.pdf.

⁴⁵² (i) (Junqueira Botelho, Stefanuto, & Veloso, 2007) Págs. 103 – 104, (ii) (Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia, 2014) Pág. 72: “[L]a interacción entre los productores nacionales de Software y Servicios Relacionados y las empresas de los diversos sectores económicos del país, mediante el desarrollo de productos y servicios atendiendo necesidades específicas, muchas veces “a la medida”, genera incrementos en términos de productividad y eficiencia operativa. Por tanto, en la medida que el sector se consolide en el mercado global, sus efectos serán más notorios sobre la economía nacional”, (iii) (The software industry and developing countries, 2012) Pág. 38: “The demand for software and IT services (particularly outsourcing/offshoring) is growing at the global level and many developing and transition economies offer attractive labour costs, relevant human resources and improved connectivity. Export markets may include the major developed countries (North America, Europe and Japan) as well as regional markets. By developing software for international clients, domestic companies can become exposed to the latest technologies and management methods. International clients may place different demands, forcing suppliers to meet international standards and to innovate”.

El que no se haya ensayado previamente en el Extranjero una protección *sui generis* de protección para el *software*, exige una declaración explícita de no contrariar los Tratados internacionales circundantes⁴⁵³, una respuesta a los más delicados cuestionamientos sobre si se debe reclamar reciprocidad⁴⁵⁴, para que las firmas extranjeras gocen de protección en Colombia, y respecto a cuál sería el alcance del agotamiento del derecho⁴⁵⁵, si se consagra alguno⁴⁵⁶, para

⁴⁵³ (i) (Ginsburg, 1994) Pág. 2564, (ii) (Nimmer & Krauthaus, 1987) Pág. 753, (iii) Anotando la imposibilidad de acoger normativamente los postulados de los movimientos de código abierto, por contrariar Tratados internacionales: (Paterson, 2013) Pág. 192, (iv) No creemos que una protección *sui generis* para el *software* provoque necesariamente la re-negociación de ADPIC, o los acuerdos de licencia privados que se hubieren celebrado antes – aunque esto sería ideal -, tal como lo señala: (Dam, 1995) Pág. 376.

⁴⁵⁴ (i) Artículo 11 de la Ley 23 de 1982: “*De acuerdo con los artículos 61 y 71 de la Constitución Política de Colombia, será protegida la propiedad literaria y artística como propiedad transferible, por el tiempo de la vida del autor y ochenta años más, mediante las formalidades que prescriba la ley. Esta ley protege las obras, interpretaciones, ejecuciones, fonogramas y emisiones de organismos de radiodifusión de los colombianos y extranjeros domiciliados en Colombia o publicadas por primera vez en el país. Los extranjeros no domiciliados en Colombia gozarán de la protección de esta ley de conformidad con los tratados internacionales a los cuales Colombia está adherida o cuando las leyes nacionales del otro país impliquen reciprocidad efectiva en la protección de los derechos consagrados a los autores, intérpretes, ejecutantes, productores de fonogramas y organismos de radiodifusión colombianos en dichos países. Parágrafo. Cuando la protección de un fonograma o una interpretación o ejecución fijada en un fonograma se otorgue en virtud del criterio de primera publicación o fijación, se considerará que dicha interpretación, ejecución o fonograma es publicada por primera vez en Colombia, cuando la publicación se realice dentro de los 30 días siguientes a la publicación inicial en otro país*”, (ii) Artículo 6 Convenio de Berna.

⁴⁵⁵ (i) Acerca del concepto de agotamiento del derecho en bienes de propiedad intelectual, (Conrteras-Jaramillo, 2015) Pág. 300: “*La figura del agotamiento ha sido consagrada normativamente como un límite intrínseco al derecho de propiedad intelectual como una consecuencia de la explicación propietaria que justifica su existencia. En este sentido, el agotamiento es reconocido como la frontera hasta donde llega el derecho exclusivo de propiedad intelectual, bien sea porque ‘naturalmente’ el derecho subjetivo es así, o porque el titular del derecho no merece más protección de la que ya se le ha otorgado en relación con su esfuerzo. De otra parte, la figura del agotamiento también ha sido utilizada como un instrumento para la regulación de mercados, tanto en su faceta de política pública de organización de mercados (objetivo fijado ex ante), como en el análisis de los efectos anticompetitivos derivados de su aplicación (ponderación ex post); situación que por supuesto es resultado de una explicación de mercados sobre la figura del agotamiento de los DPI. Como se ha descrito en el apartado anterior del presente artículo, estas dos explicaciones no se excluyen mutuamente, al punto que hoy se encuentran explicaciones conjuntas sobre del agotamiento*”, (ii) Sobre el alcance espacial que puede tener un “agotamiento del derecho”, y su poca aplicación en la práctica que han hecho las Cortes en los Estados Unidos de América, tratándose de *software*: (Duffy & Hynes, 2016) Págs 37 y 69.

⁴⁵⁶ (Conrteras-Jaramillo, 2015) Pág. 298: “*Aunque se pueda tener una fuerte sensación sobre la relevancia internacional de la figura del agotamiento de los DPI, debido a su impacto en el intercambio global de bienes y servicios, lo cierto es que en los tratados internacionales esta figura no ha sido regulada, o se le ha otorgado una relativa libertad al derecho interno para implementarla (o no) de acuerdo a sus intereses sociales, políticos y/o económicos. En este sentido, el agotamiento del derecho abandona el escenario de política internacional para aterrizar en el campo de las políticas locales sobre participación en el comercio internacional. Es decir, el agotamiento de DPI no se caracteriza por tener una relación internacional → local; sino que obedece a una lógica nacional o comunitaria con efectos internacionales (local → internacional)*”.

que se cumpla el fin de estimular la innovación, mediante derechos de exclusiva sobre algoritmos no ensayados.

Frente al odioso principio de reciprocidad, que se funda en un mercantilismo normativo⁴⁵⁷ en vez de un trinitario imperativo categórico (político, económico y jurídico), es incuestionable su utilidad en la empresa pionera de consagrar un derecho de propiedad intelectual *sui generis* para el *software*, sin el apoyo de un Tratado internacional⁴⁵⁸; no solo porque la adopción del principio de reciprocidad puede ir generando la misma conciencia nacional que debe concordarse internacionalmente, más temprano que tarde, sino porque la mejora en la determinación de derechos a los creadores de *software*, procura subsecuentemente mejorar índices macroeconómicos⁴⁵⁹ locales en desempleo, inversión extranjera y divisas⁴⁶⁰, lo cual exige el establecimiento de firmas en el territorio, y no firmas turistas o golondrinas. En todo caso, cualquier preocupación sobre el alcance de la reciprocidad se minimiza si se crea un

⁴⁵⁷ (Nguyen , Gomulkiewicz , & Conway , 2012) Pág. 39: “[W]hether China chooses the route of Japan and France, or that of South Korea and Brazil, the step it will take in the direction of industrial copyright cannot be greater than that taken by the United States”.

⁴⁵⁸ (i) Consideramos mitigada la exposición internacional de las firmas que apuesten por el modelo de protección *sui generis* para el *software*, si se incorpora en su reglamentación el principio de reciprocidad, pasado por alto por (Dam, 1995) al tocar el mismo punto (Pág. 375), (ii) Ahora bien, lo ideal sería contemplar un derecho de prioridad para otras jurisdicciones: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2419, (iii) Sobre el derecho de prioridad: (Baylos Corroza, 2009) Pág. 1121.

⁴⁵⁹ (i) Resultados evidenciados en: (Junqueira Botelho, Stefanuto, & Veloso, 2007) Pág. 123, (ii) En alegato huérfano de apoyo empírico, considerando que las patentes de *software* inciden negativamente en el progreso de los países con menor desarrollo relativo: (Uribe Arbeláez, 2005) Pág. 218.

⁴⁶⁰ (The software industry and developing countries, 2012) Pág. 2: “The software sector in itself is also an area that holds potential for technological upgrading in low-income countries. In some developing countries, such as India, Argentina, Brazil, China, Costa Rica and South Africa, the software sector has grown significantly in recent years, generating new job opportunities, innovation and export revenue (see chapters II and III). As the entry barriers are relatively low in this sector, and with the growing trend towards the outsourcing of various software-related activities, software production is potentially of interest for countries at low levels of development. This was already recognized a decade ago by UNCTAD (UNCTAD, 2002, p. 34)”... Pág. 38: “For a Government, exports of software and IT services can be seen as an attractive option to generate foreign exchange, reduce trade deficits, induce job creation and transfer technology. Moreover, software exports can accelerate the integration into global value chains and contribute to economic diversification”.

registro constitutivo de derecho sobre el algoritmo expresado para ser explotado en formato de *software*, mientras seamos el único país que lo consagre.

Corolario de lo anterior, el agotamiento del derecho debe ser nacional y fundado en el primer acto de puesta a disposición del *software*, a fin de amparar a cabalidad los derechos exclusivos y de remuneración concedidos en Colombia. Es decir, la puesta a disposición⁴⁶¹ del algoritmo que haga parte de un *software*, y que pueda rastrearse en el mercado colombiano, prohibirá al titular restringir a terceros actos de comercio que tengan por objeto tal algoritmo, si tales actos se encuentran comprendidos dentro del alcance sustancial del acto de disposición y de la reivindicación hecha en la solicitud correspondiente, como por ejemplo la licencia onerosa del algoritmo adosado a un *software*, precedida de una cesión gratuita e ilimitada; pero los titulares

⁴⁶¹ (i) Consideramos que el empleo de la expresión “puesta a disposición” evita las complicaciones detectadas para la aplicación del agotamiento del derecho en el caso del *software*, en consideración a la declaración concertada respecto de los artículos 6 y 7 del Tratado de la OMPI WPT (Contreras-Jaramillo, 2016) Pág. 125: “[E]l derecho de distribución en particular, referido a una copia, sí es susceptible de ser agotado aun cuando esa copia sea digital. La condición digital no disminuye la característica tangible de una copia, al fin y al cabo esta será determinada, tanto en su extensión como en su ubicación y, en consecuencia, también será singularizada; es decir, se podrá diferenciar de otras copias que puedan tener características similares. De allí el término ‘bienes digitales tangibles’, mediante el cual, no solo se asegurará la aplicación de las mismas prerrogativas de los bienes analógicos en favor de aquellos que son digitales, sino que además se estará reconociendo una realidad social y jurídica innegable, la de la equivalencia funcional entre soportes”, (ii) (Merges, Menell, & Lemley, 2010) Pág. 569, (iii) Sobre lo que consideramos distingue en esencia un “agotamiento del derecho” es la posición de tercero que asume el titular inicial frente al actual, bien sea de todo o parte de los derechos de explotación. Al respecto: (Waelbroeck & Frignani, 1998) Pág. 957, 958 y 959: “No se puede objetar que el cesionario, ya que no puede adquirir más derechos que los que tenía el cedente antes de la cesión, no puede impedir la importación de productos a los que el cedente no habría podido oponerse por el hecho de que dicho productos hubieran sido vendidos, supuestamente, por él mismo o con su consentimiento. En realidad, el cesionario, cuando impide la importación de productos del cedente, no hace sino ejercer su derecho a estar protegido contra la importación de productos puestos en el mercado por un tercero no autorizado: ahora bien se trata aquí precisamente del derecho que se le ha transferido en virtud de la cesión. Si el cedente no podía, antes de la cesión, oponerse a esas importaciones, no es porque tuviera «menos derechos» que los que ha adquirido el cesionario, sino porque los productos en cuestión provenían de él mismo y no de un tercero no autorizado... En conclusión, pensamos que la cesión de todo derecho de propiedad industrial produce el resultado de colocar a las partes en la situación de terceros entre sí, con la condición, por supuesto, de que no subsista entre ellas ningún vínculo económico o financiero”, (iv) Sobre “ventas” de facto, a fin de evitar la regla de la “primera venta” (“first sale rule”): (Duffy & Hynes, 2016) Pág. 66: “[T]he most important characteristic of a true lease is that the lessor must retain ‘an economically meaningful residual interest’ in the property or, stated similarly, the property must be returned while ‘it retains some substantial economic life’”.

de derechos estarán facultados para ejercer sus derechos exclusivos o de remuneración, si la puesta a disposición no se rastrea dentro del territorio nacional, por ejemplo porque esa misma licencia de *software* proviene del Exterior⁴⁶².

Consideramos necesaria la consagración del agotamiento del derecho propuesto, en los términos planteados, porque la novedad legislativa propuesta exige señalar límites al derecho *erga omnes* diseñado, que la judicatura no podría encontrar según nuestro sistema de fuentes⁴⁶³, o que empezaría a analogizar, en detrimento de la deseada determinación de derechos⁴⁶⁴. Por otra parte, el incentivo a la innovación doméstica que procura esta nueva reglamentación podría menoscabarse si la explotación del algoritmo proviene del Exterior, donde no se consagran derechos como los postulados acá⁴⁶⁵, e igualmente si a las firmas sucesoras (“*follow-on*”) y

⁴⁶² Protestando contra la decisión en el caso (Pearson Education Inc. v LIU, 2009), bajo argumentos de perjudicar el bienestar general y contrariar precedentes judiciales, ya que la autoridad judicial sentenció que NO agotaba el derecho del titular las obras fueron manufacturadas en el Exterior, con permiso del titular: (Stockalper, 2010) Págs. 532 – 539.

⁴⁶³ (Duffy & Hynes, 2016) Pág. 44: “[I]f a statute delineates a very specific IP right, that right is unlikely to be curtailed by the first sale or exhaustion principle”. Más adelante, narrando lo decidido en el caso (Vernor v. Autodesk Inc., 2010) Pág. 72: “Autodesk produces sophisticated software for computer-aided design. At the time of the litigation, Autodesk sold the software on disks, but its customers had to accept a license to install the software. That license stated, among other things, that Autodesk’s customers were merely licensing a copy of the software; that they were prohibited from reselling the copy; and that Autodesk retained title to all of the copies. Vernor began purchasing and reselling copies of the software from Autodesk’s customers (who were obviously violating their licensing agreements). After Autodesk demanded that Vernor stop his re-sales because they constituted copyright infringement, Vernor sued for a declaratory judgment that the first sale doctrine protected his re-sales from Autodesk’s copyright infringement claim. Vernor lost. The Ninth Circuit ruled that Autodesk’s customers were mere licensees, not owners, of the software copies, and therefore Vernor was not protected by the first sale doctrine”.

