UNIVERSIDAD DE CHILE
FACULTAD DE DERECHO
ESCUELA DE POSTGRADO

ALCANCE DEL PRINCIPIO DEL "*FAIR USE*" EN LA PROTECCIÓN JURÍDICA DEL *SOFTWARE*:

Con especial referencia al caso Oracle América Inc. v. Google Inc.

Afet para optar por el grado de Magíster en Derecho, mención en Contratación Comparada e Internacional

Neida Zambrano Sánchez
Profesor Guía: Salvador Millaleo H.

Santiago de Chile, septiembre 2017

Tabla de contenido

RESUMEN

En los últimos años, diversos pronunciamientos jurisprudenciales norteamericanos recaído en el caso Oracle American Inc. v. Google Inc., han llamado la atención de la industria tecnológica, sobre sí el código declarativo y las SSO de las interfaces de aplicación (*Application Programming Interface*, en sus siglas en inglés APIs) empleadas para el desarrollo del sistema operativo Android, en su condición de elementos distintos al código fuente, también estaban protegidas por el *copyright*. Una decisión que apoyó esta postura de que sí estaban protegidas hizo que un Juzgado de Distrito conociera sobre el *fair use* (uso justo en español), o no de las interfaces. Este caso falló a favor de Google declarando que la implementación del código declarativo y sus secuencias, estructura y organización de las 37 APIs de Java, propiedad de Oracle efectivamente fue un uso justo, siendo clave dentro de la interoperabilidad de los programas informáticos y de la innovación.

INTRODUCCIÓN

El vertiginoso desarrollo de nuevas tecnologías, entre ellas los programas informáticos, ha exigido una rápida adaptación del Derecho. Esto se debe a que estos programas son unos de los motores de la sociedad, posicionados como grandes intermediarios entre la información y el ser humano. Ello puede verse en cosas tan usuales, como un mensaje de texto enviado a través de telefonía móvil, o en otras más complejas, como los procesos industriales, los sistemas financieros, los servicios de salud, etc. La importancia de los programas informáticos ha dado lugar a que legislaciones y tribunales discutan sobre el alcance de su protección jurídica.

Es importante señalar que la promoción y creatividad son conceptos tomados en cuenta por el Derecho para amparar la propiedad intelectual de los autores, otorgándoles derechos exclusivos sobre sus obras, los cuales pueden ser de carácter económico, personal o moral, no obstante, estos derechos no son absolutos, pues la legislación al mismo tiempo busca promover los beneficios sociales que el libre flujo de información y conocimiento de tales obras

pudiera aportar. A fin de lograr un equilibrio entre ambos objetivos, las normas de carácter internacional y las desarrolladas por cada país a lo interno han establecido ciertos límites al derecho de autor.

Uno de los fundamentos de esas limitaciones es beneficiar y apoyar la actividad investigadora e innovadora, por lo que suelen estar presentes en la gran mayoría de las legislaciones nacionales de derecho de autor, especialmente en las de aquellos países más avanzados científica y tecnológicamente, siendo una de las más importantes dentro de la legislación norteamericana del *copyright* la doctrina del "*fair use*" o uso justo, inspirada en la eficiencia económica, que viene a constituir un balance real entre los derechos de los titulares de la obra creativa y el público en general.

Esta doctrina del "*fair use"* ha adquirido relevancia con el surgimiento de la economía digital, en la medida que permite utilizar obras de un tercero necesarias para muchos de los negocios de alta tecnología y que son la base de la economía de Internet o para los desarrolladores a la hora de utilizar parcial o totalmente programas ya existentes para otro fin distinto para el cual fueron estos creados.

No es materia de discusión aquí que los programas informáticos son protegidos por el derecho de autor y que éste resguarda la forma de expresión que utilizan los desarrolladores para manifestar sus ideas, pues ello ha sido generalmente aceptado en la mayoría de los países. Siendo el código fuente donde las ideas se encuentran expresadas, ese ha sido el núcleo central de protección.

Ahora bien, en el caso del *software* hay que destacar que éste se encuentra constituido por componentes que van más allá de las expresiones contenidas en el código fuente, que constituyen elementos funcionales a los que una gran parte de la doctrina y jurisprudencia que se han pronunciado al respecto han considerado no protegibles por derecho de autor. No obstante, otra corriente considera que estos si pueden ampararse bajo este derecho.

Las interfaces constituyen un ejemplo de los elementos respecto a los que no ha existido una postura unánime acerca de si se encuentran incluidos o, por el contrario, escapan de esta protección. Éstas son de suma importancia en esta era de la tecnología debido a que son importantes dentro de interoperabilidad de programas informáticos que son desarrollados por

distintas personas (físicas o jurídicas) y que en muchos casos es necesario que se relacionen entre sí y compartan datos imprescindibles para su ejecución.

Lo anterior ha creado un poco de incertidumbre dentro del campo tecnológico, al no saber a ciencia cierta si se infringe o no el derecho del propietario respecto a los elementos funcionales de un *software*. Basándonos en el hecho de que sí están protegidos esos elementos, la herramienta de defensa que tendrán los desarrolladores o quien utilizó ese programa es el principio de "*fair use*" o uso justo el cual, como lo mencionamos en líneas anteriores, no es más que una limitante al derecho de autor que permite el aprovechamiento lícito de la obra por parte de terceros sin autorización previa de su creador original.

En los últimos años, diversos pronunciamientos jurisprudenciales norteamericanos han llamado la atención de los desarrolladores (empresas/particulares), abogados y, en general, del público tecnológico, dentro de los que destacan los recaídos en el caso Oracle v. Google, los cuales han representado grandes aportes a la definición del alcance de la protección

del *copyright* sobre los elementos funcionales del programa de computación. Un ejemplo de esto es el asunto que se debatió ante el Tribunal del Distrito Norte de California, con sede en la ciudad de San Francisco, sobre si las interfaces de aplicación (*Application Programming Interface*, en sus siglas en inglés APIs) empleadas para el desarrollo del sistema operativo Android, en su condición de elementos distintos al código fuente, también estaban protegidas por el *copyright*.

En un primer juicio se desarrolló el análisis de los elementos funcionales del *software*, en específico de las APIs, donde el tribunal de instancia determinó que las interfaces no estaban protegidas por el *copyright*, decisión que fue revocada en apelación por una Corte Federal, la cual adujo que sí lo estaban por constituir una expresión original y que lo que procedía era determinar mediante un nuevo juicio si el uso de éstas había sido justo o no, conforme al artículo 107 de la *Copyright Act.*

Como eje central en el nuevo juicio con jurados se discutió el uso, aparentemente indebido, de 37 interfaces de Java por parte de Google en el desarrollo del sistema operativo Android y cómo dicho uso pudiera

clasificarse o no como justo. La conclusión luego del largo juicio, por unanimidad del jurado, fue que Google utilizó de manera justa las 37 APIs de Java, propiedad de Oracle.

Se observa que la protección de los programas informáticos es un tema muy interesante dentro del Derecho, pues, aparte de los elementos técnicos que ellos revisten, aun no hay nada definido sobre ellos, ni siquiera en países desarrollados, como es el caso de Estados Unidos, donde actualmente se encuentran debates en tribunales sobre esta materia.

Es por esto que hemos encontrado interesante realizar mediante el presente trabajo un análisis del alcance del derecho de autor sobre los programas informáticos, con especial énfasis en sus limitaciones y excepciones desde la perspectiva de la doctrina del "*fair use"*.

Tomaremos la postura de la Corte Federal de los Estados Unidos que considera que las interfaces sí están protegidas por el derecho autor -aunque ello no quiere decir que estemos o no de acuerdo con tal posición- a los fines de realizar un estudio respecto a las implicaciones de la resolución que adoptó

el Tribunal Distrito Norte de California dentro del comentado caso Oracle v. Google que conllevó a analizar el uso justo de este tipo de programa informático, para con ello revisar cuáles fueron los fundamentos que llevaron a determinar las circunstancias bajo las cuales puede considerarse que es justo usar las interfaces de un tercero para realizar una nueva obra.

Es así que el mayor énfasis recaerá en la mencionada decisión, dictada en el año 2016, sin dejar de hacer un recuento de las dos que le antecedieron, para de esta manera tener una visión integral del asunto. En efecto, resultará necesario abordar la historia del caso para comprender el contexto de la última decisión dictada, la cual debe advertirse que está sujeta a una nueva apelación que aún no ha sido resuelta.

Posteriormente se hará referencia a la visión general del *software* a nivel técnico y legal, explicando qué son las APIs y cómo la ley de derechos de autor se une a los elementos literales y no literales del *software*. Se concluirá y explicará la doctrina del "*fair use"* o uso justo y sus interpretaciones según sus factores y aquellos elementos más emblemáticos en materia de programas informáticos.

CAPÍTULO I

CASO: ORACLE CORPORATION INC V. GOOGLE INC.

Antecedentes del juicio de Oracle v. Google

Actualmente está en discusión una disputa en materia de propiedad intelectual sobre el *software*, específicamente sobre las interfaces, sostenida entre Oracle Corporation Inc. contra Google Inc., por la plataforma de *software* Android para dispositivos móviles.

A manera de resumen, Oracle Corporation adquirió la compañía Sun Microsystems, Inc. en el año 2010 y con ello adquirió a su vez la participación del popular lenguaje de programación Java, el cual antes de tal adquisición había sido utilizado en Android.

Posteriormente a la adquisición, Oracle Corp. demandó a Google Inc. por infringir los derechos de autor de Oracle al copiar la secuencia, estructura y organización (SSO) de elementos de interfaces de programación de

aplicaciones (APIs)[1] e incorporarlos en su sistema operativo para teléfonos inteligentes llamado Android.

Se observa que, "Google creó el sistema operativo Android, el cual incluye ciertos segmentos del software de Java, por cuanto Google utilizó en este sistema 37 paquetes del código de las API de Java. El código copiado consiste en las declaraciones y las cabeceras de varias clases y métodos en la API de Java. Estos segmentos de código permiten a los programadores utilizar los mismos comandos en Android como lo harían al escribir un programa Java ordinario y, por lo tanto, ahorraría a los programadores de aprender un lenguaje de programación diferente*"*[2]. Oracle señaló que era innecesario que Google utilizara los mismos nombres de funciones que Java o que organizara los nombres de la misma manera.

Google, entre otros argumentos, sostuvo que el lenguaje Java utilizado para su sistema operativo Android era abierto y libre, como también lo eran las 37

[1]Las API son paquetes de instrucciones que ayudan a los programas informáticos para que interactúen entre sí.
[2]TREVOR M., Jury Returns Verdict for Google in Question of Fair Use of Oracles Code, Journal of Intellectual Property Law & Practice 2016; [en línea] doi: 10.1093/jiplp/jpw112

Interfaces de Programación de Aplicaciones (APIs) de Java que fueron utilizadas, las cuales como se dijo antes, para el momento de su uso eran propiedad de Sun Microsystems, comprados posteriormente por Oracle.

En mayo de 2012, la Corte de Distrito de California (Noveno Circuito de California) determinó que las APIs no pueden ser protegidas por el *copyright*. Aun reconociendo que sean expresiones y que cumplan con el requisito de originalidad, la Corte determinó que éstos son "métodos de operación", pues conforman "una lista de más de seis mil comandos para llevar a cabo unas funciones asignadas previamente", razón por la que estarían excluidas de la protección de este derecho[3].