⁴⁶⁴ (Tamayo Jaramillo, 2011) Pág. 863: “[P]ara que pueda crearse una norma mediante la analogía para colmar una laguna legal, no basta la similitud entre los presupuestos de una norma y las del caso que carece de norma aplicable, sino que es preciso que exista la misma razón, es decir que el intérprete pueda decirse así mismo que si el legislador hubiera imaginado la hipótesis no prevista en la norma que le sirve de apoyo para la creación por analogía, la habría incluido en ésta, dada la finalidad perseguida por él al redactar la norma”. Más adelante en la Pág. 867: “En efecto, en la creación analógica se genera una norma nueva a partir de otra norma preexistente que contiene la misma ratio legis y varias características de hecho similares aunque no idénticas, a las del caso carente de norma aplicable. En cambio, en la interpretación extensiva existe identidad tanto en las afinidades de los supuestos de hecho previstos en la norma, como en la ratio legis y, por lo tanto, de lo que se trata es de hacer extensiva la aplicación de la norma preexistente a un caso idéntico pero no contemplado expresamente en ella, sin necesidad de acudir a la creación analógica”.

⁴⁶⁵ De hecho, sería absurdo considerar que se ha “agotado el derecho” del titular, internacional o regionalmente. (i) Sobre el alcance de los tipos de “agotamientos de derecho” y lo que ha decidido el Tribunal Europeo de Justicia cuando

competidoras (“*me too*”) locales se les grava con costos de transacción derivados de sucesivas autorizaciones, al estilo del principio de independencia de los actos de explotación de obras protegidas por el Derecho de Autor, como fue explicado en el punto 2.2. del Capítulo I.

Ahora bien, teniendo en cuenta que los derechos constituidos por el registro planteado no podrán exigirse en el Exterior, mientras no se consagre normas en este mismo sentido, conviene dejar a los creadores de *software* que exijan sus derechos fuera de nuestras fronteras, a través de los derechos internacionalmente admitidos para el *software*, o en el contorno de las estipulaciones contractuales que hubieren acordado⁴⁶⁶.

Es posible que el seguimiento a los efectos de la normas sugieran un cambio en algunas posiciones, pero podemos anticipar que mediante este modelo es factible atraer inversión extranjera y/o divisas vía exportaciones, y puede generarse más empleo en la industria del *software*, pues el titular del derecho podrá defenderse apropiadoramente del *cloning*, los *estándares de facto* no se generarán por el exagerado término de protección con el que cuenta el

se exportan bienes que contienen tecnología patentada, a territorios donde no se cuenta con esa protección: (Hanrahan, 2010) Págs. 8 – 11, (ii) Sobre las interpretaciones formalistas hechas por la Corte Suprema de Justicia de los Estados Unidos de América, por las cuales el agotamiento del derecho NO se extiende a las obras manufacturadas en el Exterior, por el alcance extraterritorial de la Ley que ello implicaría: (Paul, 2012) Págs. 183 – 185.

⁴⁶⁶ (Duffy & Hynes, 2016) Pág. 10: “[B]ecause of the exhaustion doctrine, general commercial law applies much the same whether the underlying goods embody IP or not. Stated another way, for transactions in goods embodying IP (which today includes everything from books and DVDs to automobiles and dishwashers), routine commercial law issues—such as whether sales contracts can restrict purchasers’ rights in using or reselling the goods, or whether the goods can be encumbered to bind downstream owners—are governed by the rules and remedies found in commercial law generally (that is, in the Uniform Commercial Code (“UCC”); state contract and property law; federal and state antitrust law; etc.), with that law neither supplemented nor supplanted by statutory IP law”.

*software*⁴⁶⁷, recuperando el gremio su autonomía para fijar el *standard*⁴⁶⁸, y la descompilación y/o ingeniería reversa se podría reemplazar por la motivación para crear nuevos productos⁴⁶⁹.

2.3. Dejemos los derechos morales para las obras protegidas por el Derecho de Autor.

Finalmente, como característica descollante del modelo de protección *sui generis* propuesto, del que anexamos un borrador de trabajo, podrán concederse derechos *sui generis* de protección para el *software* a personas naturales o jurídicas, como titulares originarios⁴⁷⁰, sin oportunidad para reclamar derechos morales⁴⁷¹ sobre el algoritmo expresado o el uso dado al algoritmo.

⁴⁶⁷ (Menell, Tailoring legal protection for computer software, 1987) Págs. 1341 – 1342.

⁴⁶⁸ (i) Nos referimos a la definición de parámetros homogéneos de interoperabilidad entre *hardware* y *software* mediante un proceso concertado gremialmente, y no mediante un acto de abuso de posición dominante. Sobre la divulgación concertada de las patentes de *software*, las licencias a precios razonables y no-discriminatorios (“FRAND”, por sus siglas en inglés) y su litigiosidad: (Contreras, 2016) Págs. 863 - 867, (ii) Proponiendo una participación gremial en el refinamiento de las normas jurídicas a medida que la industria avanza en desarrollos: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2408.

⁴⁶⁹ (Dam, 1995) Pág. 358: “*Permitting me-too copying is not necessary to permit future innovation because, by definition, the me-too copier does not seek to innovate. Moreover, permitting me-too copying restricts the incentives for the first-generation firm to innovate*”.

⁴⁷⁰ (Lema, Quadros, & Schmitz, 2012) Pág. 57.

⁴⁷¹ (i) (Gómez Segade, 2001) Pág. 863: “[E]xiste el reconocimiento general de que algunos derechos morales no se adaptan a la naturaleza y funciones de los programas de ordenador, y se pueden introducir matices... Por lo general, cuanto más funcional sea la obra, menos derechos morales se derivarán, y los programas de ordenador están muy arriba en la lista de obras funcionales”, (ii) (Calle D'Alleman, 2012) Pág. 186: “[E]l análisis [refiriéndose a la pertinencia del reconocimiento de derechos morales sobre el *software*], una vez más, debe proceder sobre la naturaleza de estas creaciones, admitiendo que, por su carácter funcional y utilitario, no es pertinente a ella el reconocimiento de prerrogativas morales, como sucede con las creaciones propias de la propiedad industrial”, (iii) (Palacio Eastman, 1999) Pág. 215: “*En cuanto a los derechos morales consagrados por el derecho de autor en las legislaciones de civil law, podríamos decir que son un obstáculo para el desarrollo del software, pues por sus características de inalienabilidad, perpetuidad e imprescriptibilidad, no permitirían que un software se modificara o adaptara a las nuevas necesidades, ni se desarrollaran aspectos aplicativos que el mismo software pudiese contener*”.

En Colombia, se dieron las “*primeras de cambio*” en el PND 2011 - 2014 hacia una aproximación a la realidad del régimen jurídico para la creación de obras protegidas por el Derecho de Autor, cuando éstas son creadas por equipos de personas bajo el auspicio de una firma especializada⁴⁷²; pero una eficiente protección jurídica para el *software*, como para cualquier otro asunto, debe ser menos tibia al momento de asignar titularidad y demás facultades sobre la creación. Por eso, el régimen propuesto sigue la lógica del registro constitutivo de derechos, en favor del sujeto de derecho que presente su solicitud.

Ahora, teniendo entre manos una creación que de todo tiene y es valiosa, menos por su contenido estético, entonces lo concerniente a la extensión de la personalidad⁴⁷³, como fundamento de los derechos morales, debe ser abandonado por completo. En primer lugar, pues la eficiencia es la ley natural que rige la aplicación de algoritmos en programas de ordenador, y detrás de ella el derecho no puede secundar santuarios personales a individuos, y reverencia a sus descendientes, cuyo mérito fue haberse plegado a la inflexible eficiencia⁴⁷⁴; y por otra parte,

⁴⁷² (i) Nos referimos a los artículos 28 a 30 del PND, Ley 1450 de 2011 (El artículo 30 de la Ley 1450 de 2011 quedó subrogado tácitamente por el artículo 181 de la Ley 1955/19), que introducen modificaciones de grande importancia a la Ley 23 de 1982, concretamente en la presunción de titularidad autoral en cabeza del contratante en contratos de prestación de servicios, o del empleador en contratos laborales; abriendo la puerta incluso a la titularidad originaria en cabeza de personas jurídicas. (ii) Previamente (Conpes 3678, 2010) Pág. 51: “*El objetivo general del sector [Refiriéndose a software & IT] es promover la competitividad de la industria para aprovechar las oportunidades que ofrece el mercado internacional, apoyando su fortalecimiento mediante acciones que mejoren la disponibilidad de recurso humano pertinente y bilingüe, faciliten el acceso a capital y provean un marco regulatorio adecuado a la dinámica del sector y a los estándares internacionales. Para lograr este objetivo se ha identificado que es necesario trabajar en cinco aspectos estratégicos: ... 3. Desarrollar un marco normativo general que se ajuste a la evolución de la industria e incorpore estándares internacionales... 5. Crear e implementar una estrategia sectorial de I+D+i que promueva el desarrollo de productos y servicios de software que respondan al proceso de internacionalización de manera competitiva y sostenible*”.

⁴⁷³ (i) (Calle D'Alleman, 2012) Pág 274, (ii) (Preinfalk Lavagni, 2010) Pág. 130, (iii) Reconstruyendo los fundamentos políticos del Decreto de 1791, por el cual se re-instituían derechos de autor a creadores de obras dramáticas, pausados por la libertad de conciencia y expresión consignadas en la Declaración de los Derechos del Hombre y del Ciudadano de 1789 (Cavalli, 2006) Pág. 57: “*En su informe, Le Chapelier pronunció esta frase hoy célebre: «La más sagrada, la más legítima, la más inatacable, y si se me permite decirlo, la más personal de las propiedades es la obra, fruto del espíritu de un escritor»*”, (iv) (Baylos Corroza, 2009) Pág. 502, 513 – 536.

⁴⁷⁴ (Paterson, 2013) Pág. 189.

porque la codificación automatizada y la inteligencia artificial no pueden dar pie a absurdas extensiones del derecho que se contempla, como sería otorgarle derecho morales a seres que no son sujetos de derecho⁴⁷⁵.

Debe tenerse presente que nada de lo aquí planteado elimina la protección de elementos no reivindicados en el algoritmo aplicado al *software*, como serían las obras manipuladas y reproducidas que procese el *software*⁴⁷⁶, como tampoco se estaría restringiendo aplausos y reconocimientos, públicos y privados, a quienes de forma ingeniosa le encuentren aplicación informática a un adelanto matemático; lo que sí se está evitando es que se difumine el contorno del derecho que se asigna, en una maraña de facultades de diferentes estirpes en favor de múltiples titulares⁴⁷⁷.

⁴⁷⁵ Como sería concederle derechos a una máquina. Sobre lo que viene a futuro en la ciencia: (Rifkin, 2009) Pág. 272: “*La integración final de las Ciencias de la información y de la vida se produce en la forma del «ordenador molecular», una máquina pensante que no estará hecha de silicio sino de hebras de ADN... Al contrario de los ordenadores corrientes, que son secuenciales y sólo pueden hacer una cosa cada vez, el ADN es una máquina computadora que trabaja en paralelo a gran escala: en teoría puede computar 100.000 billones de cosas a la vez*”.

⁴⁷⁶ (Paterson, 2013) Pág. 187.

⁴⁷⁷ (i) No estamos a favor de consagrar derechos a favor de los usuarios, particularmente de acceder al código fuente, como propone: (Calle D'Alleman, 2012) Pág. 300. Un sistema de registro constitutivo de derechos, como el que proponemos, solucionaría el interés de acceder al código fuente del público. (ii) Una cosa es mantener contacto con los usuarios para desarrollar productos, y otra cosa es concederle derechos sobre el *software*: (Lema, Quadros, & Schmitz, 2012) Pág. 35.

3. ¿Quiénes aprovecharían un nuevo modelo de protección para el *software*?

Como se indica al inicio de este capítulo, debe admitirse un cierto grado de especulación alrededor de un sistema de protección no ensayado⁴⁷⁸, así se funde en la combinación de reglas conocidas de la propiedad intelectual⁴⁷⁹; sin embargo, es previsible que el sistema de protección *sui generis* propuesto sea aprovechado en una primera etapa por firmas emprendedoras (“*start-up*”) o no-estratégicas (“*spin-off*”)⁴⁸⁰, las cuales procuran diferenciarse de las empresas establecidas en el mercado⁴⁸¹ y son generalmente pequeñas y medianas empresas generadoras de empleo, que ni pueden responder con acciones legales en contra del *cloning*, ni cuentan con el personal o los activos suficientes para atraer capital de riesgo, formular y aprovechar las deducciones tributarias dispuestas para (I+D+i)⁴⁸².

⁴⁷⁸ (Calabresi, 1991) Pág. 1228: “*Distributional analysis becomes inevitable and hence essential, and economists must, at a minimum, become explicit about the distributional judgments (or guesses) that they are making. Decisionmakers can then accept or reject such judgments and guesses as they choose*”.

⁴⁷⁹ (i) (Menell, Tailoring legal protection for computer software, 1987) Pág. 1338: “*It cannot be assumed automatically, therefore, that expansive legal protection for intellectual property at any one level will generate both the optimal amount of innovation and the optimal diffusion path*”, (ii) (Calabresi, 1991) Págs. 1221 y 1227.

⁴⁸⁰ (i) (Reinganum, 1983) Pág. 742: “*This is clearly true when there is no uncertainty in the innovation process. The natural extension of this result to the case of uncertain innovation is that the incumbent is more likely to patent the innovation than is the challenger*”, (ii) (Bases del plan nacional de desarrollo 2018 - 2022, 2019) Pág. 541.

⁴⁸¹ (Menell, Tailoring legal protection for computer software, 1987) Pág. 1341: “*Professors Katz and Shapiro demonstrate that firms might prefer to adopt noncompatible product standards even though their adoption of compatible products would increase net social welfare. The explanation for this behavior is that by adopting a compatible standard, a firm enlarges the size of a network that comprises both the adopter's product and its rivals' products. This will have the effect of increasing the desirability of the rivals' products to consumers, thereby reducing the adopter's market share (although of a larger market) relative to what it would have been had the firm adopted a noncompatible product standard*”.