En la mencionada decisión, el Tribunal de Distrito dictaminó que las APIs de Java (propiedad de Oracle), podían considerarse como lenguaje de codificación, es decir, como un "conjunto utilitario y funcional de símbolos". Es así que este Tribunal encontró que la reutilización de los paquetes de APIs

[3] SENTENCIA ORACLE AMÉRICA INC. v. Google INC., No. C 10-03561 WHA, United States District Court for the Northern District of California, [en línea] https://www.eff.org/es/document/judge-alsup-ruling-java-apis-uncopyrightable

de Java eran necesarios para lograr un grado de interoperabilidad con otros programas Java[4].

El Tribunal, para sustentar la no protección de las APIs por el *copyright,* se basó en el precedentes de Sega v. Accolade, señalando que en este caso se permitió hacer ingeniería inversa del código de Sega para que Accolade creara sus propios juegos, y que el *copyright* no era suficiente para lo que Sega buscaba proteger, pues de lo contrario Sega tendría un "monopolio de facto" en la funcionalidad de sus productos, lo cual el legislador ha rechazado específicamente, al indicar que "para poder disfrutar de un monopolio sobre la idea o el principio funcional que subyace a una obra, el titular de la obra debe satisfacer las normas más estrictas impuestas por las normas de patentes"[5]

Posterior a la decisión mencionada, una Corte de Apelaciones Federal de Estado Unidos en el año 2014 declaró que la estructura, secuencia y organización de una API sí estaba protegidas por el *copyright.*

[4] Sentencia up supra citada.
[5] GEO. L., Revista (2016), p. 66, [en línea] https://perma.cc/X6WV-Z28A

La referida decisión se fundamentó en el precedente conforme al cual la protección de derecho de autor sobre programas informáticos, abarca tanto elementos literales (protección del código fuente y del código objeto del programa), como no literales del programa (por ejemplo, la SSO y las APIs expresivas, es decir, revestida de originalidad). En esa ocasión, la Corte Federal devolvió el caso a la Corte de Distrito de California, con el propósito de que se hiciera un nuevo juicio y el jurado dictara un nuevo veredicto para determinar si la acción de Google de utilizar un material protegido podría constituir un "uso justo" de la tecnología de Oracle y de esa manera establecer si fue infringido o no el *copyright*.

Señala CANALES que "un leve ajuste en la estrategia de defensa desarrollada por Oracle en segunda instancia (que se dirigió a subrayar la copia literal de Google de las declaraciones o nombres de las API de Java, que fue considerada entonces una copia de parte del código, independiente de que la implementación hubiese sido desarrollada por Google a través de su propio código fuente), selló el entendimiento de la corte de segunda

instancia en cuanto a que Google habría copiado tanto elementos literales (declaraciones) como no literales (SSO) de 37 paquetes de API de Jav*a*"[6].

El caso retornó a la Corte de Distrito y se tramitó un nuevo juicio sobre el "*fair use"* de Google de las APIs utilizadas en el sistema operativo Android. Siendo en mayo del año 2016 cuando un jurado acordó por unanimidad que el uso por Google de las APIs de Java fue justo, por lo que éste no infringió los derechos de autor propiedad de Oracle.

Decisión del "*Fair Use"* sobre las 37 Interfaces de Java utilizadas en el SO Android

A los efectos del juicio, se siguió el criterio establecido por la Corte de Apelaciones de que Google utilizó las 37 APIs de Java protegidas por el *copyright*, y la cuestión que quedaba por decidir es si el uso de Google fue o no un uso justo.

[6] CANALES M., "O*racle versus Google: Copyright protection on functional elements of software's that puts at risk interoperability and innovation*", p. 188. [en línea] http://www.rchdt.uchile.cl/index.php/RCHDT/article/viewFile/37361/40376

Entonces, observamos que el punto de discordia es, sobre las líneas del código de declaración para cada método dentro de las 37 APIs, también conocidas como las líneas de encabezado, que Google usó en Android, así como la estructura, secuencia y organización de estas APIs[7]

Dentro de los hechos más importantes del caso que instruyó el Juez Alsup[8] a los jurados, podemos rescatar lo siguiente:

1. La plataforma Java es una plataforma de aplicación de *software* que se utiliza para escribir y ejecutar programas en el lenguaje de programación denominado "Java". Este lenguaje es gratuito y está disponible de manera libre para su uso sin permiso de nadie. La plataforma Java incluye, entre otras cosas, los paquetes de Java API (Application Programming Interface), los cuales son la figura central en este caso.
2. Los paquetes de APIs de Java, son un conjunto de programas de computadora y pregrabados que se utilizan para realizar funciones

[7]The United States District Court for The Northern District Of California Notice Of Final Charge To The Jury (Phase One) And Special Verdict Form, caso Oracle América INC v. Google INC., p. 9 [en línea] https://es.scribd.com/doc/313560685/Jury-Instruction-Oracle-v-Google
[8] Ob. citada ut supra.

comunes sin necesidad de que el programador reescriba el código desde cero.

3. Estos programas informáticos pregrabados ayudan a los desarrolladores a escribir aplicaciones.

Es importante también considerar que los programas informáticos funcionan ejecutando operaciones/instrucciones (comandos) que indican al programa qué hacer para realizar diferentes tareas.

Podemos indicar entonces que la biblioteca Java API contiene programas de código fuente Java prescritos para comunes y más avanzadas funciones de computadora. Estos están organizados en paquetes, clases y métodos.

Un paquete es una colección de clases y, a su vez, cada clase es una colección de métodos (y otros elementos). Cada método realiza una función específica, ahorrando a un programador la necesidad de escribir un código desde cero para realizar esa función.

A su vez hay dos componentes de paquetes: el código declarativo y el código de implementación. El juez instruyó al jurado explicándole lo que significa el código declarativo, al señalar que “es la línea o las líneas de código fuente que introducen, nombran y especifican el paquete, la clase o el método, los cuales reflejan la estructura, secuencia y organización (o "SSO"). El SSO especifica las relaciones entre los elementos de los paquetes de Java API y también organiza las clases, los métodos y otros elementos del paquete”[9].

Para comprender lo anterior, podemos traer a colación el ejemplo en el que se alude a la colección de paquetes de APIs de Oracle como una biblioteca. Cada paquete es como una estantería en la biblioteca. Las clases son como los libros que se encuentran en la estantería. Los métodos son como cada capítulo de instrucciones en un libro. El código declarativo sería como el título de cada capítulo. El código de implementación es el texto del capítulo.

Respecto al caso que nos ocupa, observamos los métodos se agrupan en varias clases, con las clases agrupadas bajo varios paquetes, como en "Java.lang. Math", siendo "Java.lang" el paquete y "Java.lang.Math" la clase.

[9] Ob. citada ut supra

La taxonomía particular adoptada para la API de Java refleja su sistema de archivos único, es decir, su estructura, secuencia y organización[10].

La capa inferior del sistema de Google -el núcleo del sistema operativo[11]- utiliza el kernel Linux de código abierto[12]. Para la capa intermedia, Google desarrolló su propia máquina virtual - "Dalvik" - para ejecutar programas en Java y otros lenguajes. La capa superior consistía en paquetes de APIs. Google escribió más de cien propios paquetes fuera de los paquetes de APIs de Java, pero también usó treinta y siete APIs de Java relevantes para una plataforma del sistema operativo de smartphone Android. Google copió el código declarativo para esas APIs de forma textual, para permitir a los

[10]Traducción de la autoría de caso: Oracle v. Google No. C 10-03561 WHA, 2016, antes citado, p.6.

[11] "El núcleo o kernel gestiona todos los procesos, es el encargado de llevar la cuenta de todos los procesos activos y de la planificación de los mismos, al seleccionar cuál de ellos ocupara tiempo del procesador, esta capa es muy importante, dado que define el rendimiento del sistema", [en línea] https://sistemasumma.com/2013/03/16/las-capas-del-sistema-operativo/

[12] Es un sistema operativo de uso libre y gratuito (código abierto). Linux es una alternativa a los sistemas operativos Windows y OS/X, más conocidos. Desde un punto de vista técnico, Linux es un "kernel", o el componente básico de un sistema operativo (mayor información ver: http://www3.lenovo.com/es/es/faqs/pc-life-faqs/que-es-linux/)

desarrolladores de aplicaciones encontrar aquellas funcionalidades con los mismos nombres utilizados en Java dentro del sistema operativo Android[13].

Podemos señalar hasta acá que, en cuanto a los paquetes objeto de controversia, las bibliotecas de Java y Android se organizan de la misma manera básica (código declarativo). Todos los documentos en Android se han escrito con implementaciones diferentes de Java (código de implementación), pero solucionando los mismos problemas y proporcionando las mismas funciones[14].

Todo lo anterior, se estima fue considerado por el jurado al momento de decidir el caso del uso justo de los 37 paquetes de Java, conforme a lo previsto en la Sección 107 del *Copyright Act* de 1976[15].

[13] TYLER D., North Carolina Journal of Law & Technology Volume 18, issue on.: December 2016 1 Oracle v. google: Setting a Standard or Handicapping an Industry? [en línea]
file:///C:/Users/Neida%20Zambrano/Downloads/Demasky_Final%20(1).pdf
[14] Ob. citada ut supra
[15] La referida norma señala que "el uso equitativo" de una obra protegida por derechos bajo los supuestos allí previstos no constituye una infracción del derecho de autor".

Los factores del *"fair use"* en las APIs de Java

La Sección 107 del *Copyright Act*, establece cuatro factores que deben ser analizados para determinar si el uso de una obra en puede ser catalogado como de uso justo, como son el propósito del uso, la naturaleza de la obra protegida por derecho de autor, la magnitud de la porción de la obra usada y protegida mediante derecho de autor y el efecto del uso sobre el mercado o el valor de la obra protegida.

Ahora bien, con respecto al primer factor recaído en la decisión "*fair use"* de las APIs, se puede evidenciar que éste se basó en una aparente conducta de mala fe de Google, siendo uno de los argumentos más enfáticos de Oracle sobre del "*fair use"*, alegando que el jurado podía considerar la mala fe de Google como un subfactor de este factor, en virtud de que Google "sabía" que necesitaba una licencia de los paquetes de Java que trató de conseguir pero finalmente no la obtuvo y en su lugar eligió utilizarlos sin autorización,

explotando gratuitamente un trabajo robado que podría haber obtenido de otro modo.[16]

Como defensa de lo anterior, la evidencia de la buena fe de Google se manifestaría mediante las declaraciones de sus testigos que indicaron que el "reimplementar" una biblioteca de APIs era una legítima y reconocida práctica de siempre; que todo lo que se duplicó fue el código de declaración, mientras el duplicador (autor de la segunda obra, en este caso Google) suministró su propio código de implementación[17].

Esta declaración la ratificó Google al señalar que consideraba que había seguido una práctica reconocida en la industria tecnológica respecto a la libre implementación de bibliotecas de APIs, duplicando solo el código declarativo.