⁴⁸² (Conpes 3834, 2015) Pág. 25: “*Al profundizar en el tipo de contribuyentes a los cuales se les ha aprobado proyectos, se observa que entre 2001 y 2014 la tasa de aprobación de las empresas grandes fue superior que el promedio, ya que el 79% de los proyectos presentados por ellas fueron aprobados (Tabla 3). De forma particular, se identificó que una sola empresa beneficiaria ha obtenido cerca de 383 proyectos aprobados durante los últimos 13 años, lo cual evidencia la concentración no solo en la asignación de recursos, sino también en la presentación de proyectos. Una de las principales razones de esta tendencia, se relaciona con la capacidad de formulación de proyectos para hacer uso de los beneficios tributarios en CTI. Las grandes empresas que presentaron proyectos para hacer uso de beneficios tributarios entre 2001 y 2014, registraron en promedio la presentación de 44 proyectos*”.

Esto, por cuanto las firmas establecidas probablemente ya han entrado en la onda de las patentes de *software*, hallándose satisfechas y en procura de unas nuevas solicitudes como las previamente presentadas; pero en una segunda etapa de consolidación del sistema, a medida que se vayan dando casos de éxito para las firmas de la categoría anterior, las firmas establecidas empezarán también a incursionar en esta forma de protección para el *software*, con modelos financieros y de negocios acordes con los tiempos de protección reducidos que contempla, y las limitantes al criterio de novedad que sobrevendrá para la concesión de patentes de *software* con un registro público de algoritmos⁴⁸³.

Debido a que la protección vía Derechos de Autor no les permite proceder en contra del *cloning*, y esta externalidad desincentiva la permanente producción de productos innovadores, claramente la decisión que estos deberán tomar, en algún punto en el tiempo, es respecto a si conservan secreto un desarrollo, hasta que lo exploten y se incluye al “estado de la técnica”⁴⁸⁴, con los costos que esto implica, o intercambian su revelación por diez años de explotación comercial del algoritmo.

La acogida que recibirían estas decisiones en el Derecho proviene no solo del beneficio social derivado de un único e ideal cuerpo normativo para defender sus desarrollos⁴⁸⁵, sino en el

anuales durante todo el período, una capacidad que sin duda rezaga la oportunidad de competir con empresas que tienen menos capacidades para formular proyectos de CTI con calidad técnica”.

⁴⁸³ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Págs. 2384 – 2385: “Consequently, a market-oriented legal regime that protected algorithms would regulate only commercial implementations of algorithms, whereas patent law regulates virtually all uses of inventions. It would preserve the trade secrecy option for algorithms, but might also encourage disclosure by providing a registration system through which developers of innovative algorithms could seek a period of blockage or compensation for use of their innovations. This would likely lessen the pressure that in recent years has led more software developers to seek patents for algorithms”

⁴⁸⁴ (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2410

⁴⁸⁵ (i) Considerando que un sistema de protección *sui generis* no impediría la concurrencia de modelos de protección, y en consecuencia fracasaría en su acogida: (Leith, 2007) Pág. 181, (ii) (Peña Valenzuela, 2013; Peña Valenzuela,

aumento de los presupuestos para (I+D+i) de productos innovadores. Al respecto, como trabajo de campo se hizo una encuesta pública entre programadores y firmas de la industria del *software*, sin discriminar el tamaño de la firma, en la cual se les preguntó si estarían dispuestos a registrar un algoritmo diseñado para *software*, si a cambio le permitieran ser el único en explotarlo por un período de tiempo que les habilitara recuperar los costos para su diseño.

El ochenta y seis por ciento (86%) de los encuestados manifestó estar dispuesto a registrarlo y de estos, un cincuenta y siete por ciento (57%) estaría dispuesto a hacerlo, aunque el lenguaje de programación empleado hubiera sido de código abierto, como les fue preguntado a continuación a quienes hubieren respondido afirmativamente a la pregunta del registro del algoritmo.

En la misma encuesta se les preguntó si aumentarían el tiempo y personal para el desarrollo de productos innovadores, si se les permitiera ser el único autorizado para explotar un algoritmo diseñado para *software*. Un cuarenta y tres por ciento (43%) de los encuestados respondió que definitivamente sí lo aumentaría, y otro tanto (43%) respondió que probablemente sí lo aumentaría⁴⁸⁶.

2013) Pág. 39: “*Otro sector de la doctrina se opone a la implementación de un sistema de protección sui generis, pues considera que este tipo de protección genera inseguridad jurídica frente a los sistemas jurídicos que actualmente existen*”.

⁴⁸⁶ El incentivo tributario contenido en los artículos 168, 170 y 171 de la Ley 1955/19 no se aprovecha en sí mismo, pues el resultado de la inversión en (I+D+i) debe ser bien protegido por las normas jurídicas, a fin de que generen los ingresos esperados para recuperar los costos de los innovadores. En todo caso, aunque probablemente bien intencionado, es poco probable la vinculación de personal doctorado en Mipymes, en consideración a los costos laborales que ello implicaría.

Los resultados anteriores indican que hace falta un incentivo en la industria para tomar el riesgo de innovar⁴⁸⁷, sin importar el tamaño de la firma o el lenguaje de programación empleado en el algoritmo⁴⁸⁸, y también permiten concluir que si esa hipotética voluntad se concretara en la realidad, la competencia entre las firmas del sector no se concentraría en la aparición de productos sustitutos y/o complementarios, sino en novedosos sistemas operativos, aplicativos o *drivers*⁴⁸⁹; lo cual redundará en una oferta más amplia, diversificada y vanguardista de *software* y *hardware*⁴⁹⁰, en simultáneo con una mejora en índices macroeconómicos en divisas, empleo y recaudo de impuestos⁴⁹¹.

⁴⁸⁷ (i) Resaltando la importancia de una vocación hacia la innovación (“*experimentation*”) que debe observarse en la industria para que se desarrolle, y que justifica alternativas diferentes a las seguidas por países como la India, Irlanda e Israel: (Junqueira Botelho, Stefanuto, & Veloso, 2007) Págs. 114 y 122, (ii) Lamentando la falta de documentación respecto a la percepción en países en desarrollo sobre el rol que la propiedad intelectual para la innovación: (López, 2009) Pág. 5, (iii) Describiendo el punto en el cual las firmas no toman el riesgo de la innovación, por el juego competitivo y no colaborativo del mercado, lo mismo que planteando el propósito general de permitir al innovador recuperar sus costos: (Samuelson, Davis, Mitchell D., & Reichman, 1994) Pág. 2340 y 2366.

⁴⁸⁸ El resultado anterior es consecuente con lo indicado aquí: (The software industry and developing countries, 2012) Pág. 100: “*In deciding its policy on intellectual property in the software sector, each country must reflect upon its level of development, status of domestic software sector capabilities and current system for intellectual property protection, including its capacity to enforce intellectual property law and provide legal recourse. In principle, stronger intellectual property protection creates stronger incentives to develop new products – open source as well as proprietary software. However, for the local industry to benefit from such protection, it needs the capabilities to produce what the market demands. Thus, what initially seems to be a policy choice of favouring more or less FOSS or open standards becomes a nuanced decision dependent on local conditions*”.

⁴⁸⁹ (Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia, 2014) Pag. 13: “*El documento “Global Cloud Computing Scorecard” evalúa las políticas de adopción del cloud computing en las 24 economías mundiales líderes en TI para el año 2012. Con este fin, analiza las normas y reglamentos vigentes en cada país respecto de los siguientes temas: privacidad de los datos, ciberseguridad, cibercrimen, propiedad intelectual, interoperabilidad de la tecnología y armonización legal, libre comercio e infraestructura de TI... Protección de la propiedad intelectual: las normas deben ofrecer una protección efectiva sobre los derechos adquiridos por los desarrolladores de nuevos productos y/o tecnologías en la nube y así incentivar a las empresas a mantener el proceso innovativo*”.

⁴⁹⁰ (Menell, Tailoring legal protection for computer software, 1987) Pág. 1358.

⁴⁹¹ Interesantemente se ha demostrado el importante aporte de la industria, en época de “vacas flacas”: (Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia, 2014) Pág. 40: “*La gráfica evidencia que el comportamiento del sector de software frente al total de empresas de la economía es opuesto: en los años que uno crece el otro decrece y viceversa. Se destaca el crecimiento del nivel de endeudamiento de la industria de software entre los años 2011 y 2012 -8.8% - que contrasta con la caída de 2.1% en el indicador del total de empresas*”.

En nuestro concepto, *un nuevo modelo de protección para el software* como el propuesto acompaña perfectamente los propósitos manifestados en el Conpes 3834 de 2015, por el cual se definen:

“[L]ineamientos alrededor de dos objetivos de política: i) incentivar la inversión privada en actividades de CTI a través de las deducciones tributarias; ii) mejorar las condiciones y alternativas de acceso a las deducciones tributarias. Para lograrlo se recomienda, entre otras acciones, ampliar el ámbito de aplicación del instrumento para que incluya proyectos de innovación como lo establece la Ley 1739 de 2014, establecer un esquema de calificación automática para empresas altamente innovadoras, y habilitar a empresas en etapa temprana y Pymes que, aunque no generan renta líquida hoy en día, podrán deducir estas inversiones en innovación en el futuro cuando deban realizar contribuciones a través del impuesto de renta”.

Aprovechando la legitimidad política que indudablemente reportan los documentos Conpes en Colombia⁴⁹², el borrador de **“Proyecto de Ley mediante el cual se protege el algoritmo, o la combinación de algoritmos, contenidos en un programa de ordenador”** acoge los lineamientos y proposiciones hechos en el Conpes 3834 de 2015, así como el siguiente apartado del documento Bases del Plan Nacional de Desarrollo 2018 – 2022 (Página 655)⁴⁹³:

⁴⁹² Artículo 17 Ley 152 de 1994

⁴⁹³ En sentido similar: Informe Nacional de Competitividad 2018 – 2019 *“Es necesario que el Gobierno cree un marco apropiado para el desarrollo del emprendimiento de base digital, lo que puede incluir acciones como incentivos para*

“MinTIC, en coordinación con las entidades que corresponda, identificará y removerá las barreras que obstaculizan el desarrollo de la industria a escala nacional, el aumento de la competencia, la inversión extranjera, el despliegue de grandes centros de datos, entre otros. La promoción de esta industria se focalizará en el desarrollo de herramientas (software, aplicaciones, etc.) que funcionen sobre Internet, que respondan a los estándares internacionales de calidad de software y que estén dentro de un esquema de trabajo conjunto entre el sector productivo y la industria TI para garantizar la pertinencia de los productos resultantes de la innovación”.

emprendedores tecnológicos, asesoría en diferentes aspectos de la creación de empresas y el mejoramiento de la propia infraestructura digital del Gobierno”.

Capítulo V. Texto del nuevo modelo de protección jurídica del *software*

A fin de estimular la creación de *software* innovador, impulsar la industria nacional de *software*, mejorar índices macroeconómicos en el mediano y largo plazo, y a fin de solucionar el déficit de protección presente en el *software*, producto de la aplicación de un sistema de protección lejano a la realidad y desarrollos evidenciados en la industria; a continuación se postula un nuevo modelo de protección para el *software*, en el marco de los Tratados internacionales de los cuales la República de Colombia es parte:

Proyecto de Ley mediante el cual se protege el algoritmo, o la combinación de algoritmos, contenidos en un programa de ordenador.

Título I. Disposiciones generales y fundamentos para la protección

Capítulo I. Observancia de Tratados internacionales y normas supranacionales.

Artículo 1. Ninguna disposición de las contenidas en la presente Ley irá en detrimento de las normas del Convenio de París para la Protección de la Propiedad Industrial, del Convenio de Berna para la Protección de las Obras Literarias y Artísticas (Convenio de Berna), del Acuerdo sobre los Aspectos de los Derechos de Propiedad Intelectual Relacionados con el Comercio (ADPIC), del Tratado de la OMPI -Organización Mundial de la Propiedad Intelectual- sobre Derechos de Autor (WCT), de la Decisión 351 de la Comisión de la Comunidad Andina de Naciones y de la Decisión 486 de la Comisión de la Comunidad Andina de Naciones.

Capítulo II. Principios rectores de la protección

Artículo 2. De acuerdo con los artículos 61 y 71 de la Constitución Política de Colombia, mediante las formalidades establecidas en la presente Ley se protege la propiedad intelectual respecto de los programas de ordenador.

Los sujetos de derecho extranjeros no domiciliados en Colombia gozarán de la protección de esta Ley de conformidad con los Tratados internacionales a los cuales Colombia se adhiera o cuando las leyes nacionales del otro país impliquen reciprocidad efectiva en la protección de los derechos consagrados a los titulares de derechos de propiedad intelectual respecto de los programas de ordenador.

Artículo 3. Los derechos de propiedad intelectual respecto del algoritmo o combinación de algoritmos que sean concedidos según lo indicado en la presente Ley excluyen otro tipo de protección jurídica respecto de la funcionalidad del programa de ordenador correspondiente.

Artículo 4. Sujeto a lo dispuesto en los artículos siguientes, se concederá un derecho de propiedad intelectual respecto de los algoritmos que compongan un programa de ordenador, siempre que sean nuevos y tengan nivel inventivo.

La protección que se concede no exigirá que el algoritmo o combinación de algoritmos se encuentre inserto en un soporte físico o en un producto, mientras se trate de un algoritmo o

combinación de algoritmos implementado en un programa de ordenador ejecutable, cuyas funcionalidades se encuentren descritas conforme se indica en los artículos 14 y 15 de esta Ley.

Artículo 5. Un algoritmo se considerará nuevo cuando no esté comprendido en el estado de la técnica.

El estado de la técnica comprenderá todos aquellos algoritmos que hayan sido explotados gratuita u onerosamente, dentro del territorio nacional, mediante su implementación en un programa de ordenador ejecutable.

Artículo 6. Para efectos de determinar la novedad, no se tomará en consideración la explotación ocurrida dentro del año precedente a la fecha de la presentación de la solicitud de protección, siempre que tal explotación hubiese provenido del sujeto de derecho creador del programa de ordenador, o algún otro sujeto autorizado para ello, cuyo algoritmo o combinación de algoritmo sea el objeto de la protección solicitada conforme a esta Ley.

Tampoco afectará la novedad el debate probatorio que ocurra en los procesos que tratan los artículos 56 y 57 de esta Ley, a menos que se advierta temeridad o mala fe según lo dispuesto en el artículo 79 de la Ley 1564 de 2012.

Salvo autorización expresa de las partes, el expediente no se podrá consultar por terceros, ni tomar copia de las actuaciones escritas u orales debatidas en tales procesos. A las audiencias

podrán asistir únicamente las personas autorizadas por las partes, y aquellos que hayan sido citados por el juez.

Artículo 7. Se considerará que un algoritmo tiene nivel inventivo, si para una persona versada en codificación de programas de ordenador ese algoritmo o combinación de algoritmos, su uso o implementación en un programa de ordenador, no es corriente en el momento de presentación de la solicitud.

La ingeniosa combinación de algoritmos previamente conocidos es suficiente para cumplir con el requisito de nivel inventivo.

Artículo 8. Los algoritmos comprendidos en el estado de la técnica podrán ser objeto de protección según las formalidades de esta Ley, si:

1. Es implementado para un programa de ordenador funcionalmente distinto al programa de ordenador asociado al primer registro otorgado;
2. Se cumple el criterio de nivel inventivo indicado en el artículo 7 de esta Ley.

Capítulo III. Requisitos de la protección del algoritmo o combinación de algoritmos implementados para programas de ordenador.

Artículo 9. No podrán ser objeto de protección los algoritmos o combinación de algoritmos que:

1. No se encuentre implementado, o no se encuentren implementados, en un programa de ordenador ejecutable;
2. Sean idénticos o se asemejen a un algoritmo o combinación de algoritmos anteriormente solicitado para registro conforme esta Ley, o registrado por un tercero conforme esta Ley, implementado para las mismas funcionalidades de un programa de ordenador previamente asociado a un registro o solicitud de registro en trámite;
3. Sean idénticos o se asemejen a las reivindicaciones de una patente de invención o procedimiento anteriormente solicitada para registro, o registrada, en el territorio de la República de Colombia;
4. Infrinja el derecho de propiedad industrial o el derecho de autor reconocido en el territorio de la República de Colombia a un tercero, salvo que medie el consentimiento de éste.

Artículo 10. Cuando la Superintendencia de Industria y Comercio tenga indicios razonables que le permitan inferir que un registro se hubiese solicitado para perpetrar, facilitar o consolidar un acto de competencia desleal dentro del territorio de la República de Colombia, podrá denegar dicho registro.

Título II. Solicitud de protección, concesión del derecho y registro de actos

Capítulo I. Características y contenido de las solicitudes de protección

Artículo 11. La solicitud de protección solo podrá comprender un algoritmo o una combinación de algoritmos relacionadas entre sí, de manera que conformen una única funcionalidad con el programa de ordenador respectivo.

Artículo 12. La solicitud para obtener protección sobre un algoritmo o combinación de algoritmos se presentará ante la Superintendencia de Industria y Comercio y deberá contener lo siguiente:

1. El petitorio;
2. La descripción;
3. Una o más reivindicaciones, que se refieran a la funcionalidad del algoritmo en el programa de ordenador correspondiente;
4. Uno o más dibujos, cuando fuesen necesarios para comprender la funcionalidad del algoritmo en el programa de ordenador correspondiente, los que se considerarán parte integrante de la descripción;
5. El resumen;
6. Los poderes que fuesen necesarios;

7. El comprobante de pago de la tasa establecida;
8. De ser el caso, la copia del documento en el que conste la transferencia de los derechos sobre la expectativa de protección.

Artículo 13. El petitorio de la solicitud estará contenido en un formulario y comprenderá lo siguiente:

1. El requerimiento de concesión del derecho de protección;
2. El nombre y la dirección del sujeto de derecho solicitante;
3. La nacionalidad o domicilio del sujeto de derecho solicitante. Cuando no fuere una persona natural, deberá indicarse el lugar de creación del sujeto de derecho;
4. El nombre del programa de ordenador en el cual se implementa el algoritmo o combinación de algoritmos;
5. El nombre y el domicilio del sujeto de derecho creador del programa de ordenador, cuando no fuese el mismo solicitante;
6. De ser el caso, el nombre y la dirección del representante legal del solicitante;

7. La firma del sujeto de derecho solicitante o de su representante legal, y;
8. De ser el caso, la fecha, el número y la oficina de presentación de toda solicitud de protección u otro título de protección que se hubiese presentado u obtenido en el Extranjero por el mismo solicitante o su causante y que se refiera total o parcialmente al mismo algoritmo o combinación de algoritmos reivindicado en la solicitud de protección.

Artículo 14. La descripción deberá divulgar el algoritmo o combinación de algoritmos de manera suficientemente clara y completa para su comprensión, y para que una persona capacitada en codificación de programas de ordenador pueda implementarlo en un programa de ordenador.

En particular, deberá incluir la siguiente información:

1. La funcionalidad anterior que cumplía el algoritmo o combinación de algoritmos que sea conocida por el solicitante y fuese útil para la comprensión y el examen de la solicitud, y las referencias a los documentos y publicaciones anteriores relativas a dicha tecnología;
2. Una descripción de las funcionalidades del programa de ordenador en el cual se implementa el algoritmo o combinación de algoritmos, exponiendo las diferencias y eventuales ventajas con respecto a algoritmos implementados para funcionalidades similares;
3. Una reseña sobre los dibujos, cuando los hubiera.

Artículo 15. Las reivindicaciones definirán el algoritmo o combinación de algoritmos que se desea proteger. Deben ser claras, concisas y estar enteramente sustentadas por la descripción. Las reivindicaciones podrán ser independientes o dependientes. Una reivindicación será independiente cuando defina el algoritmo o combinación de algoritmos que se desea proteger sin referencia a otra reivindicación anterior. Una reivindicación será dependiente cuando defina el algoritmo que se desea proteger refiriéndose a una reivindicación anterior.

Una reivindicación que se refiera a dos o más reivindicaciones anteriores se considerará una reivindicación dependiente múltiple.

Artículo 16. El resumen consistirá en una síntesis de la divulgación del algoritmo o combinación de algoritmos contenida en la solicitud de protección. Dicho resumen servirá solo para fines de información técnica y no tendrá efecto alguno para interpretar el alcance de la protección conferida al titular del derecho.

Artículo 17. Se considerará como fecha de presentación de la solicitud, la de su recepción por la Superintendencia de Industria y Comercio, siempre que al momento de la recepción hubiera contenido al menos lo siguiente:

1. La indicación de que se solicita la protección para un algoritmo o combinación de algoritmos;
2. Los datos de identificación y contacto del sujeto de derecho solicitante;

3. La descripción de la funcionalidad que cumple el algoritmo o combinación de algoritmos en el respectivo programa de ordenador;
4. Los dibujos, de ser éstos pertinentes, y;
5. El comprobante de pago de las tasas establecidas.

La ausencia de alguno de los requisitos enumerados en el presente artículo, ocasionará que la solicitud sea considerada por la Superintendencia de Industria y Comercio como no admitida a trámite y no se le asignará fecha de presentación.

Artículo 18. El sujeto de derecho solicitante de la protección podrá actuar, en lo que sea concorde a esta Ley, conforme los artículos 34 a 38 de la Decisión 486 de la Comisión de la Comunidad Andina de Naciones, mediante el pago de la tasa correspondiente.

Capítulo II. Del trámite de la solicitud

Artículo 19. La Superintendencia de Industria y Comercio publicará la solicitud que cumpla los requisitos de forma previstos en los artículos 12 y 13 de esta Ley, dentro de los treinta (30) días siguientes a la fecha en la cual se haya presentado dicha solicitud; de lo contrario, dará aplicación al trámite previsto en los artículos 39 y 41 de la Decisión 486 de la Comisión de la Comunidad Andina de Naciones.

Antes de la publicación acá indicada, la Superintendencia de Industria y Comercio conservará en absoluta reserva la información contenida en la solicitud.

El contenido de la publicación acá indicada será publicado igualmente en el mecanismo de difusión que hubieren dispuesto las Oficinas de transferencia de resultados de investigación – OTRIS -.

Artículo 20. Hecha la publicación anterior, quien tenga legítimo interés podrá oponerse al registro solicitado. Para el efecto, se seguirá el trámite previsto en los artículos 42 y 43 de la Decisión 486 de la Comisión de la Comunidad Andina de Naciones.

Artículo 21. A efectos de lo previsto en el artículo anterior, se entenderá que tienen legítimo interés para presentar oposiciones los titulares de derechos sobre algoritmos o combinación de algoritmos implementados en un programa de ordenador domiciliados fuera del territorio nacional, en donde se proteja de forma efectiva al algoritmo o combinación de algoritmos implementados en un programa de ordenador mediante un sistema de protección semejante al señalado en esta Ley.

En este caso, el opositor deberá acreditar su interés real en el mercado colombiano, debiendo a tal efecto solicitar la protección según la presente Ley al momento de interponer la oposición.

Artículo 22. La interposición de una oposición con base en un registro Extranjero según el artículo 21 de esta Ley, facultará a la Superintendencia de Industria y Comercio a denegar la protección solicitada.

Artículo 23. La interposición de una oposición con base en una solicitud de registro Extranjero según el artículo 21 de esta Ley, que tenga fecha anterior a la solicitud de protección local, acarreará la suspensión del trámite de la solicitud de protección local, hasta tanto se decida sobre el registro de la solicitud de protección Extranjera.

Artículo 24. La Superintendencia de Industria y Comercio no considerará admitidas a trámite las oposiciones que estén comprendidas en alguno de los siguientes casos:

1. Que se presenten sin indicar los datos esenciales relativos al opositor y a la solicitud contra la cual se interpone la oposición;
2. Que la oposición fuere presentada extemporáneamente;
3. Que no haya pagado las tasas de tramitación correspondientes.

Artículo 25. Vencido en silencio el plazo para presentar oposiciones, o transcurrido el trámite señalado en el artículo 20 de esta Ley, la Superintendencia de Industria y Comercio procederá a examinar si el objeto de la solicitud se ajusta a lo establecido en los artículos 4, 8 a 10 de esta Ley.

En caso de haberse presentado oposiciones, la Superintendencia de Industria y Comercio se pronunciará sobre éstas y sobre la concesión o denegatoria del registro solicitado mediante resolución.

La decisión de concesión o denegación será debidamente motivada, y podrá conceder parcialmente las reivindicaciones hechas por el sujeto de derecho solicitante.

Artículo 26. La decisión de concesión o denegación de la protección es susceptible del recurso de apelación en sede administrativa.

Capítulo VIII. Del registro de actos posteriores a la radicación de la solicitud.

Artículo 27. Los derechos que confiere el registro de protección para el algoritmo o combinación de algoritmos implementados en un programa de ordenador, lo mismo que la mera expectativa de concesión mientras se encuentre en trámite, podrán ser objeto de cualquier tipo de acto jurídico.

Artículo 28. Todo acto que implique transferencia, renuncia, limitación o gravamen al derecho conferido, deberá registrarse ante la Superintendencia de Industria y Comercio para el perfeccionamiento del respectivo acto, mediante la radicación del documento escrito en el cual conste la manifestación de voluntad del titular, con nota de presentación personal o debidamente apostillado.

Los demás actos jurídicos podrán registrarse ante la Superintendencia de Industria y Comercio para fines de oponibilidad, y no exigirán que el documento en el que conste la manifestación de voluntad del titular contenga nota de presentación personal o apostilla.

Artículo 29. La Superintendencia de Industria y Comercio se abstendrá de registrar los actos jurídicos que no se ajusten a las disposiciones del Régimen Común de Tratamiento a los Capitales Extranjeros y sobre Marcas, Patentes, Licencias y Regalías, o que no se ajusten a las disposiciones comunitarias o nacionales sobre prácticas comerciales restrictivas de la libre competencia.

Artículo 30. Los cambios en la información suministrada para la obtención del registro deberán registrarse ante la Superintendencia de Industria y Comercio.

Cualquier notificación realizada conforme a los datos que figuren en el registro, se reputará válida.

Artículo 31. La Superintendencia de Industria y Comercio resolverá las peticiones de aclaración, corrección o adición al registro que no impliquen alteración de la titularidad del derecho, lo limiten o amplíen en cualquier forma.

Serán aplicables en lo pertinente, lo dispuesto en el artículo 18 de esta Ley.

Artículo 32. La Superintendencia de Industria y Comercio establecerá el cobro de tasas para todos los actos referidos al trámite de la solicitud y los posteriores a su radicación.

Título III. Efectos de la protección conferida

Capítulo I. De los titulares del derecho

Artículo 33. Los derechos conferidos sobre el algoritmo o combinación de algoritmos implementados para un programa de ordenador corresponderán al sujeto de derecho que haya obtenido su registro.

Artículo 34. Salvo estipulación en contrario, pertenecerá al empleador o contratante el código fuente del programa de ordenador cuyo algoritmo o algoritmos hayan sido registrados conforme se indica en esta Ley.

La misma regla se aplica cuando el trabajador o contratista no haya sido contratado para la elaboración de un programa de ordenador o el diseño de un algoritmo o combinación de algoritmos para empleo en un programa de ordenador, si el programa de ordenador, el algoritmo o combinación de algoritmos se elabora mediante datos, medios conocidos o utilizados debido al trabajo o servicio contratado.

En este último caso el trabajador o contratista tendrá derecho a una compensación que se fijará de acuerdo con el monto del salario, la utilidad que reporte la explotación del programa de

ordenador al empleador o contratante u otros factores similares. A falta de acuerdo entre las partes, el juez fijará el monto de la compensación en equidad.

Artículo 35. Si varios sujetos de derecho implementaran el algoritmo o combinación de algoritmos en un programa de ordenador, el derecho sobre el algoritmo pertenecerá en común a todos ellos.

Artículo 36. Si varios sujetos de derecho de forma independiente los unos de los otros implementaran el mismo algoritmo o combinación de algoritmos, en un programa de ordenador cuyas funcionalidades son sustancialmente las mismas, el derecho se concederá a aquél o aquellos sujetos de derechos, incluidos sus causahabientes, que primero hubieren presentado la solicitud de protección correspondiente.

En este caso, no se considerará una infracción de derechos la explotación que hayan hecho aquellos que no hubieran solicitado el registro de la protección, hasta la fecha de concesión del derecho conforme a esta Ley.