[16] CASO: ORACLE V. GOOGLE No. C 10-03561 WHA, 2016 Distrito Noveno de California 8 de junio de 2016), p.2 [en línea]
https://assets.documentcloud.org/documents/2856497/06-08-16-Order-Denying-Rule-50-Motions.pdf

[17] Sentencia cit. *ut supra*, traducción de la autora, p. 3-4

Es importante, recordar en este sentido que las interfaces de *software* "se pueden considerar como un grupo de expresiones funcionales que deben ser duplicadas palabra por palabra, por lo que no da la posibilidad de que exista opción de expresar la misma idea de una manera diferente, y esto es necesario entre otras cosas, a la estandarización de los programadores al utilizar una expresión idéntica de la información de la interfaz con el fin de establecer la interoperabilidad"[18].

A su vez, Google argumentó que, si sus acciones fueron consistentes con la práctica industrial elevándola al nivel de "costumbre", se descartaría el alegato de mala fe expuesto por Oracle, basándose en el caso "*Wall Data INC v. Departamento del Shérif del Condado de los Ángeles*", donde se afirmó que "el uso justo es apropiado cuando un titular razonable del derecho de autor hubiese consentido el uso, es decir, cuando la costumbre o la política pública en ese momento hubieran definido el uso como razonable"[19]

[18] Ob. citada GONZÁLEZ B., The Software Interoperability Debate on European Copyright Law, p. 45

[19] En comillas es traducción literal al español de la autora de la decisión citada ut supra, p. 4.

El jurado bajo las instrucciones del Juez consideró que, respecto a la buena fe de Google, debían evaluar la medida en que éste actuó junto con todas las demás circunstancias, y si infringió cualquier práctica reconocida en la industria con respecto a la reinstalación de las APIs.

Observamos entonces que en este caso se consideró la costumbre como un alegato a los fines de evaluar la buena fe de Google, pues es un hecho notorio que en el sector de la tecnología se copian las interfaces y es un uso común para los desarrolladores del lenguaje Java utilizar las APIs de ésta.

Otro punto que se trató en el primer factor fue lo que respecta a lo comercial en el sistema operativo Android. El Tribunal de Distrito determinó que el estatus de "código abierto" de Android atenuaba los objetivos comerciales generales de Google. Es decir, dependía de la "disponibilidad libre y abierta de Android"[20] como contrapeso a la comercialidad de la conducta de Google alegada por Oracle.

[20] Ob. citada *ut supra.*

Uno de los fundamentos que se utilizó para desvirtuar la no comercialización del referido sistema operativo, es que éste funciona como un *software* de código abierto de conformidad con los propósitos de la Ley de Derecho de Autor. La Corte finalmente encontró que Google copió sólo el código necesario para mantener la continuidad intersistemas en beneficio de todos los usuarios de Java.

No obstante, al jurado se le dio la instrucción que el uso por Google de las APIs podría considerarse como comercial, por lo que ellos tenían que determinar el grado de comercialidad y, dependiendo del resultado, debían considerar si afectaba o no el uso justo.

Es importante citar textualmente lo que señala la sentencia sobre el grado de comercialización que se le dio a la utilización de las APIs de Java por parte de Google. La decisión indica que “En este caso coinciden en que el uso de Google era de naturaleza comercial, pero no se está de acuerdo con el alcance. El uso comercial pesa en contra de una constatación de uso justo, pero incluso un uso comercial puede ser (o no, según sea el caso) suficientemente transformador del primer factor estatutario que, en conjunto, sigue recortando

a favor del uso justo. Para decirlo de otra manera, cuanto más transformador sea un trabajo, otros factores, como es el comercial, retrocede en importancia. Por el contrario, cuanto menos transformador sea el trabajo del acusado, el factor comercial dominará"[21].

Entonces, el uso justo va a sustentarse en gran parte en lo diferente que puede ser el propósito de la obra, esto es, en lo transformador que puede llegar a hacer el nuevo programa cuando el desarrollador reimplementa el código declarativo y las SSO de la interface preexistente en un contexto totalmente diferente para el cual es utilizada, es decir, cuando hablamos del propósito y el carácter del uso se tiene que determinar en qué medida el trabajo es transformador o no.

Aquí nos parece importante traer a colación una frase de FISHER (citado por GARCÍA), que indica lo siguiente: *"The concept behind fair use is that creativity often requires the use of others works for the expression of ideas*",

[21] Caso: Oracle v. Google No. C 10-03561 WHA, 2016 Distrito Noveno de California 8 de junio de 2016), p. 13, [en línea] https://assets.documentcloud.org/documents/2856497/06-08-16-Order-Denying-Rule-50-Motions.pdf,. Traducción de la autora.

con lo que se pudiera decir que detrás del concepto del uso justo a menudo, se requiere el uso de obras de otros para crear/expresar obras/ideas nuevas[22].

Ahora bien, para determinar si es transformador, es importante que el juez o en este caso los jurados se preguntaran si la obra suplantaba meramente a la obra original, esto es, verificar para qué se aplican las APIs de Java o si ésta añadía algo nuevo, de tal manera que hiciera ver que el sistema operativo Android tuvo un propósito diferente.

En el caso Oracle v. Google, el Juez cita el caso Campbell v. Acuff-Rose Music, indicando lo que se entiende por obras transformadoras, sosteniendo que es todo aquello que "agrega algo nuevo, con otro propósito o carácter diferente, alternando el primero con una nueva expresión, significado o mensaje"[23].

Dentro de los argumentos de Oracle, éste señala que no sería razonable encontrar que el uso de Google del código declarativo y SSO de 37 paquetes

[22] Ob. citada, [en línea] http://propintel.uexternado.edu.co/el-papel-del-juez-frente-a-las-limitaciones-y-excepciones-al-derecho-de-autor-en-los-paises-de-tradicion-continental/)
[23] Ob. ut supra citada

de APIs de Java en Android, fue implementado en una nueva expresión, significado o mensaje.

Podemos traer a colación el caso Sony, donde el Tribunal determinó que, a pesar de lo similitud de usos y funciones de la obra copiada, el software del demandado podía considerarse como una obra transformadora, por cuanto era un nuevo producto, que consistía en cantidades significativas de códigos que la parte demandada había implementado (escrito).

El anterior criterio es similar al caso Oracle v. Google, pues mientras Google copió 37 de 166 posibles APIs de Java en Android, éste escribió su propio código de aplicación, la máquina virtual y otras APIs para dicho sistema operativo, por lo que debe ser considerado como una plataforma de teléfono inteligente innovadora[24].

Lo anterior fue una de las defensas de Google al argumentar que sus acciones no tenían relación con el mercado de Oracle para Java, en virtud de que

[24] Ob. citada, [en línea] https://www.stanfordlawreview.org/online/transformative-use-in-software/

Oracle y Sun Microsistems no adoptaron Java para dispositivos móviles, incluso utilizando las mismas APIs de este caso.

Demostró Google que no sólo el uso de su propio código de implementación para las treinta y siete APIs de Java, cambia elementos de la obra protegida por derechos de autor, sino que también puso las APIs en un contexto diferente. Esto fue transformador, incluso de acuerdo con los propios empleados de Oracle[25].

El Juez ALSUP[26] señaló lo anterior, al indicar que el jurado encontró que los 37 paquetes de Java fueron combinados con nuevos métodos, clases y paquetes escritos por Google para el sistema Android, lo cual permite concluir que este sistema era un nuevo contexto y que daba una nueva obra con nueva expresión, significado o mensaje duplicado distinto para lo cual aplicaban las APIs de Java.

[25] DEMASKY T., *Oracle v. Google: Setting A Standard Or Handicapping An Industry?*, p. 22.
[26] Decisión citada, p. 14

Así lo tratan de justificar unos científicos de la computación que introdujeron un *Amicus curiae*[27] a favor del uso de las APIs Java en Android. Por un lado, señalan que "los teléfonos inteligentes utilizan pantallas táctiles, mientras algunas APIs Java presuponen una interfaz de usuario con ratón y teclado, es decir al reimplementar estas APIs, Android fabricó una nueva API específica para pantalla táctil. Por otro que, "los teléfonos inteligentes utilizan varios sensores, incluyendo los GPS, acelerómetros, cámaras, brújulas, y los micrófonos. La API de Java no se aborda adecuadamente a estos periféricos, que son menos centrales a la configuración de escritorio"[28].

Asimismo, argumentaron que "por ser las baterías de los teléfonos móviles más pequeñas que las de una computadora, ello significaba que los primeros están casi siempre necesitados de energía eléctrica, por lo que se ejecutan en un tipo diferente que en un procesador de la computadora (chips ARM), que

[27] Una persona o grupos de personas con un fuerte interés o puntos de vista sobre el objeto de una acción judicial y que no es parte de ésta, puede solicitar a un Tribunal permiso para presentar un escrito para aportar su punto de vista respecto a los hechos que se discuten en el juicio. Estos escritos *amicus curiae* se presentan comúnmente en apelaciones relativas a asuntos de interés público general.

[28] BRIEF *of Computer Scientists as Amici Curiae in Support of Defendant-App*ellee, caso: Oracle v. Google. Nos. 2017-1118, 2017-1202, [en línea] https://www.eff.org/files/2017/05/31/2017.05.30_computer-scientists-fair-use-amicus-brief_oracle_v_google.pdf

son más eficientes energéticamente" [29]. Aunque Android utilizó el lenguaje Java y reimplementó muchas APIs esenciales de Java, al mismo tiempo incorporó muchas APIs nuevas para formar una plataforma que trasciende al programa original.

Después de todo, una interfaz sirve a un propósito funcional, pero no puede funcionar sin una implementación. Mientras que las APIs declarativas son necesarias para lograr la interoperabilidad, estas constituyen una porción insignificante del código necesario para definir y ejecutar la interface dentro de la plataforma Android, porque como se evidencia de la sentencia aquí citada, los códigos declarativos comprenden menos de una décima parte de un por ciento del código.[30] Por ello, la reutilización de esa pequeña porción de Java para la interoperabilidad es suficiente para considerar que Google fue capaz de desarrollar una plataforma innovadora y transformadora.

Entonces para un análisis relevante del uso justo respecto a la implementación del código declarativo "no debería ser si el uso transforma

[29] Ob. citada ut supra.
[30] Ob citada ut supra.

el código declarativo per se, sino si la implementación como un todo -incluyendo el código declarativo- es una obra nueva y diferente. Un programador que reutiliza una pieza de código existente puede apartarse radicalmente del propósito subyacente de la obra original implementando el paquete de forma diferente, conservando el código de declaración básico para asegurar que el trabajo permanezca compatible con otros"[31].

Es así que, si bien Google utilizó las 37 APIs no puede considerarse que el uso haya sido bajo el mismo concepto para el cual se utilizaban éstas, es decir, para computadores, por cuanto Google las utilizó en un sistema y un mercado totalmente diferente y bajo la combinación de un nuevo contexto, lo cual resultó una obra novedosa, de uso libre para los desarrolladores y con nuevos beneficios para los consumidores finales.

En cuanto al segundo factor consistente en el análisis de la naturaleza de la obra no se enfatizó mucho. El análisis del jurado, señala el Juez Alsup, iba a

[31] BRIEF *of Amicus Curiae Mozilla Urging Affirmance of the Judgment*, [en línea] https://www.eff.org/es/document/mozilla-amicus-brief

depender de si el trabajo realizado por Google en el sistema Android era creativo o funcional.

Aquí debemos señalar, como lo veremos más adelante, que cuando mencionamos este factor, se debe tomar en consideración la creatividad del material que fue utilizado en la obra nueva, por tal motivo menos material utilizado en la obra nueva implique protección de derechos de autor, más factible será el uso justo.