Capítulo II. De los derechos que se confieren

Artículo 37. Los derechos sobre el algoritmo o combinación de algoritmos implementados en un programa de ordenador conforme a esta Ley se adquirirán por su registro ante la Superintendencia de Industria y Comercio.

Artículo 38. El registro de la protección confiere a su titular una protección sobre el algoritmo o combinación de algoritmos implementados para un programa de ordenador, por un plazo de duración de diez (10) años contados a partir de la fecha de concesión del derecho.

Artículo 39. El alcance de la protección conferida estará determinado por el tenor de las reivindicaciones. La descripción y los dibujos servirán para interpretarlas.

Artículo 40. Durante los primeros cinco (5) años de protección, contados desde la fecha de concesión del derecho, el titular del derecho tendrá el derecho exclusivo de impedir a terceras personas que no tengan su consentimiento, realizar cualquiera de los siguientes actos:

1. Divulgar el código fuente del programa de ordenador, o facilitar su divulgación a cualquier persona, antes de que sea accesible al público con el consentimiento del propietario;
2. Permitir o facilitar el acceso de cualquier persona a cualquier objeto que almacene o reproduzca el programa de ordenador, antes de que el programa de ordenador se haga accesible al público con el consentimiento del propietario;
3. Utilizar el algoritmo o la combinación de algoritmos para ser implementados en un programa de ordenador cuyas funcionalidades sean sustancialmente similares a aquellas del programa de ordenador asociado al registro;

4. Utilizar la descripción de la solicitud de protección para la elaboración de un programa de ordenador cuyas funcionalidades sean sustancialmente similares a aquellas del programa de ordenador asociado al registro;
5. Incorporar a un soporte físico o producto el algoritmo o la combinación de algoritmos junto a un programa de ordenador cuyas funcionalidades sean sustancialmente similares a aquellas del programa de ordenador asociado al registro;
6. Celebrar actos jurídicos de puesta a disposición, reproducción, transferencia, licenciamiento, distribución, exportación, importación; gratuitos, oneroso o aleatorios; que tengan por objeto el algoritmos o combinación de algoritmo protegido.

Artículo 41. Toda persona que con motivo de su trabajo, empleo, cargo, puesto, contrato, desempeño de su profesión o relación de negocios, tenga acceso al algoritmo o combinación de algoritmo implementado para un programa de ordenador sobre cuya confidencialidad se le haya prevenido, deberá abstenerse de usarlo, comunicarlo, divulgarlo, o revelarlo, sin causa justificada y sin consentimiento del sujeto o sujetos de derecho que hayan cubierto los costos y gastos para su producción.

Artículo 42. Durante los primeros cinco (5) años de protección, contados desde la fecha de concesión del derecho, el titular del derecho tendrá igualmente derecho a despedir con justa causa al trabajador, o terminar unilateralmente el contrato de prestación de servicio al contratista, que incurra en alguna de las siguientes conductas:

1. Divulgue, facilite su divulgación, permita o facilite el acceso al código fuente del programa de ordenador antes de que el programa de ordenador se haga accesible al público con el consentimiento del sujeto de derecho solicitante o titular;

2. Divulgue, facilite su divulgación, permita o facilite el acceso a información de la cual haya tenido conocimiento por razón de su actividad personal al servicio del sujeto de derecho solicitante, referida a la confección, arquitectura, diseño o funcionalidades del programa de ordenador antes de que el programa de ordenador se haga accesible al público con el consentimiento del sujeto de derecho solicitante o titular;

Para el ejercicio de las facultades concedidas al titular conforme a este artículo, deberá: (i) Demostrar haber tomado medidas razonables para mantener secreto el código fuente del programa de ordenador, y (ii) Haber permitido al trabajador o contratista defenderse de los cargos que se le hubieren imputado.

Artículo 43. Durante los últimos cinco (5) años de protección, contados desde la fecha de concesión del derecho, el titular del derecho concedido únicamente podrá ser remunerado por la explotación que se haga del algoritmo o combinación de algoritmos en programas de ordenador cuyas funcionalidades sean sustancialmente similares a las funcionalidades del programa de ordenador asociado al registro.

El titular tendrá derecho a ser remunerado según este artículo aunque la explotación se haga a título gratuito, salvo que medie renuncia particular por parte del titular, o que el acto de agotamiento del derecho haya sido a título gratuito.

Artículo 44. El titular, o sus causahabientes, tendrá derecho a ser remunerado por lo actos de explotación que hayan celebrados quienes no obtuvieron la protección conforme el artículo 36 de esta Ley, a partir de la fecha de concesión del derecho.

Artículo 45. El registro obtenido según la presente Ley automáticamente calificará al proyecto como de investigación, desarrollo tecnológico o innovación a fin de acceder a las deducciones, descuentos y créditos según los artículos 158-1, 256 y 256-1 del Estatuto Tributario.

El Gobierno Nacional reglamentará la materia, para que las empresas beneficiadas con el registro puedan deducir las inversiones realizadas en la innovación, desarrollo tecnológico e investigación y desarrollo experimental del algoritmo o algoritmos implementados en programas de ordenador que hayan obtenido el registro según esta Ley, teniendo en cuenta lo siguiente:

1. Las deducciones no deben limitarse a las inversiones en capital o gastos de nómina realizados durante el año en que se hubiere obtenido el registro,
2. Las deducciones no deben referirse necesariamente a la depreciación de la propiedad, planta y equipo del titular de derecho,

3. Los titulares del derecho podrán obtener un punto porcentual adicional (+1%) a la deducción aplicable, si:

(i) El ochenta por ciento (80%) del personal dedicado a la innovación, desarrollo tecnológico e investigación y desarrollo experimental del algoritmo o algoritmos implementados en programas de ordenador registrado según esta Ley fue vinculado laboralmente al titular del derecho.

(ii) Se acredita el pago completo de los aportes parafiscales asociados al personal indicado en el *ítem* (i) anterior.

Parágrafo.- Nada de lo dispuesto en este artículo modifica o deroga las funciones asignadas por el Decreto 121 de 2014 al Consejo Nacional de Beneficios Tributarios en Ciencia, Tecnología e Innovación (CNBT); al cual le corresponderá certificar lo señalado en el numeral 3. de este artículo.

Capítulo III. De las limitaciones al derecho concedido

Artículo 46. El registro de un algoritmo o combinación de algoritmos implementados para un programa de ordenador no dará el derecho de impedir a un tercero realizar actos de comercio respecto del algoritmo o combinación de algoritmos, después de que este o estos hagan parte de un programa de ordenador puesto a disposición en el territorio nacional por acto jurídico, gratuito u oneroso, con el consentimiento del titular.

Para los efectos de este artículo, no se entenderá que el algoritmo o combinación de algoritmos se han puesto a disposición con el consentimiento del titular en los supuestos de los artículos 6, 19, 56 y 57 de esta Ley.

Artículo 47. El titular no podrá ejercer los derechos conferidos en los artículos 40 y 43 de esta Ley, respecto de lo siguiente:

1. Actos realizados en el ámbito privado, que no impliquen la puesta a disposición del algoritmo o combinación de algoritmos implementados en un programa de ordenador.
2. Actos realizados exclusivamente con fines de enseñanza o de investigación científica o académica.

Artículo 48. Si el algoritmo o combinación de algoritmos implementado para programa de ordenador se encuentra incorporado en un soporte físico incluido dentro de los componentes esenciales para la navegación y seguridad de naves y aeronaves, el titular únicamente tendrá el derecho de remuneración indicado en el artículo 43 de esta Ley.

El plazo del derecho de remuneración según este párrafo es de diez (10) años, desde el momento de la concesión del derecho al titular.

Artículo 49. El Gobierno Nacional, a través de la Superintendencia de Industria y Comercio, convocará cada tres años a una audiencia pública con el fin de realizar una revisión periódica de

las limitaciones y excepciones a los derechos concedidos en virtud de esta Ley, con el objetivo de determinar la necesidad y conveniencia de presentar ante el Congreso de la República un proyecto de ley que reforme, elimine o adicione limitaciones y excepciones.

Dicho proyecto deberá observar las reglas establecidas en los tratados internacionales ratificados por Colombia sobre protección a programas de ordenador, lo mismo que los derechos adquiridos por los titulares conforme la legislación precedente.

Título IV. Acciones judiciales, medidas cautelares y de frontera

Capítulo I. De la nulidad de la protección concedida

Artículo 50. Los vicios de los actos administrativos que sean dictados por la Superintendencia de Industria y Comercio podrán ser objeto de la acción de nulidad absoluta indicada en el artículo 172 de la Decisión 486 de la Comisión de la Comunidad Andina de Naciones, cuando el cargo contra el acto administrativo corresponda a violación de los artículos 4, 8 y 9-1 de esta Ley.

Artículo 51. Los vicios de los actos administrativos que sean dictados por la Superintendencia de Industria y Comercio podrán ser objeto de la acción de nulidad relativa indicada en el artículo 172 de la Decisión 486 de la Comisión de la Comunidad Andina de Naciones, cuando el cargo contra el acto administrativo corresponda a violación de los numerales 2, 3 y 4 del artículo 9 de esta Ley.

Artículo 52. En los demás casos, los vicios de los actos administrativos que sean dictados por la Superintendencia de Industria y Comercio podrán ser objeto de la acción contemplada en el artículo 138 de la Ley 1437 de 2011.

Capítulo II. De la acción reivindicatoria

Artículo 53. Cuando la protección se hubiese solicitado u obtenido por sujeto sin derecho a obtenerlo, o en perjuicio de sujeto que también tuviese tal derecho, el sujeto afectado podrá reivindicarlo ante la jurisdicción pidiendo que le sea transferida la solicitud en trámite o el derecho concedido, o que se le reconozca como co-solicitante o cotitular del derecho.

En la misma acción de reivindicación podrá demandarse la indemnización de daños y perjuicios.

Artículo 54. Esta acción se podrá interponer a prevención, ante la jurisdicción ordinaria o ante la Superintendencia de Industria y Comercio.

Artículo 55. Esta acción prescribe a los dos (2) años contados desde la fecha de concesión del derecho. No prescribirá la acción si quien obtuvo el derecho lo hubiese solicitado de mala fe.

Capítulo III. De las acciones por infracción de derechos

Artículo 56. El titular del derecho podrá entablar acción a prevención, ante la jurisdicción ordinaria o ante la Superintendencia de Industria y Comercio, contra cualquier sujeto de derecho que infrinja su derecho.

En caso de cotitularidad de un derecho, cualquiera de los cotitulares podrá entablar la acción contra una infracción, sin que sea necesario el consentimiento de los demás, salvo acuerdo en contrario entre los cotitulares.

Artículo 57. Quien tenga legítimo interés podrá presentar acción judicial contra el titular de un derecho concedido conforme a esta Ley, para que se declare que no es infractor del derecho sobre el algoritmo o combinación de algoritmos que explota el accionante en un determinado programa de ordenador.

Para interponer la presente acción, el accionante deberá acreditar la presentación de una solicitud de protección a su favor, conforme el capítulo I Título II de esta Ley.

Artículo 58. En los casos en los que se alegue una infracción al derecho concedido en esta Ley, el demandado no podrá excepcionar lo concerniente al artículo 13 a) de la Ley 1915.

Adicionalmente, corresponderá al demandado probar que el algoritmo o combinación de algoritmo implementado en el programa de ordenador correspondiente es:

1. Distinto al reivindicado en la solicitud concedida;

O

2. Fue implementado para un programa de ordenador cuyas funcionalidades son sustancialmente diferentes a las funcionalidades del programa de ordenador asociado al registro.

Artículo 59. Se presume, salvo prueba en contrario, que un programa de ordenador implementa el algoritmo o combinación de algoritmos protegido, si:

1. El demandado ha contratado personal que hubiere estado vinculado al titular, mediante contrato laboral o de prestación de servicios, dentro de los cinco (5) años anteriores a la fecha de presentación de la demanda;

2. El demandado es socio, representante legal o consultor de una empresa en la cual se elaboran programas de ordenador o productos informáticos;

3. La demandada es una sociedad de la cual es socio algún personal que hubiere estado vinculado al titular, mediante contrato laboral o de prestación de servicios, dentro de los cinco (5) años anteriores a la fecha de presentación de la demanda;

Artículo 60. El titular podrá demandar la indemnización de perjuicios derivado de la infracción de derechos, incluyendo los causados durante el período comprendido entre la fecha de radicación de la solicitud y la fecha de concesión del derecho; salvo por el supuesto de hecho señalado en el segundo inciso del artículo 36 de esta Ley.

Artículo 61. Para efectos de calcular la indemnización de daños y perjuicios se tomará en cuenta, entre otros, los criterios siguientes:

1. El daño emergente y el lucro cesante sufrido por el titular del derecho como consecuencia de la infracción;
2. El monto de los beneficios obtenidos por el infractor como resultado de los actos de infracción; o,
3. El precio que el infractor habría pagado por concepto de una licencia contractual, teniendo en cuenta el valor comercial del derecho infringido y las licencias contractuales que ya se hubieran concedido.
4. El valor agregado específico que el algoritmo protegido ha aportado al aparato en el cual se ha implementado.

Artículo 62. El titular del derecho podrá optar por la indemnización preestablecida en este artículo, en cuyo caso no tendrá que hacer juramento estimatorio ni probar la cuantía de los daños y perjuicios causados por la infracción, así:

1. En aquellos casos en los cuales la explotación del algoritmo o combinación de algoritmos implementado para programa de ordenador estuviere por fuera del acto consentido por el titular, dicha indemnización será equivalente a un mínimo de diez (10) salarios mínimos legales

mensuales vigentes y hasta un máximo de cien (100) salarios mínimos legales mensuales vigentes;

2. En aquellos casos en los cuales no se desvirtúe alguna de las presunciones contenidas en el artículo 59 de esta Ley, dicha indemnización será equivalente a un mínimo de cincuenta (50) salarios mínimos legales mensuales vigentes y hasta un máximo de quinientos (500) salarios mínimos legales mensuales vigentes.

Parágrafo. Para cada caso particular el juez ponderará y declarará en la sentencia que ponga fin al proceso el monto de la indemnización teniendo en cuenta las pruebas que obren en el proceso, entre otras la duración de la infracción, su amplitud, la cantidad de productos y/o usos infractores.

Artículo 63. Para aquellos casos en los cuales el titular del derecho exija su derecho de remuneración, podrá optar por un sistema de remuneración preestablecida, en cuyo caso no tendrá que hacer juramento estimatorio, así:

1. En aquellos casos en los cuales la explotación del algoritmo o combinación de algoritmo implementado para programa de ordenador lo hubiere realizado una empresa en la cual se elaboran programas de ordenador o productos informáticos, y tal explotación ocurre únicamente en el mercado nacional, dicha remuneración será equivalente a un mínimo de un mínimo de diez (10) salarios mínimos legales mensuales vigentes y hasta un máximo de mil (1000) salarios mínimos legales mensuales vigentes;

2. En aquellos casos en los cuales la explotación del algoritmo o combinación de algoritmo implementado para programa de ordenador lo hubiere realizado una empresa en la cual se elaboran programas de ordenador o productos informáticos, y tal explotación ocurre en el mercado mundial, dicha remuneración será equivalente a un mínimo de un mínimo de mil (1000) salarios mínimos legales mensuales vigentes y hasta un máximo de diez mil (10000) salarios mínimos legales mensuales vigentes

3. En los demás casos, dicha remuneración será equivalente a un mínimo de un (1) salario mínimo legal mensual vigente y hasta un máximo de cuarenta (40) salarios mínimos legales mensuales.

El derecho de remuneración obtenido mediante este sistema corresponderá a un único pago durante el lapso en el cual se encuentre vigente el derecho de remuneración.

Parágrafo.- Para cada caso particular el juez ponderará y declarará en la sentencia que ponga fin al proceso el monto de la remuneración teniendo en cuenta las pruebas que obren en el proceso, entre otras su amplitud, la cantidad de productos y/o usos.

Artículo 64. La acción del artículo 56 de esta Ley prescribirá a los dos (2) años contados desde la fecha en que el titular tuvo conocimiento de la infracción o, en todo caso, a los cinco (5) años contados desde que se cometió la infracción por última vez.

Artículo 65. La acción del artículo 57 de esta Ley prescribirá a los dos (2) años contados desde la fecha en que el accionante tuvo conocimiento de la solicitud de protección presentada por el titular o, en todo caso, a los cinco (5) años contados desde que se concedió el derecho al titular.

Capítulo IV. Del tipo penal

Artículo 66. El artículo 307 del Código Penal – Ley 599 de 2000 – quedará así:

“El que fabrique producto sin autorización de quien tiene el derecho protegido legalmente, o use sin la debida autorización medio o proceso patentado, o implemente algoritmo o combinación de algoritmo en un programa de ordenador sin la debida autorización del titular del derecho, incurrirá en prisión de dieciséis (16) a setenta y dos (72) meses y multa de veintiséis punto sesenta y seis (26.66) a mil quinientos (1.500) salarios mínimos legales mensuales vigentes.

En la misma pena incurrirá el que introduzca al país o saque de él, exponga, ofrezca en venta, enajene, financie, distribuya, suministre, almacene, transporte o adquiera con fines comerciales o de intermediación producto fabricado con violación de patente, soporte físico o producto con violación al derecho concedido sobre el algoritmo o combinación de algoritmo implementado en un programa de ordenador”.

Capítulo VI. Medidas cautelares

Artículo 67. El sujeto de derecho que inicie o vaya a iniciar la acción indicada en el artículo 55 de esta Ley, podrá pedir a la autoridad judicial competente que ordene medidas cautelares inmediatas conforme el artículo 590 de la Ley 1564 de 2012.

Las medidas cautelares podrán pedirse antes de presentar la demanda, juntamente con la demanda o la denuncia, o con posterioridad a su inicio.

Parágrafo.- La solicitud de medidas cautelares podrá solicitarla el interesado a prevención, ante los jueces ordinarios o ante la Superintendencia de Industria y Comercio.

Artículo 68. El sujeto de derecho que inicie o vaya a iniciar la acción indicada en el artículo 56 de esta Ley, antes de interponerla podrá únicamente pedir a la autoridad judicial competente que ordene medidas cautelares innominadas conforme el artículo 590-1-c) de la Ley 1564 de 2012.

Artículo 69. Cuando se hubiera ejecutado una medida cautelar sin intervención del sujeto de derecho afectado, el decreto de medidas cautelares se le notificará al momento de ejecutadas estas, y el término de ejecutoria de dicha providencia correrá para el afectado a partir del día siguiente a la fecha en la cual se hubieren ejecutado las medidas.

El sujeto de derecho afectado podrá interponer el recurso que corresponda, o los recursos que correspondan, en el término previsto en la norma procesal aplicable.

Artículo 70. Cuando como consecuencia de la interposición de recursos se revoque el decreto de medidas cautelares, el sujeto de derecho afectado podrá iniciar, privativamente ante la jurisdicción ordinaria, un incidente de regulación de perjuicios y ejecutar la caución que se hubiere prestado, conforme a las normas procesales aplicables.

Para el efecto, las actuaciones que se hubieren adelantado ante la Superintendencia de Industria y Comercio deberán remitirse al juez competente que hubiere desplazado. Si las medidas cautelares se hubieren adelantado ante un juez ordinario, las actuaciones indicadas en el inciso anterior deberán adelantarse ante el mismo juez que decretó las medidas cautelares, prorrogando su competencia si fuere el caso.

Artículo 71. Toda medida cautelar decretada sin intervención del sujeto de derecho afectado quedará sin efecto automáticamente, si la acción correspondiente no se presentare dentro de los diez (10) días siguientes contados desde la ejecución de la medida.

Parágrafo.- La acción correspondiente podrá interponerse ante una autoridad judicial competente diferente de aquella que la decretó.

Artículo 72. El demandante podrá solicitar que se ordenen, entre otras, las medidas cautelares indicadas en los artículos 568 del Decreto 410 de 1971, 246 de la Decisión 486 de la Comisión de la Comunidad Andina de Naciones, una o más de las siguientes medidas:

1. El cese de los actos que constituyen la infracción, contra el presunto infractor o usuario final del programa de ordenador correspondiente;

2. El retiro de los circuitos comerciales físicos o digitales, de los programas de ordenador, soportes físicos o productos que incluyan el algoritmo o combinación de algoritmos;

3. La prohibición de importación o la exportación de los soportes físicos o productos que incluyan el algoritmo o combinación de algoritmos;

4. La adjudicación en propiedad de los soportes físicos o productos que incluyan el algoritmo o combinación de algoritmos, en cuyo caso el valor de los bienes se imputará al importe de la indemnización de daños y perjuicios;

5. La adopción de las medidas necesarias para evitar la continuación o la repetición de la infracción, incluyendo:

1. La destrucción de los soportes físicos o productos que incluyan el algoritmo o combinación de algoritmos;

2. La eliminación de todo soporte lógico en el cual se encuentren reproducidos los algoritmos, combinación de algoritmos o el programa de ordenador correspondiente, sea en código fuente u objeto;

6. Ordenar al demandado o denunciado informar al titular del derecho sobre la identidad de los terceros que hayan participado en la producción y distribución de soportes físicos o productos infractores, y sobre sus circuitos de distribución.

Artículo 73. Las medidas cautelares se aplicarán sobre los productos resultantes de la presunta infracción y sobre los materiales o medios que sirvieran principalmente para cometerla.

Capítulo VII. Medidas en frontera

Artículo 74. El titular del derecho que tuviera motivos fundados para suponer que se va a realizar la importación o la exportación de productos que infringen sus derechos, podrá solicitar a la autoridad competente suspender esa operación aduanera, conforme el decreto 4540 de 2006.

Artículo 75. El artículo 1 del Decreto 4540 de 2006 quedará así:

“Artículo 1º. Definiciones. Las siguientes expresiones tendrán el significado que a continuación se indica:

*- **Autoridad competente:** Es la autoridad judicial con competencia en materia de Propiedad Intelectual, quien resolverá sobre el fondo del asunto que originó la suspensión de la operación aduanera.*

- **Derecho de autor:** Forma de protección jurídica, en virtud de la cual se le otorga al creador de una obra literaria o artística, un conjunto de prerrogativas de orden moral y patrimonial, que le permiten autorizar o prohibir su utilización de cualquier manera o por cualquier medio conocido o por conocer.

- **Derechos conexos:** Conjunto de derechos reconocidos a los artistas intérpretes o ejecutantes, productores de fonogramas y organismos de radiodifusión, en relación con sus interpretaciones o ejecuciones, fonogramas y emisiones de radiodifusión, respectivamente.

- **Marca:** Es cualquier signo capaz de distinguir los bienes o servicios de una empresa de los de otras empresas, siempre que sea susceptible de representación gráfica.

- **Mercancía pirata:** Cualquier copia hecha sin el consentimiento del titular del derecho de autor o derechos conexos, o de una persona debidamente autorizada por él en el país de producción y que se realicen directa o indirectamente a partir de una obra o producción protegida cuando la realización de esas copias habría constituido una infracción al derecho de autor o de un derecho conexo.

- **Mercancía de marca falsa:** Cualquier mercancía, incluido su embalaje, que lleve puesta sin autorización una marca de fábrica o de comercio idéntica a la marca válidamente registrada para tal mercancía, o que no pueda distinguirse en sus aspectos esenciales de esa marca, o sea confusamente similar, y que de ese modo lesione los derechos del titular de la marca de que se trate.

- **Titular del derecho de autor o del derecho conexo:** Es la persona natural o jurídica que, en condición de titular originario o derivado, se encuentra facultada para autorizar o prohibir todo acto de explotación o utilización por cualquier medio de una obra literaria o artística, de una interpretación artística, de una emisión de radiodifusión o de un fonograma.

- **Titular de marca:** es el propietario de un registro marcario y en su caso, sus causahabientes, cesionarios y licenciatarios exclusivos.

- **Titular de un algo timo o combinación de algoritmos:** es el sujeto de derecho que se encuentra facultado para impedir la operación aduanera de soportes físicos que incluyan un programa de ordenador vinculado a un algoritmo o combinación de algoritmos protegidos, tales como: Aparatos eléctricos o electrónicos, discos compactos, tarjetas de memoria, entre otros.

Para los efectos de este Decreto, el titular de un algoritmo o combinación de algoritmos protegidos será asimilado al titular de marca, y la mercancía infractora a la mercancía de marca falsa”.

Título V. Disposiciones finales, entrada en vigor y derogatorias

Artículo 76. La Superintendencia de Industria y Comercio queda facultada para la determinación de las tasas aplicables al trámite de concesión del derecho, desde su radicación hasta el registro de actos posteriores a la radicación de la solicitud.

Previa consulta pública a los interesados por un término máximo de seis (6) meses y mínimo cada cinco (5) años, a partir de la puesta en funcionamiento del registro de algoritmos implementados para programas de ordenador, señalará mediante acto administrativo de carácter general, los criterios para definir los programas de ordenador a los que se aplicará el registro acá contemplado, la clasificación de los algoritmos o combinación de algoritmos implementados para programas de ordenador y las funcionalidades de programas de ordenador a las que se asocian éstos.

Parágrafo. - La Superintendencia de Industria y Comercio dejará de cumplir con lo indicado en el Segundo inciso de este artículo, a partir del momento en que entre en vigor un Tratado internacional que señale la clasificación allí indicada, y del cual haga parte la República de Colombia.

Artículo 77. La Superintendencia de Industria y Comercio tiene un plazo improrrogable de un (1) año, contado a partir de la apropiación de recursos del presupuesto nacional conforme el artículo siguiente, para la estructuración de la puesta en funcionamiento del sistema de registro, el recaudo de un repositorio inicial de algoritmos o combinación de algoritmos de uso corriente

en el mercado nacional, la determinación de las tasas aplicables para adelantar todo trámite relacionado con el derecho que se instituye mediante la presente Ley.

Artículo 78. El Gobierno Nacional deberá presentar al congreso la partida presupuestal correspondiente a la puesta en funcionamiento del sistema de protección acá señalado, en el siguiente proyecto de ley de presupuesto general de la Nación.

Artículo 79. La presente Ley entra a regir cumplido el plazo indicado en el artículo 77 de esta Ley.

Tabla de contenido

Análisis económico de la protección al <i>software</i> : Hacia un nuevo modelo de protección	1
Capítulo I. Mejor malo conocido, que bueno por conocer: Pros y contras de la protección del <i>software</i> vía el Derecho de Autor	8
1. ¿Qué incentiva la protección del <i>software</i> a través del Derecho de Autor?	14
1.1. La creación cumulativa del <i>software</i> entre innovadores.	15
1.2. La protección jurídica que reciben los elementos gráficos del <i>software</i>	17
1.3. El conjunto de Tratados internacionales en materia de Derechos de Autor.	20
2. Algunas externalidades negativas aparejadas a la protección jurídica del <i>software</i> mediante el Derecho de Autor	21
2.1. Dicotomía entre idea y expresión de la idea/Test de Abstracción-Filtración-Comparación.	23
2.2. Alcance de los derechos morales y patrimoniales concedidos al <i>software</i>	26
2.3. Efecto económico de la ingeniería reversa.	30
2.4. Efecto político, económico y jurídico del <i>cloning</i>	33
2.5. Efecto económico de la <i>estandarización</i> del <i>software</i>	37
Capítulo II. Reaccionando al déficit de protección para el <i>software</i> : Mitos y realidades sobre las patentes de <i>software</i>	41
1. ¿Por qué hay patentes de <i>software</i> ?	44
1.1. Beneficios generales de conceder patentes de <i>software</i>	47
1.2. Beneficios particulares de conceder patentes de <i>software</i>	49
2. Crítica al otorgamiento de patentes de <i>software</i>	51
2.1. Reservas sobre la “materia patentable”, novedad, nivel inventivo y aplicación industrial del <i>software</i>	53
2.2. Las patentes de <i>software</i> no generan beneficios transversales para la industria del <i>software</i>	58
2.3. El problema de las <i>patent trolls</i> es la interoperabilidad y los costos de reentrenamiento.	61
Capítulo III. Feudalismo informático: Esto pasa cuando se juntan el <i>software</i> y los secretos empresariales	64
1. Ventajas de la protección del <i>software</i> mediante secretos empresariales	68
1.1. Modelo de protección automático y ampliado.	69
1.2. Menor propensión hacia el litigio.	70
2. Desventajas de la protección del <i>software</i> mediante la Protección a Secretos Empresariales	72
2.1. Ingeniería reversa, ¿para replicar productos?... ..	73
2.2. Costosa protección del secreto.	77
2.3. Fuga de talentos.	79
Capítulo IV. Hagamos lo siguiente: Marco normativo para un nuevo modelo de protección.....	82
1. ¿Por qué no despegar una protección <i>sui generis</i> para el <i>software</i> ?	86
1.1. Las leyes especiales de protección para el <i>software</i> y el <i>cloning</i>	88
1.2. La realidad del <i>software</i> de código abierto.	91
2. Características de un modelo <i>sui generis</i> de protección para el <i>software</i>	94
2.1. Protejamos los algoritmos que componen el <i>software</i> , para usos reivindicados y segundos usos innovadores.	100