En su defensa Oracle se basó en destacar lo que dijo uno de los testigos de Google, quien diseñó y desarrolló varias de las APIs mientras trabajaba en Sun Microsystems y quien más tarde trabajó en el equipo de Android para Google. Dicho testigo explanó que uno de los desafíos que enfrentó él al diseñar las Apis de Java fue la complejidad de implementar la mejor manera de expresar lo que el programador quería hacer[32].

[32] *The United States District Court for The Northern District of California Notice of Final Charge to the Jury (Phase One) and Special Verdict Form*, en el caso: Oracle America INC v. Google INC., p. 15 [en línea] https://es.scribd.com/doc/313560685/Jury-Instruction-Oracle-v-Google

Oracle tomó dicha declaración y dijo que el testigo se había referido a que el diseño de la API era "un arte no una ciencia"[33], con lo que trató de decir que sus interfaces eran altamente creativas y no funcionales. Por su parte, otros testigos de Google enfatizaron el papel funcional de las líneas del código declarativo y su SSO, lo cual minimizó el aspecto creativo de las APIs.

Al respecto, se instruyó al jurado que:

1. Si el código declarativo/estructura, secuencia y organización son dictados por consideraciones funcionales tales como la compatibilidad o los estándares de la industria, debe considerarse menos creatividad en la obra.
2. Cuando los elementos esencialmente funcionales se implementan en un programa protegido por derechos de autor y es necesario copiar los elementos creativos asociados para utilizar esos elementos funcionales, entonces esta circunstancia también favorece el uso justo.

[33] OB. Citada *ut supra*

3. Copiar la expresión creativa que no era necesaria para realizar las funciones iría en contra el uso justo[34].

Aquí, es importante traer a colación, tal y como lo hizo el Juez Alsup, al caso Sega Enterprises, cuando se señala que, si se considera que un trabajo es en gran parte funcional, solo puede recibir una protección débil[35].

En el caso de Sega se mencionó el ejemplo del conocido teclado Qwerty. Allí se refirió que quizás existirán novedosos teclados de escritura, pero el teclado Qwerty es el que domina el mercado por cuanto es el que todos han aprendido a usar[36].

Asimismo, es oportuno traer a colación lo que señaló Mozilla Corporation en el Amicus Curiae presentado en el caso aquí analizado, al señalar que utilizar un código declarativo de una interfaz para crear un nuevo programa, puede

[34] In the United States District Court for The Northern District of California Notice of Final Charge to the Jury (Phase One) and Special Verdict Form, en el caso Oracle América INC v. Google INC., p. 16, [en línea] https://es.scribd.com/doc/313560685/Jury-Instruction-Oracle-v-Google
[35] Citado ut supra en la decisión en la nota al pie de la p. 16.
[36] Citado María Paz Canales L. en nota a pie p. 192 "Lotus v. Borland, 49 F.3d 807 (1995), párrafos 819-820".

pesar en favor de un hallazgo de uso justo, particularmente debido a la naturaleza funcional que reviste el código en cuestión, precisando que si los 37 paquetes APIs están protegidos por derechos de autor, sus aspectos funcionales pueden considerarse un peso en el análisis del uso justo, es decir, cuanto más funcional sea el código, más probable es que su uso sea justo.[37]

En el caso de las 37 APIs, el grado de creatividad está limitado por la función con que éstas están diseñadas. Se observa entonces que, para descartar este factor, unas de las consideraciones que predominó es el papel funcional del código declarativo y su secuencia estructura y organización en contraposición al aspecto creativo de las APIs, lo cual debió haber prevalecido en la decisión del jurado, tal y como concluyó el Juez Alsup en su decisión.

En lo que respecta al tercer factor, aquí se debe considerar la cantidad y sustancia que fue utilizado/copiado de la obra original.

[37] Brief of Amicus Curiae Mozilla Urging Affirmance of the Judgment, [en línea] https://www.eff.org/es/document/mozilla-amicus-brief

Así pues, se instruyó al jurado que debían considerar como un uso no justo el hecho que la parte sustancial de una obra infractora fuese copiada textualmente o, si fuese copiada una pequeña parte, ésta fuese la más importante. Al contrario, debían evaluar si el usuario secundario sólo copiaba lo necesario para un uso transformador, entonces este factor no pesaría en su contra[38].

El referido Juez puso énfasis en la cantidad cualitativa de trabajo protegida por derechos de autor utilizada en lugar de la cantidad cuantitativa, incluso concluyendo que el número total de líneas Android fue considerado irrelevante desde la perspectiva que las líneas copiadas fueron una pequeña porción dentro del sistema operativo. Señalando además que Google solo copió el código declarativo que era necesario para mantener la consistencia entre sistemas de usuarios de Java.

[38] In The United States District Court for The Northern District of California Notice of Final Charge to the Jury (Phase One) and Special Verdict Form, en el caso Oracle América INC v. Google INC., p. 17, [en línea] https://es.scribd.com/doc/313560685/Jury-Instruction-Oracle-v-Google

Es así que, se desprende que el noventa y siete por ciento del código fuente en los paquetes de API es diferente; es sólo el tres por ciento que señalan formó el corazón de la reclamación de Oracle. Ese tres por ciento incluía paquetes, métodos y nombres de clases. Pero esas declaraciones sólo pueden usarse de ciertas maneras, ya que, por su funcionalidad, el lenguaje exige que la declaración del método se implemente de una forma particular[39].

Se observa entonces que bajo este supuesto se examinó tanto el porcentaje que abarca las 37 APIs dentro de todo el contenido del sistema Android, como la importancia que éstas tienen dentro del funcionamiento del sistema, descartándose sin duda que éstas puedan constituir en algún sentido una parte tan considerable en la obra como para que pueda violentar el *copyright*.

Con respecto al cuarto factor, se tomó en consideración el mercado potencial, el daño o el uso irrestricto y generalizado de los materiales protegidos y si fue afectado o no o el valor de la obra protegida. Este se debe de analizar en conjunto con los otros factores y considerar si la segunda obra (infractora) se

[39] MULLIN J., Google wins crucial API ruling, Oracle's case decimated, ARS Technica, [en línea] https://arstechnica.com/tech-policy/2012/05/google-wins-crucial-api-ruling-oracles-case-decimated/

ofrece o se utiliza como un sustituto de la obra original protegida con derechos de autor y la importancia de este factor variará no sólo con la cantidad de daño mostrado, sino también con la fuerza de las proyecciones sobre los otros factores[40].

En este factor el efecto del mercado se relaciona con la manera en que se utilizó el material protegido por el *copyright*. En cuanto a Google, se evidencia que éste utilizó las APIs de Java para un sistema operativo móvil y Oracle no utilizaba las APIs bajo este mercado, sino en el de computadoras de escritorio y portátiles.

Por todo lo anterior, se concluyó que no podía considerarse que hubo un daño al mercado que pudiera afectar al uso justo de las APIs implementadas en Android.

[40] *In The United States District Court for The Northern District Of California Notice Of Final Charge To The Jury (Phase One) and Special Verdict Form,* en el caso Oracle América INC v. Google INC., p. 18 [en línea]
https://es.scribd.com/doc/313560685/Jury-Instruction-Oracle-v-Google

CAPÍTULO II

GENERALIDADES DEL SOFTWARE Y SU PROTECCIÓN COMO PROPIEDAD INTELECTUAL

El *software* desde una perspectiva técnica

Dentro del amplio mundo de la tecnología se encuentra la industria del *software*, el cual constituye uno de los elementos imprescindibles de cualquier sistema informático y, por consiguiente, encuentra utilidad en casi todos los ámbitos de la vida humana. Se puede observar como la economía de los países, especialmente la de los más desarrollados, depende en gran parte del *software*, en virtud de que éste controla los sistemas informáticos que son requeridos en los procesos industriales, el sistema financiero, las telecomunicaciones, entre otras muchas áreas.

Cabe recordar que al inicio de la era informática la separación económica entre *software* y *hardware* no existía, pues estos hacían un uno, lo que se vendía era el hardware por lo que el *software* no tenía ningún valor

económico, ni tampoco era un tema quién lo creo y por consiguiente quién era propietario.

Como señala GUERRERO, fue a finales de los años sesenta del siglo pasado que surgieron corrientes que se pronunciaron a favor de considerar la autonomía del hardware y a separar éste del software, lo que motivó que comenzaran a aparecer desarrolladores independientes de programas informáticos. Tal circunstancia ha conllevado a su vez a que desde finales del siglo XX la industria del *software* haya crecido a ritmo acelerado, transformándose en unos de los mercados más relevantes, por lo que su protección se ha hecho indispensable para salvaguardar los derechos e intereses de sus creadores.[41].

Definición y Naturaleza

Para entender cómo el derecho de autor protege esta clase de obra, es necesario tener un conocimiento técnico general del concepto de *software*. Para ello es importante señalar que los programas informáticos están

[41] GUERRERO G., Aspectos Globales de la Patentabilidad de las Invenciones Implementadas por Ordenador, Estado actual y Nuevas Perspectivas, p., 117, [en línea] http://revistas.uexternado.edu.co/index.php/propin/article/view/904/858

compuestos por un conjunto de instrucciones que ejecuta un ordenador para poder realizar una función específica. Normalmente los programadores escriben en un lenguaje que los humanos pueden entender y que posteriormente es traducido a secuencias de unos (1) y ceros (0), que es el único lenguaje que las máquinas entienden[42].

Estos se crean mediante lenguajes de programación, los cuales forman parte esencial, pues son éstos los que permiten que el desarrollador le dé instrucciones a la máquina. Estos lenguajes de programación, al igual que el leguaje natural, están conformados por un conjunto de símbolos (letras, números, espacios, etc.) y como éstos se ordenan para generar un significado. Es así que, por ejemplo, en el mundo de la programación existen diversos lenguajes como el Java, C, Python, etc., y en el lenguaje natural tenemos el inglés, castellano, mandarín, etc.