2.2. Concedamos un derecho de exclusiva, que se transforme en un derecho de remuneración al cabo de un tiempo y esté sujeto a un agotamiento local del derecho.	105
2.3. Dejemos los derechos morales para las obras protegidas por el Derecho de Autor.	115
3. ¿Quiénes aprovecharían un nuevo modelo de protección para el <i>software</i> ?.....	118
Capítulo V. Texto del nuevo modelo de protección jurídica del <i>software</i>	124
Tabla de contenido.....	163
Bibliografía.....	165

Bibliografía

- Adler, E. (1986). Ideological "guerrillas" and the quest for technological autonomy: Brazil's domestic computer industry. *International Organization*, 40(3), 673-705.
- Aguirre Soriano, D. A. (2014). Criterios del análisis económico del Derecho aplicables al estudio de patentabilidad de programas de computador. *Con-Texto*, 41, 87-114.
- Alice Corp. v. CLS Bank International, 573 U.S. 208 (Supreme Court of The United States 19 de Jun de 2014).
- Allen, M., Armstrong, D., Reid, M., & Riemenschneider, C. (2009). IT Employee Retention: Employee Expectations and Workplace Environments. *SIGMIS CPR '09* (págs. 95 - 100). New York: ACM.
- Anderlini, L., Felli, L., Immordino, G., & Riboni, A. (2013). Legal institutions, innovation and growth. *International Economic Review*, 54(3), 937 - 956.
- (2018). *Aproximación al impacto de las TIC en la desigualdad de ingresos en Colombia*. Bogotá D.C.: DNP.
- Arenas Correa, J. D. (2013). *Estrategias de autorregulación en bienes intangibles: El caso del software*. Medellín: Universidad de Antioquia.
- As Colombia Ltda. v. Informática y Gestión S.A (Tribunal Arbitral agosto de 2006).
- Ascarelli, T. (1970). *Teoría de la concurrencia y de los bienes inmateriales*. Barcelona: Bosch.
- Athreye, S. S. (2007). The Indian software industry. En A. Ashish, & A. Gambardella (Edits.), *From underdogs to tigers: The rise and growth of the software industry in Brazil, China, India, Ireland, and Israel* (págs. 1-55). New York: Oxford Scholarship Online.
- (2019). *Bases del plan nacional de desarrollo 2018 - 2022*. Bogotá D.C.: DNP.
- Baylos Corroza, H. (2009). *Tratado de derecho industrial*. Cizur Menor: Civitas Thomson Reuters.
- Berkemeyer, H. (2009). Estudio de derecho y propiedad intelectual. Homenaje a Arturo Alessandri Besa. En R. Velasco Santelices, & M. Morales Andrade (Edits.), *Estudios de derecho y propiedad intelectual* (págs. 123-138). Jurídica de Chile.
- Bessen, J. (3 de Sep de 2013). The patent troll crisis is really a software patent crisis. *Washington Post*.
- Bessen, J., & Hunt, R. M. (2004). *The software patent experiment*. Philadelphia: Federal Reserve Bank of Philadelphia.
- Bilski v. Kappos, 561 U.S. 593 (Supreme Court of The United States 28 de Jun de 2010).
- Bird, R. C., & Chaudhry, P. E. (2010). Pharmaceuticals and the European Union: Managing Gray Markets in an Uncertain Legal Environment. *Virginia Journal of International Law*, 50, 719 - 756.
- Blind Knut, J. E., & Firedewald, M. (2005). *Software patents. Economics impacts and policy implications*. Northampton, Massachusetts, United States of America: Edward Elgar Publishing Inc.
- Botana Agra, M. J. (2001). En torno a la patentabilidad del software y de los métodos de negocios aplicables en el comercio electrónico. En J. A. Gomez Segade (Ed.), *Comercio electrónico en internet* (págs. 157-184). Madrid: Marcial Pons.

- Bouza López, M. A. (1997). Idea y expresión en los programas de ordenador (Comentario a la sentencia Lotus c. Borland del primer circuito de los Estados Unidos de 9 de marzo de 1995). En *Actas de derecho industrial y de derecho de autor* (Vol. 18, págs. 325-346). Madrid: Marcial Pons.
- Buitrago Díaz, E. (2012). Tributación internacional de los derechos de propiedad intelectual. En R. Metke Méndez, E. I. León Robayo, & E. Varela Pezzano, *Propiedad intelectual. Reflexiones* (págs. 453 - 478). Bogotá D.C.: Universidad del Rosario.
- Buitrago Restrepo, F., & Duque Márquez, I. (2013). *La economía naranja. Una oportunidad infinita*. Washington: BID.
- Buitrago, E. (2007). *El concepto de cánones y/o regalías en los convenios para evitar la doble tributación sobre la renta*. Valencia: Wolters Kluwer.
- Buitrago, E. (2019). Patent boxes and the erosion of trust in trade and in governance. *International journal of public law and policy*, in print.
- Burtis, M. M., & Kobayashi, B. H. (2001). Why an original can be better than a copy: Intellectual property, the antitrust refusal to deal, and ISO antitrust. *Supreme Court Economic Review*, 143-170.
- Byungkwon, L. (1989). Protection of computer programs under the computer program protection law of the Republic of Korea. *Harvard International Law Journal*, 30(1), 171-194.
- Calabresi, G. (1968). Transaction Costs, Resource Allocation and Liability Rules-A Comment. *The Journal of Law & Economics*, 11(1), 67 - 73.
- Calabresi, G. (1991). The Pointlessness of Pareto: Carrying Coase Further. *The Yale Law Journal*, 100(5), 1211 - 1237.
- Calabresi, G. (2009). *A Common Law for the Age of Statutes*. Cambridge: Harvard University Press .
- Calle D'Alleman, S. B. (2012). *Protección jurídica del software. Crítica a su actual regulación y redefinición de su naturaleza frente al derecho*. Bogotá D.C.: Ibañez.
- Camargo, D. (2008). Límites al uso de la patente, desde el punto de vista del Derecho de la competencia, propósito del caso Microsoft. *Con-Texto*, 24, 119-142.
- (2016). *Caracterización del sector de teleinformática, software y TI en Colombia. Año 2015*. . Fedesoft, Bogotá D.C.
- Carranza Torres, M. (2008). *El derecho de la innovación tecnológica. Una historia de tecnotropismo capitalista*. Buenos Aires: Abeledo Perrot.
- Carranza Torres, M. (2004). *Problemática jurídica del software libre*. Buenos Aires: Lexis Nexis.
- Castán Pérez-Gómez, A. (1999). Perspectivas actuales en la protección de los programas de ordenador. En C. Rogel Vide (Ed.), *Nuevas tecnologías y propiedad intelectual* (págs. 115 - 130). Madrid: Reus.
- Cavalli, J. (2006). *Génesis del Convenio de Berna para la protección de las obras literarias y artísticas de septiembre 9 de 1886*. Bogotá D.C.: Dirección Nacional de Derechos de Autor.
- Charfoos, A. D. (2002). How far have we come, and where do we go from here. The status of global computer software protection under TRIPS Agreements. *Northwestern Journal of International Business*, 22(2), 261-290.

- Chisum, D. S. (1986). The patentability of algorithms. *University of Pittsburg Law Review*, 47, 950-1022.
- Cohen, J. E. (1994-1995). Reverse engineering and the rise of electronic vigilantism: Intellectual property implications of "lock-out" programs. *Southern California Law Review*, 68, 1091-1202.
- Cohen, J. E., & Lemley, M. (2001). Patent scope and innovation in the software industry. *California Law Review*, 89(1), 1-57.
- Comander, S., Chanda, R., Kangasniemi, M., & Winters, L. (2007). India's software industry and cross-border labour mobility. *Journal Compilation*, 187 - 211.
- (1983). *Committee of Experts on the Legal Protection of Computer Software Second Session*. OMPI.
- Computer Associates Int'l v. Quest Software Inc., 333 F. Supp. 2d 688 (United States District Court, Northern District of Illinois, Eastern Division 28 de Jun de 2004).
- Computer Associates International Inc. v. Altai Inc., 982 F.2d 693, 23 USPQ 2d 124 (United States Court of Appeals, Second Circuit. 22 de Jun de 1992).
- Conpes 3582. (2009). *Política nacional de ciencia, tecnología e innovación*. Bogotá D.C.: DNP.
- Conpes 3678. (2010). *Política de transformación productiva: un modelo de desarrollo sectorial para Colombia*. Bogotá D.C.: DNP.
- Conpes 3834. (2015). *Lineamientos de política para estimular la inversión privada en ciencia, tecnología e innovación a través de deducciones tributarias*. Bogotá: DNP.
- Conrteras-Jaramillo, J. C. (2015). Origen y sustento del agotamiento de los derechos de propiedad intelectual. *Vniversitas*, 277 - 322.
- Contreras, J. (2016). A Tale of Two Layers: Patents, Standarization and the Internet. *Denver University Law Review*, 93, 855 - 898.
- Contreras, J. L. (2013). Fixing Frand: A Pseudo-Pool Approach to Standards-Based Patent Licensing. *Antitrust Law Journal*, 47 - 98.
- Contreras-Jaramillo, J. C. (2016). Algunos problemas del agotamiento de los derechos de propiedad intelectual respecto de las licencias de software. *Vniversitas*(132), 111 - 132.
- Cooter, R., & Ulen, T. (2012). *Law & Economics*. Boston: Pearson.
- Córdoba Marentes, J. F. (2006). Patentes sobre software: Desafío a la propiedad intelectual en la sociedad de la información. En E. Ossa Gómez (Ed.), *Hacia una comprensión humana del derecho. Estudios en homenaje a Roberto Suarez Franco* (págs. 123-144). Bogotá D.C.: Temis.
- Dam, K. W. (Junio de 1995). Some economic considerations in the intellectual property protection of software. (U. o. Press, Ed.) *The Journal of Legal Studies*, 24(2), 321-377.
- de la Cruz Camargo, D. (2008). Límites al uso de la patente, desde el punto de vista del derecho de la competencia, a propósito del caso Microsoft. *Con-texto*, 24, 119 - 141.
- Diamond v Diehr, 450 U.S. 175 (Supreme Court of The United States 3 de Mar de 1981).

- Díaz Vera, L. M. (2014). *Criterios de patentabilidad y derecho de la competencia: Comparación entre la legislación estadounidense y la colombiana desde la perspectiva de la biotecnología*. Bogotá D.C.: Universidad Externado de Colombia.
- Drexler, J. (2011). Refusal to grant access to trade secrets and abuse of market dominance. En S. Aderman, & A. Ezrachi (Eds.), *Intellectual property and competition Law. New frontiers* (págs. 165-185). New York: Oxford.
- Duffy, J. F., & Hynes, R. (2016). Statutory domain and the commercial Law of intellectual property. *Virginia Law Review*, 102, 1 - 78.
- Dukrok, S., & Hwang, J. (2010). An analysis of the effect of software intellectual property rights on the performance of software firms in South Korea. *Technovation*, 30, 376-385.
- Dynamic, Inc. v. Structural Software, Inc. , 785 F.Supp. 576. (United States District Court, E.D. Louisiana 1991).
- Enger, W. (1990). Korean copyright reform. *UCLA Pacific Basin Law Journal*, 7, 199-213.
- (2014). *Estudio de la caracterización de productos y servicios del sector de software y tecnologías relacionadas de Colombia*. Fedesoft, Bogotá D.C.
- Fedesarrollo. (2005). *Estudio sobre la evaluación del impacto de los incentivos tributarios otorgados para ciencia, tecnología e innovación*. Bogotá D.C.: Colciencias.
- Galán Corona, E. (2011). Violación de secretos. En A. Bercovitz Rodríguez-Cano (Ed.), *Comentarios a la ley de competencia desleal* (págs. 351-378). Cizur Menor: Thomson Reuters-Aranzadi.
- García, G. (2 de Jul de 2019). Financiar la economía naranja. *La República*, pág. 1.
- Ginsburg, J. C. (1994). Four reasons and a paradox: The manifest superiority of copyright over sui generis protection of computer software. *Columbia Law Review*, 94(8), 2559-2572.
- Gómez Segade, J. A. (2001). *Tecnología y derecho*. Madrid: Marcial Pons.
- Gottschalk v. Benson, 409 U.S. 63 (Supreme Court of The United States 20 de Nov de 1972).
- Gozzer, S., & Vicente, A. (2013). Cinco verdades sobre la piratería. *ANDA*.
- Graef, I. (2014). How can Software Interoperability be achieved under European Competition Law and Related Regimes? *Journal of European Competition Law & Practice*, 5(1), 6 - 19.
- Gruner, R. S. (2002). Intangible inventions: Patentable subject matter for and information age. *Loyola of Los Angeles Law Review*, 35, 355-470.
- Guerrero Gaitán, M. (2006-2007). Aspectos globales de la patentabilidad de las invenciones implementadas por odenador. Estado actual y nuevas perspectivas. *Propiedad Inmaterial*(10-11), 117-148.
- Gugliuzza, P. R., & Lemley, M. A. (2018). Can a Court change The Law by saying nothing? *Vanderbilt Law Review*(71), 765 - 820.
- Guía para examen de solicitudes de patente y modelo de utilidad*. (2012). Bogotá D.C.: SIC.
- Guzmán Caballero, A. (Agosto de 2018). 'Blockchain', el cambio de los paradigmas jurídicos. *Ámbito Jurídico*(496), pág. 16.