Los autores BROCCA y CASAMIQUELA señalan que el *software* es "la secuencia coherente de instrucciones que permite a la maquina efectuar una

[42] CULEBRO J., GÓMEZ H., y TORRES S., S. Software Libre vs Software Propietario, [en línea] http://www.rebelion.org/docs/32693.pdf

determinada tarea. El poner en funcionamiento un programa se la denomina ejecución o utilización del programa." Agregan dichos autores que, para un programa pueda operar adecuadamente, éstos requieren medios que permitan su funcionamiento, tales como las estructuras de datos y la documentación, que describe la operación de los mismos.[43]

Asimismo, precisan que el término *software* según el *Institute of Electrical and Electronics Engineers* (IEEE), es la parte de un sistema constituido por programas, procedimientos, reglas informáticas, documentación y datos asociados, necesarios para ejecutar funciones.[44]

Estos programas son escritos mediante algoritmos, los cuales -de manera simple- podemos decir que son una estructura, paso a paso, de cómo un problema dado debe ser resuelto de forma coherente, eficaz y simple[45]. En

[43] BROCCA J. y CASAMIQUELA R., Las Licencias de Software desde la Perspectiva del Usuario Final, Revista Pilquen Sección Ciencias Sociales, año VII No. 7, 2005, [en línea] http://www.scielo.org.ar/pdf/spilquen/n7/n7a12.pdf
[44] Ob. cit. ut supra.
[45] ARIAS A., y otro. Curso de Programación y Análisis de Software, 2da Edición, p. 48 [en línea]
https://books.google.cl/books?id=2Wj0DAAAQBAJ&pg=PA52&dq=software+instrucciones&hl=es-419&sa=X&ved=0ahUKEwjz4Pe-gonVAhWHlJAKHfZtCyo4ChDoAQhCMAY#v=onepage&q=software%20instrucciones&f=false

efecto, como indica GOTT, tenemos que un programa informático se construye en varios pasos. El programador creará un esquema del programa propuesto en forma de diagrama de flujo que desglosará subrutinas o módulos cuya disposición determinará la eficiencia del programa y, una vez que se ha logrado un diseño detallado, comenzará otro paso conocido como codificación. En la codificación, el programador redactará el programa en "código fuente" que, como se indicó arriba, puede estar en uno de varios lenguajes de programación y éste a su vez se transformará entonces en "código objeto" por un compilador.[46]

Un ejemplo para entender el punto de las instrucciones para quienes no manejamos un lenguaje muy técnico sobre el *software*, es el que el autor ARIAS[47] da al comparar a los programadores como cocineros, señalando que, si estos cocinaran en lugar de procesar datos, esta sería la receta para crear un plato de programa informático:

[46] GOTT J., Lotus Development Corporation v. Borland International: The United States Court of Appeals for The First Circuit Takes a Step Backward for The Copyright Protection of Computer Programs, p. 1355, [en línea] http://dspace.creighton.edu:8080/xmlui/bitstream/handle/10504/40205/49_30CreightonLRev1349(1996-1997).pdf?sequence=1

[47] Ob. citada, p. 22

PROGRAMA FREÍR_ HUEVO
RESERVAR HUEVO, PAN, SAL Y MANTEQUILLA;
USAR COCINA;
COLOCAR SARTEN EN COCINA;
PONER MANTEQUILLA EN LA SARTÉN;
ENCENDER LA COCINA;
ESPERAR QUE LA COCINA SE CALIENTE;
ROMPER EL HUEVO; DERRAMAR EL HUEVO EN LA SARTEN;
PONER SAL EN EL HUEVO;
ESPERAR QUE EL HUEVO SE FRÍA;
APAGAR LA COCINA

Se observa entonces que, el *software* es un bien intangible que se encuentra construido por un conjunto de instrucciones que se expresan inicialmente en código fuente textos (líneas de instrucciones en lenguaje de computadora/programación que se orienta a lograr un programa entendible y fácil de modificar por el ser humano).

A su vez, estas instrucciones se compilan en un código objeto (instrucciones legibles solo por la máquina donde se instala y pierde la capacidad de ser legible directamente por el ser humano) lo cual hace que se produzca un determinado resultado.

Las instrucciones se efectúan mediante diversos lenguajes de programación, que tendrán por objetivo hacer posible que los programadores escriban y/o desarrollen sus programas. Tales lenguajes pueden ser de bajo nivel, intermedios, o de alto nivel.

Cuando hablamos de los primeros, nos estamos refiriendo a aquellos que se alejan en mayor medida al lenguaje del ser humano; tenemos el de nivel intermedio, como por ejemplo son los lenguajes Java, y los de alto nivel, que son muy parecidos al idioma humano, siendo un ejemplo de ello el lenguaje denominado Basic.

En resumen, el *software* es cualquier tipo de programa que es creado mediante un lenguaje de programación y ejecutado por el *hardware* o dispositivo electrónico, el cual expresa adecuadamente las instrucciones que se le dará a la computadora por varias etapas para que se ejecuten funciones.

Etapas de desarrollo

La autora GONZÁLEZ sostiene que para crear un *software* se requiere principalmente de dos etapas: la definición y el desarrollo.

La primera etapa consiste en plantear el problema, es decir, se identificarán los requisitos claves del sistema y del *software*. Respecto al desarrollo, éste consiste en el cómo, o sea, la manera según la cual el programador va a definir cómo ha de diseñarse el programa (las estructuras de datos, las interfaces, los procedimientos, etc.), cómo ha de traducirse el diseño en un lenguaje comprensible por el ser humano (código fuente) y en un lenguaje comprensible por la máquina (código objeto), para luego realizarse la prueba del *software* y la corrección de errores[48].

Hay una tercera etapa que es posterior a la programación del *software* pero que pertenece a la vida útil de éste, como es la fase de mantenimiento. En esta etapa, el programador adapta el programa a las necesidades del usuario y a los avances que se produzcan en el entorno del *software* (cambios correctivos, adaptaciones y mejoras).[49]

[48] GONZÁLEZ B., 2013, El Debate Sobre La Interoperabilidad Informática en El Derecho De Autor Comunitario, p. 58, Tesis doctoral, mención europea, [en línea] file:///C:/Users/Neida%20Zambrano/Downloads/Dialnet-ElDebateSobreLaInteroperabilidadInformaticaEnElDer-46836%20(6).pdf
[49] Ob. citada *ut supra*

Clasificación

El autor NORTON[50] clasifica a los programas informáticos en *software* de sistema y de aplicaciones. Esta clasificación se basa en el objetivo del programa, es decir, la finalidad del uso.

Dicho autor indica que un programa informático equivale, pudiéramos decir, al cerebro que le permite a una computadora realizar una tarea específica, que consiste en instrucciones que dirigen a una computadora para que realice funciones preestablecidas y concretas.

Respecto a la primera clasificación, esto es, el *software* de sistema, el referido autor señala que el mismo refiere al "conjunto del sistema de programas generalizados que administran los recursos de la computadora, como el procesador central, los enlaces de comunicación y los dispositivos

[50] NORTON P., Introducción a la computación, 6ta. Edición, [en línea] https://alkedua.files.wordpress.com/2015/07/218982559-introduccion-a-la-computacion-peter-norton.pdf

periféricos”[51,] es decir, aquellos que se refieren al conjunto de programas que tienen como objetivo controlar el *hardware* de la computadora y que puede dar soporte a otros programas.

Dentro de la clasificación de los programas de sistema, tenemos el Sistema Operativo (SO), el cual se puede definir como un conjunto de programas esenciales para cualquier sistema de computación (el cual media entre el *software* de aplicación y el *hardware*)[52], indicándole a una computadora cómo debe utilizar y/o administrar los recursos. Un sistema operativo está formado por varios componentes que necesitan funcionar entre ellos, de manera que su diseño estará condicionado por las especificaciones de la interfaz que hacen posible que todos los componentes trabajen en la red.[53]

Según HEREDERO, puede señalarse que este tipo de sistema realiza un conjunto de funciones básicas, entre las que se encuentran el suministro de la

[13]AMAYA, J., Sistema de Información Gerenciales, pág. 31, [en línea] https://books.google.cl/books?id=nZzFAQAAQBAJ&pg=PA31&dq=tipos+de+software&hl=es-419&sa=X&ved=0ahUKEwiJqayEj8HUAhUBgJAKHQjXBLUQ6AEIODAE#v=onepage&q=tipos%20de%20software&f=false

[52] Ob. citada *ut supra*, p. 32.

[53] GONZÁLEZ B., ob. citada p. 42.

interfaz al usuario, permitiendo que éste se comunique con el computador por medio de interfaces basadas en comandos, interfaces que utilizan menús e interfaces gráficas de usuario[54].

El referido autor da como ejemplos al sistema operativo Android, el cual administra los recursos del *hardware* como la CPU, memoria, dispositivos de almacenamiento secundario y periféricos de entrada y de salida; administración de archivos, mediante los cuales controla la creación, borrado, copiado y acceso de archivos de datos y de programas; administración de la información sobre los programas y procesos que se ejecutan.

Por otra parte, los *softwares* de aplicación son descritos por AMAYA, como aquellos que se crean o desarrollan para los usuarios con el fin de aplicar el computador a una tarea específica[55]. Esto abarca como por ejemplo los procesadores de texto, las bibliotecas de programas para resolver operaciones

[54] HEREDERO C. y otros, Organización y transformación de los sistemas de información en la empresa: [en línea] https://books.google.cl/books?id=2pqwKkqxxosC&pg=PA93&dq=El+Sistema+Operativo+(SO)&hl=es-419&sa=X&ved=0ahUKEwiOsbO2mq_UAhVJhZAKHSY7A0cQ6AEIRDAG#v=onepage&q=El%20Sistema%20Operativo%20(SO)&f=false
[55] Ob. citada, p. 32.

de cálculo o estadísticas, los sistemas de bases de datos, programas para gráficos y dibujos, gestores de correo electrónico, navegadores de Internet, videojuegos, entre otros.[56]

Un ejemplo de programas de aplicación son las interfaces, las cuales en general permiten a dos o más programas comunicarse entre sí, pudiendo clasificarse en: interfaces de programas, de usuarios, de datos y de comunicación.

Las interfaces en general consisten en la conexión entre dos programas, o entre dos partes de un mismo programa a través de las cuales se transmite datos/información.

Estas son necesarias, por cuanto sin ellas dos sistemas no pudieran comunicarse, sea entre dos sistemas informáticos, dos partes de que conforman un sistema informático o un sistema o programa con el *hardware.*

[56] Ob. citada, p. 50.

Interfaces de programas de aplicación y la interoperabilidad

Las interfaces de programas de aplicación, (*Application Programming Interface* o sus siglas en ingles API) son un tipo de las interfaces, las cuales se utilizan en el mundo de la programación y se pueden definir como el conjunto de "estándares del programa que incluye procesos de interrupción, llamadas, funciones y formatos de datos que pueden ser utilizados por un programa para acceder a servicios de red, dispositivos o sistemas operativos*".*[57]

Éstas permiten desarrollar programas de aplicación en el entorno del sistema operativo, de allí que sean esenciales, pudiendo interoperar interfaces de sistema de aplicación a sistema operativo o viceversa.[58]

Se tiene entonces que las interfaces o APIs son utilizadas por programadores de lenguaje de programación para que sus aplicaciones interactúen con otros programas preestablecidos, y el rol que desempeñan en el desarrollo del *software* ha sido expresado de diversas formas.

[57] Ob. citada ut supra
[58] Ob. citada ut supra, p.105

Una API define cuáles son las funcionalidades y cómo se utilizan, mientras que la implementación de una API establece cómo se logran las funcionalidades a través de la interoperabilidad. Un concepto básico de la informática es que las APIs pueden tener múltiples implementaciones, lo que proporciona la posibilidad de sustituir implementaciones nuevas y mejoradas.[59]

En este sentido, se puede decir que las APIs están constituidas por paquetes, estos a su vez por clases y éstas por métodos. Los métodos son las funciones básicas del programa que permiten ejecutar una tarea específica, estos se agrupan por su similitud en su funcionalidad, es decir, por cuanto tienen funcionalidad en común o algo similar en "clases". A su vez las clases se agrupan paquetes.

Un ejemplo de esto, sería diversos "métodos" de sumar y restar dos números enteros que se agrupan en la "clase" "entero" y ésta "clase" pertenece al paquete "matemática" junto con otras clases como "reales".