- Hangzhou Huatai Yimei Culture Media Co., Ltd. v. Shenzhen Daotong Technology Development Co., Ltd. , 055078 (2018) Zhe 0192 No. 81 (Hangzhou Internet Court Province of Zhejiang, China 27 de Jun de 2018).
- Hanrahan, D. (2010). Get your dirty gray market goods out of my country - Enforcement of patent rights in the EU. *National Italian American Bar Association Law Journal*, 18, 1 - 20 .
- Heshusius Rodríguez, K. (2009). Colombia: Desafíos de una industria en formación. En P. Bastos Tigre, & F. Silveira Marques (Edits.), *Desafíos y oportunidades de la industria del software en América Latina* (págs. 139-170). Bogotá D.C.: Cepal.
- Hong, X., & Cheng-ssu, C. (1999). *Software protection in China. A complete guide*. Hong Kong, Asia: Sweet & Maxwell.
- (2018). *Informe Nacional de Competitividad 2018 - 2019*. Bogotá D.C.: Zetta Comunicadores.
- Jacobsen v. Katzer, 535 F.3d 1373 (United States Court of Appeals, Federal Circuit 13 de Ago de 2008).
- Jiménez Valderrama, F. (2019). *Derecho de la competencia*. Bogotá D.C.: Legis.
- Junqueira Botelho, A. J., Stefanuto, G., & Veloso, F. (2007). The Brazilian software industry. En A. Ashish, & A. Gambardella (Edits.), *From Underdogs to Tigers: The rise and growth of the software industry in Brazil, China, India, Ireland, and Israel* (págs. 1-51). New York: Oxford Scholarship Online.
- Klein, B. (1993). Market power in antitrust. Economic analysis after Kodak. *Supreme Court Economic Review*, 3, 43-92.
- Laboratorios California S.A. v. System Software Associates Inc. y S.S.A. Colombia S.A. (Tribunal Arbitral enero de 2001).
- Landes, W. M., & Posner, R. A. (Junio de 1989). An economic analysis of copyright law. (T. U. Press, Ed.) *The Journal of Legal Studies*, 18(2), 325-363.
- Lasercomb America Inc. v. Job Reynolds, 911 F.2d 970 (United States Court of Appeals for the Fourth Circuit 16 de August de 1990).
- Leith, P. (2007). *Software and patents in Europe*. New York: Cambridge University Press.
- Lema, R., Quadros, R., & Schmitz, H. (2012). *Shifts in innovation power to Brazil and India: Insights from the auto and software industries*. Brighton: Institute of Developments Studies.
- Lemley, M. A. (2004). Ex Ante versus Ex Post Justifications for Intellectual Property. *The University of Chicago Law Review*, 71(1), 129-149.
- Lemley, M. A., Menell, P. S., Merges, R. P., Samuelson, P., & Carver, B. W. (2011). *Software and internet Law* (Fourth ed.). New York: Wolters Kluwer.
- Lexmark International Inc. v. Static Control Components Inc., 387 F.3d 522 (United States Court of Appeals for the Sixth Circuit 26 de Oct de 2004).
- Lipzyc, D. (2004). *Nuevos temas de derechos de autor y derechos conexos*. Bogotá D.C.: Cerlalc.
- Lipzyc, D. (2001). *Derecho de autor y derechos conexos*. Bogotá D.C.: Cerlalc.

- López, A. (2009). *Innovation and appropriability: Empirical evidence and research agenda*. OMPI, Geneva.
- Lotus Development Corp. v. Borland International Inc., 516 U.S. 233 (Supreme Court of The United States 16 de Ene de 1996).
- Maggiolino, M. (2011). The economics of antitrust and intellectual property rights. En S. Aderman, & A. Ezrachi (Edits.), *Intellectual property and competition Law. New frontiers* (págs. 73-92). New York: Oxford.
- Mansfield, E. (1984-2002). R&D and innovation: Some empirical findings. En T. Ruth, & R. Holzhauser (Edits.), *The economics of intellectual property* (Vol. III, págs. 15-42). Cheltenham: Edward Elgar Publishing Ltd.
- Márquez Escobar, C. P. (2004). *Introducción a la relación entre el derecho y la economía*. Bogotá D.C.: Pontificia Universidad Javeriana.
- Márquez Escobar, C. P. (2004). *Violación a los derechos de propiedad intelectual. El caso de la piratería de software*. Bogotá D.C.: Pontificia Universidad Javeriana.
- Mas-Collel, A., Whinston, M. D., & Green, J. R. (1995). *Microeconomic Theory*. Oxford University Press.
- Mayo v. Prometheus, 566 U.S. 66 (Supreme Court of The United States 20 de Mar de 2012).
- McGee, J. S. (1966). Patent exploitation: Some economic and legal problems. *The Journal of Law & Economics*, 9, 135-162.
- (1979). *Measures to enhance international cooperation in the field of legal protection of computer software*. OMPI. Geneva: OMPI.
- Menell, P. S. (1987). Tailoring legal protection for computer software. *Stanford Law Review*, 39(6), 1329-1372.
- Menell, P. S. (1989). An analysis of the scope of Copyright protection for application programs. *Stanford Law Review*, 1045-1104.
- Menell, P. S. (1994). The challenges of reforming intellectual property protection for computer software. *Columbia Law Review*, 94(8), 2644-2654.
- Merges, R. P., Menell, P. S., & Lemley, M. A. (2010). *Intellectual property in the new technological age* (Fifth ed.). New York: Wolters Kluwer.
- Michael, N., & Shankerman, M. (2006). *Strategic patenting and software innovation*. London: Centre for economic performance.
- Microsoft Corp. v. Motorola Inc, 14 - 35393 (Court of Appeal for the 9th Circuit 30 de Jul de 2015).
- (1978). *Model provisions on the protection of computer software*. Geneva: OMPI.
- Monroy Rodríguez, J. C. (2012). Cuestiones jurídicas en torno a los contratos de desarrollo y licencia de software. *La propiedad inmaterial*(16), 103-136.
- Murphy-Hill, E., & Grossman, D. (2014). How programming languages will co-evolve with software engineering: a bright decade ahead. *FOSE 2014 Proceedings of the on Future of Software Engineering* (págs. 145 - 154). Hyderabad: ACM.

- Nguyen, X.-T. N., Gomulkiewicz, R. W., & Conway, D. M. (2012). *Intellectual property, software, and information licensing: Law and practice*. Washington D.C.: BNA.
- Nimmer, R. T., & Krauthaus, P. (1987). Classification of computer software for legal protection: International perspectives. *International Lawyer*, 21(3), 733-754.
- Palacio Eastman, C. H. (1999). Protección jurídica del software. *Homos Iuris*(1).
- Parker v. Flook, 437 U.S. 584 (Supreme Court of The United States 22 de Jun de 1978).
- Paterson, M. (2013). Properly protecting code: solving copyright and patent rights overlap via computer software suitability in copyright. *Intellectual Property Journal*, 25(2), 173-194.
- Paul, R. A. (2012). Black and White: A Path Toward Clarity for Copyright Law and Gray Market Goods. *DePaul Journal of Art, Technology and Intellectual Property Law*, 23, 155 - 200.
- Pearson Education Inc. v LIU, 656 F. Supp. 2d 407, 93 (United States District Court for the Southern District of New York 25 de Sep de 2009).
- Peña Valenzuela, D. (Ed.). (2013). *Software libre y software propietario: Impacto jurídico, económico y cultural en Colombia*. Bogotá D.C.: Universidad Externado de Colombia.
- Pérez Gómez-Tétré, A. M. (2006-2007). El análisis económico del derecho de autor y del copyright, un estudio del equilibrio entre los intereses público y privado. *La Propiedad Inmaterial*(10-11), 3-22.
- Pérez Gómez-Tétré, A. M. (2012). Perspectivas económicas. En R. Metke Méndez, E. I. León Robayo, & E. Varela Pezzano, *Propiedad intelectual. Reflexiones* (págs. 395 - 421). Bogotá D.C.: Universidad del Rosario.
- Plata López, L. C. (2010). *Responsabilidad civil por infracciones al derecho de autor*. Barranquilla: Uninorte.
- Posner, R. (2001). *Antitrust law* (Segunda ed.). Chicago: University of Chicago.
- Posner, R. (2011). *Economic analysis of Law*. New York: Wolters Kluwer.
- Preinfalk Lavagni, I. (2010). *El derecho moral de autor de programas informáticos*. Valencia: Tirant lo Blanch.
- Rawls, J. (2010). *Teoría de la justicia* (2 ed.). Fondo de Cultura Económica.
- Reinganum, J. F. (1983). Uncertain Innovation and the Persistence of Monopoly. *The American Economic Review*, 73(4), 741 - 748.
- Rengifo García, E. (2003). *Propiedad intelectual* (Segunda ed.). Bogotá D.C.: Universidad Externado de Colombia.
- Rifkin, J. (2009). *El siglo de la biotecnología. El comercio genético y el nacimiento de un mundo feliz*. Barcelona: Paidós.
- Ríos Ruíz, W. R. (2002). El derecho de autor en la protección jurídica de los programas de ordenador – soporte lógico (software) y lo bancos o bases de datos. *La propiedad inmaterial*(5), 81-92.
- Rocha, H., & Ducasse, S. (2018). Preliminary Steps Towards Modeling Blockchain Oriented Software. *1st International Workshop on Emerging Trends in Software Engineering for Blockchain* (págs. 52 - 57). New York: ACM.

- Rodríguez Yong, C. (2012). Ley aplicable, libertad contractual y abuso del derecho en el contrato de licencia de software desde la experiencia norteamericana. En R. Metke Méndez, E. I. León Robayo, & E. Varela Pezzano, *Propiedad intelectual. Reflexiones* (págs. 321 - 340). Bogotá D.C.: Universidad del Rosario.
- Sakakibara, M., & Branstetter, L. (2001-2002). Do stronger patents induce more innovation? Evidence from the 1988 Japanese patent law reforms. En *The economics of intellectual property* (Vol. III, págs. 144-167). Cheltenham: Edward Elgar Publishing Ltd.
- Salas Pasuy, B. (2016). El software o programa de ordenador y el concepto de invención patentable. En E. Rengifo García (Ed.), *Derecho de patentes* (págs. 387- 490). Bogotá D.C.: Universidad Externado de Colombia.
- Salazar López, J. L. (2015). Patentes y software Colombia. *III Seminario Internacional de Propiedad Intelectual*. Bogotá D.C.
- Samuelson, P. (2015). Software patents are falling down. *Communications of the ACM*, 58(11), 27 - 29.
- Samuelson, P., & Nordhaus, W. (2006). *Economía*. Madrid: Mc Graw Hill.
- Samuelson, P., & Scotchmer, S. (May de 2002). The law an economics of reverse engineering. *The Yale Law Journal*, 111(7), 1575-1663.
- Samuelson, P., Davis, R., Mitchell D., K., & Reichman, J. H. (Dic de 1994). A manifiesto concerning the legal protection of computer programs. *Columbia Law Review*, 94(8), 2308-2431.
- Schortgen, S. (Nov de 1994). "Dressing" up software interface protection: The application of two pesos to "look and feel". *Cornell Law Review*, 80, 158-201.
- Sega Enterprises Ltd. v. Accolade, 977 F.2d 1510 (United States Court of Appeals for the Ninth Circuit 20 de Oct de 1992).
- Sen, A. (2006). What Do We Want from a Theory of Justice? *The Journal of Philosophy*, 103(5), 215 - 238.
- Sharma, A., Sengupta, S., & Gupta, A. (2011). Exploring risk dimensions in the Indian software industry. *Project Management Journal*, 42(5), 78 - 91.
- Sidak, J. G. (2013). The meaning of Frand: Part I Royalties. *Journal of Competition Law & Economics*, 9, 931 - 1055.
- Sidak, J. G. (2015). The meaning of Frand: Part II Injunctions. *Journal of Competition Law & Economics*, 11, 201 - 269.
- Smith, B. L., & Mann, S. O. (2004). Innovation and intellectual property in the software industry: An emerging role for patents? *The UNiversity of Chicago Law Review*, 71(1), 241-264.
- Soler Matute, P. (2004). *El contrato para la elaboración de programas de ordenador. Contrato de desarrollo de software*. Barcelona: Thomson Aranzadi.
- Stern, R. H. (2018). Who Should Own the Benefits of Standardization and the Value It Creates. *Minnesota Journal of Law, Science & Techonology*, 9, 107 - 250.
- Stockalper, M. (2010). Is there a foreign right of price discrimination United States Copyright Law - An examination of the first-sale doctrine as applied to gray-markets goods. *DePaul Journal of Art, Technology & Intellectual Property Law*, 20, 513 - 546.

- Swinson, J. (1991). Copyright or patent or both: An algorithmic approach to computer. *Harvard Journal of Law & Technology*, 145-214.
- Tamayo Jaramillo, J. (2011). *La decisión judicial* (Vol. I). Medellín: Diké.
- Téllez Valdés, J. (1989). *La protección jurídica de los programas de computación*. México: Universidad Nacional Autónoma de México.
- Teruo, D. (1992). *Intellectual property protection and mangement: Law and practice in Japan*. Tokyo: Institute of Comparative Law, Waseda University.
- (2016). *The Internet of things. Seizing the benefits and addressing the challenges*. OECD Digital Economy Policy Papers.
- (2012). *The software industry and developping countries*. United Nations, New York.
- Unwired Planet v Huawei, [2017] EWHC 711 (Pat) (Royal Court of Justice 5 de Abril de 2017).
- Uribe Arbeláez, M. (2005). *La transformación de la propiedad intelectual*. Bogotá D.C.: Doctrina y Ley.
- Vernor v. Autodesk Inc., 621 F.3d 1102 (9th Circuit 10 de Sep de 2010).
- Villalba, C. A., & Lipzyc, D. (2001). *El derecho de autor en la Argentina*. Buenos Aires: La Ley.
- Waelbroeck, M., & Frignani, A. (1998). *Derecho europeo de la competencia* (Vol. II). Barcelona: Bosch.
- West, J. (1995). Software rights and Japan's shift to an information society: The 1993-1994 Copyright. *Asian Survey*, 1118-1139.
- Whelan v. Jaslow, 797 F.2d 1222 (United States Court of Appeals for the Third Circuit 4 de Ago de 1986).
- Zuleta Londoño, A. (2005). Microsoft ante las autoridades comunitarias: reflexiones sobre política antimonopolios en operaciones internacionales. En C. Londoño, I. Macías, & J. Oviedo Albán, *Colección de derecho privado y globalización: Derecho mercantil contemporáneo*. Bogotá D.C.: Ibañez.