[59] Brief of Computer Scientists as Amici Curiae in Support of Defendant-Appellee, [en línea] https://www.eff.org/es/document/computer-scientists-amicus-brief

Las APIs se pueden utilizar a nivel bajo para usar el *hardware* (componentes físicos) como el conector de una cámara, y a nivel alto o lógico, como las interfaces de datos entre dos procesadores de textos, como por ejemplo cuando se convierte un documento de PDF a Word, pero es necesario para que dos programas interceden entre sí, que éstos se adhieran a las mismas especificaciones de la interfaz. Es importante también dejar claro que estas especificaciones son la descripción o reglas del funcionamiento de la interfaz[60].

Respecto a la utilidad de las APIs en programas informáticos, CANALES señala que éstas no pueden ser consideradas como obras creativas, sino funcionales, pues una de las finalidades de éstas es alcanzar una estandarización que permita una amplia interoperabilidad[61].

[60] Ob citada, p. 96.
[61] CANALES M., "Oracle versus Google: Protección de derecho de autor sobre elementos funcionales de programas computacionales que ponen en riesgo la interoperabilidad y la innovación", Revista Chilena de Derecho y Tecnología, Universidad de Chile. [en línea] http://www.rchdt.uchile.cl/index.php/RCHDT/article/viewFile/37361/40376

Las APIs por definición no pueden ser consideradas como creativas porque se hacen para que dos o más sistemas puedan comunicarse entre sí. Sustenta el comentario antes señalado citando al autor CURTIS, en lo que respecta a que las interfaces son una preocupación para los desarrolladores de *software*, por cuanto los estándares están siendo construidos en torno a ellas. Este autor señala como ejemplo, cuando un gobierno exige estándares para el acceso a computadores de personas con discapacidad, o los estándares que observamos en beneficio de los consumidores en los equipos para teléfono móvil, donde se encuentran presente las interfaces[62].

Algo parecido a lo anterior señalan FARRELL y SALONER cuando dicen que la compatibilidad de los programas informáticos son un aspecto crucial en muchas industrias, especialmente la tecnología, por cuanto ésta puede ser alcanzada a través de la estandarización, es decir, para hacer ciertas cosas de manera uniforme, lo cual ocurre por ejemplo, "cuando los fabricantes de computadores usan las mismas interfaces para la conexión de complementos, cuando las cámaras son diseñadas para usar una película común de formato

[62] Ob. citada ut supra

de 35 mm, o cuando los diseñadores de programas computacionales adoptan una interfaz de usuario común"[63].

Entonces, las APIs son fundamentales dentro de la interoperabilidad de los programas informáticos, ya que con éstos tienen la posibilidad de compartir una interfaz de usuario común, lo cual sin duda alguna beneficia a los consumidores finales que adquieren o necesitan de la tecnología, como por ejemplo, que podamos utilizar la aplicación WhatsApp en nuestro computador (Windows, Linux, Mac) o que podamos instalarla en nuestro teléfono (Android, iOS).

Para que dos programas sean interoperables o compatibles deben adherirse exactamente a las mismas especificaciones de la interfaz del primer programa, la cual va a establecer las reglas acerca de cómo otros programas van a trabajar juntos para efectuar tareas específicas, deben ser idénticas/compatibles.

[63] FARRELL J. y otro., *The Journal of Industrial Economics* 1992), Introducción. [en línea]
http://www.jstor.org/stable/2950625?seq=1#fndtn-page_scan_tab_contents

Es por lo anterior, que podemos decir que la interoperabilidad de un programa informático, tiene gran importancia dentro del mundo globalizado. Es evidente que, unas de las funciones primordiales de las APIs es la necesidad de compartir datos y/o intercambiarse información, otro ejemplo de ello sería la posibilidad de acceder con el uso de un teléfono con un sistema operativo Android a Internet.

Así pues, para permitir que un programa interactúe con otros programas, un programador debe cerciorarse de que envía y recibe señales en la manera requerida por esos programas. Una de las tareas críticamente importantes del desarrollo del *software* está diseñando este flujo de la información para permitir interoperación.

El software desde una perspectiva legal

Marco Normativo Internacional

La propiedad intelectual, según la definición de la Organización Mundial de la Propiedad Intelectual (OMPI), es entendida como toda creación de la mente humana, entre las que se encuentran las invenciones, obras literarias y

artísticas, símbolos, nombres, imágenes, dibujos y modelos y que corresponde al autor por el solo hecho de crearla. Así, se distinguen: las patentes, los derechos de autor o *copyright*, los derechos conexos, los diseños industriales, indicaciones geográficas y las marcas.

La mencionada Organización Mundial se creó en el año 1967, como organismo de la Organización de las Naciones Unidas especializado en materia de propiedad intelectual. Actualmente cuenta con 188 Estados Miembros y tiene dentro de sus funciones administrar los tratados internacionales en la materia (ej.: Convenio de Paris y Berna), la protección de la propiedad intelectual, la resolución de conflictos que se planteen respecto a estos asuntos, la organización de programas de cooperación, apoyo a los gobiernos, las empresas y la sociedad en todas las actividades concerniente con la propiedad intelectual que tengan como objetivo el desarrollo económico, social y cultural.

Existe un conjunto de acuerdos y tratados internacionales que recogen la importancia de la propiedad intelectual bajo un contexto inventivo, científico y tecnológico. Entre otros, se pueden señalar el Convenio de Paris, el

Convenio de la Organización Mundial de Comercio, el Convenio de Berna, Convenio de Roma y el Convenio Centroamericano para la Protección de la Propiedad Intelectual.

En el año 1883, se redactó el Convenio de París para la Protección de la Propiedad Intelectual que fue el primer tratado internacional para facilitar que las personas de un país tuvieran protección en sus creaciones intelectuales en otros países. Fue aquí donde se crearon los derechos de propiedad intelectual para las patentes, marcas y diseños industriales.

Posteriormente, en el año 1886 se celebró el Convenio de Berna para la Protección de las Obras Literarias y Artísticas, cuyo claro objetivo era que las personas de los Estados tuvieran una protección internacional para sus obras creativas, siendo adoptado por más de 164 países, incluyendo a los Estados Unidos de América desde 1989. Ha sufrido múltiples revisiones y enmiendas, teniendo como principio el trato nacional, la protección automática, y la "independencia" de la protección.

Otras normativas de importancia internacional son los Acuerdos sobre Determinados Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio, en los cuales se acordaron aspectos que implican, entre otras cosas, extender los principios de Berna a los Estados signatarios del Acuerdo de la Organización Mundial del Comercio.

En cuanto al derecho de autor, se protegen todas las obras del ingenio de carácter original, ya sean literarias, artísticas o científicas, pero no las ideas. Cuando se crea una obra innatamente vienen derechos que van a proteger tanto a la obra como al autor y que son inmediatos desde el momento de la creación, no necesita de ninguna formalidad, es decir, no requieren de la inscripción en un registro o el depósito de copias. Los derechos de autor nacen con la creación de la obra y sólo el autor o aquellos cuyos derechos derivan del autor pueden reclamar propiedad.

Este derecho de autor a su vez se divide en derechos morales, conexos y patrimoniales. En cuanto al derecho moral, este no es no es apreciable en dinero, pero que permite al autor defender su nombre y su obra contra publicaciones indeseadas, alteraciones o usurpaciones y se conforma de

cuatro atributos: derecho a la divulgación, el derecho al respeto del nombre, a la obra y el derecho al arrepentimiento o retractación. Por otro lado, observamos al derecho patrimonial, el cual permite de manera exclusiva la explotación de la obra y va a existir durante toda la vida del autor y años subsiguientes después de la muerte según la legislación de cada país y por último, los derechos conexos, mediante los cuales se van a proteger a personas distintas al autor, como pueden ser los artistas, intérpretes, traductores, editores, productores, etc.[64]

Catalogado como obra, el *software* se encuentra regulado por el derecho de autor en la mayoría de los países, lo que le confiere a su titular un derecho exclusivo sobre el mismo.

El Acuerdo de 1994 sobre los Aspectos de los Derechos de la Propiedad Intelectual relacionados con el Comercio, conocido como Acuerdo sobre los ADPIC o, por sus siglas en inglés, TRIPs, dictado en el marco del Convenio de Berna, es uno de los instrumentos normativos declarativo de protección

[64] COLOMBET, C., Grandes principios del derecho de autor y los derechos conexos en el mundo, Ediciones UNESCO/CINDOC, Madrid, 1997, p45, 95.

de los programas informáticos como obras literarias. En su artículo 10 declara que los programas de ordenador, sean programas fuente o programas objeto, serán protegidos como obras literarias en virtud del mencionado Convenio.[65]

Un año más tarde (1996) se lleva a cabo la Conferencia Diplomática de la Organización Mundial de la Propiedad Intelectual, donde producto de la gran demanda de problemas por la proliferación de las tecnologías digitales, se dispuso que los programas informáticos son creaciones literarias, suscribiéndose dos tratados internacionales en el seno de dicha conferencia, conocidos como "Tratados Internet" que complementan el Convenio de Berna (Convenio de Berna, 1971, enmendado en 1979). Estos son el Tratado de la OMPI sobre Derecho de Autor y el Tratado de la OMPI sobre Interpretación y Ejecución y Fonogramas.

Es así que, el Convenio de Berna en su artículo 9.2 permite establecer limitaciones al derecho exclusivo de reproducción, de manera excepcional,

[65] Oficina Internacional de la OMPI, La Protección Internacional del Derecho de Autor y de los Derechos Conexos, [en línea] http://www.wipo.int/export/sites/www/copyright/es/activities/pdf/international_protection.pdf

cuando no atenten contra la explotación normal de la obra ni causen un perjuicio injustificado a los intereses del autor, lo que ha sido denominado como "regla de los tres pasos". Ello resulta aplicable no solamente para el derecho de reproducción, sino también para los demás derechos protegidos según el Acuerdo sobre los ADPIC.[66]

Otro instrumento internacional es el Tratado de la OMPI sobre Derecho de Autor, el cual en su artículo 4 señala que "los programas de ordenador están protegidos como obras literarias en el marco de lo dispuesto en el Artículo 2 del Convenio de Berna del Convenio de Berna. Dicha protección se aplica a los programas de ordenador, cualquiera que sea su modo o forma de expresión."[67]

Se observa también que la Comunidad Andina de Naciones, anteriormente Pacto Andino, también regula en materia de derecho de autor. Así vemos que

[66] ORGANIZACIÓN MUNDIAL DE LA PROPIEDAD INTELECTUAL (OMPI), El derecho de autor y los derechos conexos en el entorno digital, [en línea] http://www.wipo.int/edocs/mdocs/lac/es/ompi_sgae_da_asu_05/ompi_sgae_da_asu_05_2.pdf

[67] TRATADO DE LA OMPI SOBRE DERECHO DE AUTOR, [en línea] http://www.wipo.int/treaties/es/text.jsp?file_id=295167#P74_6370

mediante la resolución 351 del año 1993 se aprueba un Régimen Común de Derecho de Autor y Derechos Conexos para los países miembros de la Comunidad, donde regula una adecuada y efectiva protección a los autores y demás titulares de derechos, sobre las obras de ingenio, en el campo literario, artístico o científico, cualquiera que sea el género o forma de expresión y sin importar el mérito literario o artístico ni su destino.

Se evidencia entonces que, internacionalmente la protección del *Software* se encuentra regulada dentro del derecho de propiedad intelectual, siendo clasificada por la mayoría de la doctrina y legislación nacional e internacional dentro de la categoría del derecho de autor.

La protección del *software* y la propiedad intelectual

Entonces, se desprende que el derecho de autor o *copyright* le da al creador de la obra la potestad de controlar su uso respecto a terceras personas, por lo que, en términos generales, un *software* en principio no puede ser duplicado, distribuido, ni objeto de apropiación por otros sin el permiso del creador.

Por ejemplo, si se utiliza una parte de la obra, los caracteres o algunos elementos de una obra protegida por derechos de autor para desarrollar un trabajo propio, sin la autorización del creador (propietario de la obra), probablemente se infringiría los derechos de propiedad intelectual de ese creador. Sin embargo, estos derechos exclusivos de autores están limitados de varias maneras, como lo veremos más adelante.

La OMPI ha señalado que el *software* no es solamente una expresión literaria y que las líneas del código tienen una función que no depende de su construcción gramatical, es así que el código fuente de un programa informático puede ser completamente diferente al de otro programa, y con todo, cumplir la misma función a la hora de dar lugar a un conjunto similar de instrucciones que producen un resultado parecido.[68]

Una de las justificaciones que se ha dado para que un programa informático pueda estar protegido por el derecho de autor, es que éste se realiza a través de un lenguaje que entiende el ser humano donde da instrucciones a una

[68] ORGANIZACIÓN MUNDIAL DE PROPIEDAD INTELECTUAL (OMPI), "El Seminario Regional de la OMPI para América Latina y el Caribe sobre Propiedad Intelectual y Software en el siglo XXI, 2008".

máquina, a lo cual se denominan código. Todo el conjunto de instrucciones se les denominan código fuente.

Se ha dicho que el asimilar el *software* como obra literaria dentro de la protección de los derechos de autor, "donde la obra es expresión de la personalidad de su autor", es un problema, pues si bien un programa informático es una forma de expresión, esto no quiere decir que sea un reflejo de la personalidad de su autor, sino que su fin es para ejecutar tareas específicas, por lo que siempre será funcional.[69]

Compartiendo lo que señala SAMUELSON, el *software* posee características que lo distinguen como una clase de obras únicas dentro el sistema del derecho de autor (tanto en el *copyright* como en el "*droit de auteur"* continental). Entre tales características se encuentran la funcionalidad y el que no tenga un fin comunicativo, hace que haya una fricción con algunos de

[69] BEGOÑA O., El Debate Sobre La Interoperabilidad Informática en el Derecho de Autor Comunitario, tesis doctoral, p. 65, [en línea] file:///Users/neidazambrano/Downloads/Dialnet-ElDebateSobreLaInteroperabilidadInformaticaEnElDer-46836%20(7).pdf

requisitos que debe reunir una obra para sea considerada como objeto protegible, como son la originalidad y la expresión de la obra.[70]

Entonces, ciertamente el software puede contener expresión creativa, al mismo tiempo, tener un propósito esencialmente funcional, como lo es instruir a una computadora para lograr el resultado deseado.

En tal sentido, se puede decir que diversos aspectos del software sirven como "artículos utilitarios que contienen muchos elementos lógicos, estructurales y visuales que van a hacer netamente funcionales en virtud de factores externos tales como requisitos de compatibilidad y demandas de la industria. (citando Comput. Assocs Int'l, Inc. v. Altai, Inc). Así, mientras que los programas informáticos se equiparan a las obras literarias y tienen protección por el derecho, su naturaleza híbrida requiere que esas protecciones se equilibren con la necesidad de fomentar y proteger la innovación.[71]

[70] SAMUELSON P., "Three Fundamental Flaws in CAFC's Oracle v Google Decision" University of California, Berkeley, European Intellectual Property Review, p.1.
[71] BRIEF *for Microsoft Corp., Red Hat, Inc., and Hewlett Packard Enterprise Co. as Amici curiae in support of Defendant-cross-appellant and affirmance*, p.10 [en línea] https://www.eff.org/es/document/microsoft-red-hat-and-hp-amicus-brief

Cuando se habla de funcionalidad de un programa, se hace referencia a todo lo que ocurre en la ejecución de un computador. Existe consenso en cuanto a qué es lo que se protege con el derecho de autor o mediante el *Copyright Act*, pues se sostiene que se protegen las expresiones mediante las cuales se materializa el *software*, es decir, el código fuente y no los demás elementos funcionales, métodos o procedimientos que conforman también a un programa informático[72].

No obstante, tanto en la doctrina como en la jurisprudencia, especialmente la norteamericana, han surgido discusiones respecto a si estos últimos elementos también deben considerarse protegidos por el *copyright*, tomándose en cuenta la funcionalidad que los caracteriza.

En el caso Lotus v. Borland, se señaló que "El programa computacional es un medio para causar que algo suceda; tiene una utilidad mecánica, un rol instrumental, completando el trabajo del mundo. Conceder protección [...] puede tener algunas de las consecuencias de la protección otorgada por las

[72] Ob cit. P. 64.

patentes limitando la posibilidad de otras personas de ejecutar una tarea en la manera más eficiente".[73]

Una forma que se puede utilizar para verificar qué parte de un programa informático puede estar o no protegido por el derecho de autor y que ha sido utilizado por tribunales estadounidense, es aplicar lo siguiente: "Paso uno: descomponer el programa presuntamente infringido en sus partes estructurales constitutivas. Paso dos: sacar todo el material no protegible, incluyendo ideas y expresión que es necesariamente incidental a esas ideas; y Paso tres: El tribunal compara la expresión creativa restante con el programa presuntamente infractor".[74]

Ahora bien, en el ámbito específico del *software* se puede observar, por ejemplo, como diversos programas informáticos utilizan otros programas o grupos de expresiones funcionales ajenos, entre ellos, las interfaces (API),

[73] CANALES, M. Oracle versus Google: Protección de derecho de autor sobre elementos funcionales de programas computacionales que ponen en riesgo la interoperabilidad y la innovación. Revista Chilena de Derecho y Tecnología, Universidad de Chile, ISSN 0719-2584, VOL. 4 NÚM. 2 (2015), PÁGS. 179-213 • DOI 10.5354/0719-2584.2015.37361.
[74] North Carolina Journal of Law & Technology Volúme 18, issue on.: december 2016 1 Oracle v. google: SETTING A STANDARD OR HANDICAPPING AN INDUSTRY? Tyler J. Demasky [en línea]
file:///C:/Users/Neida%20Zambrano/Downloads/Demasky_Final%20(1).pdf

para darles distintos usos, o incluso para la creación de un programa nuevo, y ello ocurre generalmente sin la autorización del titular de los derechos intelectuales de la obra original.

Cuando se habla de la parte funcional de las interfaces, por ejemplo, el *issues* que se plantea es, si estas pueden entrar dentro de la calificación de ideas o métodos de operación, por lo que pudiera considerarse obras no protegidas por el derecho de autor, o al revés que éstas sean parte de la expresión del programa y por tanto, objeto de protección.

Es importante recalcar que, respecto a la parte funcional de las interfaces, la postura más fuerte de la jurisprudencia y doctrina es que éstas no pueden ser protegidas por el *copyright* o derecho de autor, aun aceptado que son expresiones y que cumplen con el requisito de originalidad, pues al fin y al cabo son métodos de operación, lo cual como se mencionó anteriormente, no deja de tener protección por medio de otra figura como es la patente, pero no por medio del *copyright*.

Debemos señalar que las protecciones de las SSO han estado en un punto gris dentro de la materia del c*opyright*. Es así que, a mediados y finales de los años 80 del siglo pasado comenzó a argumentarse que las SSO de las APIs podrían ser protegidas por *copyright*[75] y es por la década de los 90 que los tribunales decidieron que eran elementos de programas no protegidos, por lo que algunas empresas optaron por la opción de patentar los diseños de interfaz.

Este punto gris podemos observarlo en dos sentencias recaídas en un mismo caso (Oracle v. Google), donde, por un lado, el Tribunal del Noveno Circuito ratificó la postura mayoritaria que la SSO de la API es un método de operación conforme a lo previsto en la sección 102 (b) del *Copyright Act*, y no es susceptible de protección bajo este contexto, por cuanto se caracteriza funcionalmente para la interoperabilidad de los programas.

[75] Ver decisión Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., La Corte de Apelaciones de los Estados Unidos para el Tercer Circuito llegó a la conclusión de que la protección del derecho de autor de los programas informáticos puede extenderse más allá del código literal de los programas a su estructura, secuencia y organización. En casos posteriores, varias cortes rechazaron el análisis de la infracción de Whelan, como por ejemplo en Altai/Sega.

Por otro lado, respecto a la postura, que las SSO de las interfaces si están protegidas por el derecho de autor, tenemos la decisión de la Corte Federal que revocó la sentencia arriba mencionada y fijó lo contrario, al indicar que las APIs sí se encuentran inmersas dentro de la protección del *copyright*.

Lo anterior, como lo ha señalado una gran parte de la doctrina, ha sido enormemente criticado pues afecta el mercado de la tecnología. Se sostiene que las SSO pueden ser consideradas como parte funcional del *software* y de utilidad necesaria y pueden ser utilizadas sin autorización por parte de los programadores en la creación de otros programas.

No obstante, tomando en cuenta la última decisión en esta materia sobre las APIs, que rige en materia de interfaces en el derecho estadounidense, esto es, que son protegibles bajo el *Copyright Act.*, a los fines del presente trabajo será lo que se tomará en consideración, es decir, que la copia de las estructuras, secuencias y organización de las interfaces son protegibles y que lo que se debe de evaluar para considerar si hubo o no infracción de ésta protección es si fueron utilizadas de manera justa.

Limitaciones

Existen una serie de excepciones y limitaciones en función del interés social que las creaciones de diversa índole pudieran conlleva [76] . Como mencionamos arriba, a los fines de crear una nueva obra puede ser necesario utilizar otra obra que constituye propiedad intelectual de un tercero. Por ejemplo, los derechos de autor protegen la expresión del autor, pero no las ideas contenidas en esa expresión, lo cual protege el derecho del público a construir y utilizar cualquier idea que se incorpora a la expresión protegida por derechos de autor.

Es claro que, una de las características fundamentales del derecho de propiedad consiste en la exclusividad que se le otorga al titular sobre el bien del cual es propietario. Sin embargo, podemos estimar que se puede utilizar bienes intelectuales para crear una obra, incluso sin pedir permiso al titular de los derechos intelectuales, como lo es por ejemplo, mediando el uso justo de ésta.

[76] GRIJALVA A., Internet y derechos de autor, [en línea] http://www.flacso.org.ec/docs/sfintgrijalva.pdf

Es así que, por ejemplo, en el campo tecnológico el desarrollo de un *software* puede conllevar que los ingenieros o especialistas utilicen porciones de programas/obras que son propiedad intelectual de otras personas. En estos casos, en principio, el "nuevo creador" pudiera incurrir en violación al derecho de propiedad intelectual de esos titulares del programa informático, incluyendo el código fuente, como otros elementales funcionales, como los son las interfaces utilizadas en la programación.

Si bien la normativa de derechos de autor proporciona un incentivo para la expresión y la innovación, permitiendo que un creador controle y se beneficie de su trabajo mediante el derecho exclusivo de reproducir la obra, crear obras derivadas, distribuirlas o mostrarlas, observamos que en el sistema anglosajón, Estados Unidos ha adoptado un sistema abierto bajo la doctrina del "*fair use"*, excepción más conocida del *copyright*, que habilita el uso ilimitado por parte de terceros del material protegido por derecho de autor sin la necesidad de requerir permiso a los titulares de tal derecho.

En la mayoría del sistema continental, por el contrario, al anterior sistema las limitantes al derecho de autor se encuentran enumeradas taxativamente en la

normativa aplicable y establecen las circunstancias que admiten el uso de las obras sin autorización del autor, como por ejemplo en España, Italia, Francia, y en países latinoamericanos como Chile, Argentina o Colombia.

En Chile, por ejemplo, la Ley otorga excepciones y limitaciones al derecho de autor respecto a los programas informáticos. La Ley Nº 17.336[77], con su reforma N° 20.435[78], en su artículo 71 Ñ letra b), prevé que "las siguientes actividades relativas a programas computacionales están permitidas, sin que se requiera autorización del autor o titular ni pago de remuneración alguna: b) Las actividades de ingeniería inversa sobre una copia obtenida legalmente de un programa computacional que se realicen con el único propósito de lograr la compatibilidad operativa entre programas computacionales o para fines de investigación y desarrollo. La información así obtenida no podrá utilizarse para producir o comercializar un programa computacional similar que atente contra la presente ley o para cualquier otro acto que infrinja los derechos de autor".

[77]CHILE, Biblioteca del Congreso Nacional (BCN), Ley sobre de Propiedad Intelectual, [en línea] https://www.leychile.cl/Navegar?idNorma=28933
[78]CHILE, Biblioteca del Congreso Nacional (BCN), Ley sobre de Propiedad Intelectual (Reforma) https://www.leychile.cl/Navegar?idNorma=1012827

Se observa cómo se fija expresamente cuáles son los supuestos donde una persona puede utilizar un programa sin violentar el derecho de propiedad intelectual de un tercero en Chile, contrario a lo que pasa en Estados Unidos, donde no se encuentra expresamente cuáles son las limitantes que tiene la propiedad intelectual de una obra, sino que va a depender del análisis de caso a caso por parte del juez en base a lo que demuestren las partes respecto a lo justo que fue usar o no una obra protegida.

Definir cuál de las dos posiciones es la mejor dentro de ambos derechos ha sido objeto de discusión por parte de la doctrina en esta materia. Pero, por ejemplo, hay una gran parte que se inclina a defender que indiscutiblemente la aplicación del *fair use* tanto en el derecho europeo como en Latinoamérica, debe ser acogida, por la gran flexibilidad que aporta esta teoría dentro del campo del derecho de propiedad intelectual[79].

[79] Citando a Kenneth Crews como defensor de la doctrina del *fair use*, en su artículo "El papel del juez frente a las limitaciones y excepciones al derecho de autor en los países de tradición continental [en línea] http://propintel.uexternado.edu.co/el-papel-del-juez-frente-a-las-limitaciones-y-excepciones-al-derecho-de-autor-en-los-paises-de-tradicion-continental/

Se tiene que, una de las diferencias fundamentales en cuanto al tema de las limitaciones y excepciones al derecho de autor y conexos entre los países de tradición jurídica continental y los países de tradición anglosajona, es que en estos últimos las limitaciones o excepciones "no están sujetas a una estructura de numerus clausus (la utilización libre y gratuita de las obras protegidas se conoce como fair use)", como sí sucede en el continental, donde las éstas son objeto de una enumeración taxativa, en la que no hay lugar a la aplicación analógica o extensiva de las causales"[80].

Aquí, es importante señalar en cuanto a la tradición anglosajona[81] que, aunque por cada país que aplican este sistema tienen nombre distinto, el fin es similar, éste será aplicado por la razón y no se encuentran de forma taxativa.

[80] MONROY J., Organización Mundial de la Propiedad Intelectual (OMPI), Estudio sobre las Limitaciones o Excepciones al Derecho de Autor y los Derechos Conexos en Beneficio de las Actividades Educativas y de Investigación en América Latina y El Caribe, p. 35. [en línea]
http://www.wipo.int/edocs/mdocs/copyright/es/sccr_19/sccr_19_4.pdf

[81] De manera de ejemplo podemos citar lo que establece GARCÍA (ob. citada) donde señala que "*El fair dealing* del Reino Unido es similar al *fair use* norteamericano. Pero que la diferencia radica en que mientras el *fair use* está establecido en el artículo 107 de la *Copyright Act, el fair dealing* ha sido enteramente de creación jurisprudencial, por lo que ambos criterios, representan categorías jurídicas indeterminadas que tienen que ser precisadas por el juez.

Podemos observar que las limitaciones y excepciones al derecho de autor de un programa informático, van a depender del sistema que se infringió. Bajo el sistema continental, los desarrolladores de *software* deberán estar más atentos a lo que establece la ley para saber qué pueden y de qué manera pueden reproducir o reimplantar un programa ya existente y protegido, lo contrario al sistema bajo el *fair use* donde los desarrolladores, sin pedir permiso o tener una autorización expresa, tienen más libertad de utilizar ampliamente programas de otro con solo considerar que la utilización debe ser de manera justa.

Entonces, se puede decir que el *fair use* se encuentra bajo un sistema abierto, donde los jueces aplicaran la razón y ciertos criterios establecidos en la ley, analizando caso a caso, para determinar cuáles obras fueron usadas de manera justa o no. Caso contrario al continental, el cual es un sistema cerrado, esto es, que las excepciones y limitaciones son enumeradas de forma taxativa y serán aplicadas por el juez de la misma manera a todos los casos, dependerá de cada legislación la lista de qué es lo que está permitido y no respecto a la reproducción o copia de un programa informático.

CAPÍTULO III

LA DOCTRINA DEL "*FAIR USE*"

El *Fair Use* como limitación y su Aplicabilidad a las Interfaces.

Definición y Alcance

Partiendo de lo expuesto, cabe preguntarse entonces, cuáles han sido los fundamentos y el alcance de la aplicación del principio del *fair use* sobre el derecho de autor o *copyright* según la jurisprudencia comparada y, en consecuencia, de qué manera incide en la protección jurídica del *software* en cuanto a su uso por parte de terceros.

Esta doctrina ha sido aplicada en muchas decisiones de los Estados Unidos conociendo de controversias en materias de programas informáticos. Así observamos que en casos emblemáticos en materia de propiedad intelectual de *software* esta doctrina ha sido protagonista a la hora de decidirse si el uso de unas interfaces (objeto de análisis de esas sentencias respecto si estas se encuentran o no protegidas por el derecho de autor) exceden o no las previsiones del *fair use* previsto en la *Copyright Act.*

En el derecho estadounidense esta doctrina se recogió en la sentencia recaída en el caso *Folsom v. Marsh* del año 1841, en la cual se establecieron los cuatro factores que años después conformaron la Sección 107 del *Copyright Act* norteamericano de 1976.

Es importante mencionar que ésta, por ser considerada tanto por la jurisprudencia como por la doctrina estadounidense como una regla equitativa de la razón, ha dejado claro que no es posible una definición generalmente aceptada, pues las cuestiones que se plantean deben decidirse caso a caso.

No obstante, aquí vamos a definirla como el privilegio que tienen otras personas distintas al propietario de utilizar el material protegido por derechos de autor de manera razonable, sin perjuicio del monopolio otorgado al propietario.[82]

[82]¿AGUIRRE C., Does the Fair Use Doctrine Constitute a Proper Balance Between the Appropriation of Characters?, University of Washington School of Law Seattle, Washington, p. 12.

Observamos que el Congreso de los Estados Unidos incorporó la defensa del uso justo en la *Copyright Act* de 1976, para proteger a los infractores de la responsabilidad en ciertas circunstancias limitadas, cuando ello no erosiona los derechos exclusivos y los beneficios previstos de la protección de derechos de autor.

La doctrina del "*fair use",* o uso justo en español, según GRIJALVA constituye el principal límite al derecho de autor en la legislación estadounidense, tratándose de un límite de carácter general, no ligado a objetivos concretos. Señala además este autor que, es de carácter absolutamente flexible, puesto que en cada caso concreto habrá que tener en cuenta factores predeterminados para ver si ese uso puede ser considerado "fair"-y por tanto permitido- o no[83].

En su sentido más general, puede decirse que un uso justo es cualquier copia de material con derechos de autor realizado para un propósito limitado y "transformador", como comentar una obra protegida por *copyright*. Tales

[83] FERNÁNDEZ J., El Futuro de los Límites a los Derechos de Autor que Benefician y Apoyan la Investigación, [en línea]
http://enancib.ibict.br/index.php/enancib/vienancib/paper/viewFile/1801/942

usos pueden hacerse sin el permiso del propietario de los derechos de autor. En otras palabras, el uso justo es una defensa contra una reclamación de infracción de derechos de autor. Si su uso califica como un uso justo, entonces no sería considerado una infracción[84].

Lo anterior, también lo han señalado los Tribunales estadounidenses. Por ejemplo, el Juez del Noveno Circuito de Estados Unidos señaló que, si bien unas de las funciones detrás del *Copyright Act*, es proteger las obras de la explotación por otros, no obstante, el derecho del *fair use* permite el uso de obras protegidas por derechos de autor por terceros sin el consentimiento de su titular, por cuanto uno de los fundamentos detrás de esta doctrina es fomentar y permitir el desarrollo de nuevas ideas que se basan en ideas anteriores, lo que proporciona un contrapeso a la política de derechos de autor para proteger las obras creativas[85].

[84]Stanford University Libraries, Copyright and fair use, [en línea] http://fairuse.stanford.edu/overview/fair-use/what-is-fair-use/

[85] In The United States District Court for The Northern District Of California Notice Of Final Charge To The Jury (Phase One) And Special Verdict Form, en el caso Oracle America INC v. Google INC., p. 9 [en línea] https://es.scribd.com/doc/313560685/Jury-Instruction-Oracle-v-Google

Se observa que la OMPI, definió esta doctrina señalando que "el uso leal está permitido para fines como la crítica, el comentario, noticias, informes o reportajes, docencias, estudios especiales o investigación. El uso leal habrá de determinarse teniendo presentes los factores previstos en el artículo 107 de la Copyright Act."[86]

Cuando se utiliza, reimplementa o se copia un programa informático ajeno, mediando el *fair use*, el nuevo programa que resulte del uso del original mezclado con elementos creativos de ésta, será considerado una obra nueva cuyo titular es únicamente quien transformó el programa original. Así pues, se considerará que el titular de los derechos intelectuales de la obra original, que fue utilizada sin su autorización para la creación de la obra nueva no ostenta ninguna porción de la titularidad de los derechos intelectuales de ésta, si se configura un supuesto de "*fair use*".[87]

[86] WIPO, Glossary of Terms of The Law of Copyright and Neighboring Rights. [en línea] ftp://ftp.wipo.int/pub/library/ebooks/wipopublications/wipo_pub_816_efs-ocr-sp-image.pdf

[87] ROJO F., Philosophical Foundations of the Fair Use Doctrine, Escuela de Derecho, Universidad Torcuato di Tella, Argentina, [en línea] file:///C:/Users/Neida%20Zambrano/Desktop/Carpeta%20provisional%20RED/Fair%20Use/Fundamentos%20filos%C3%B3ficos%20de%20la%20doctrina%20del%20fair%20use.html

Entonces es el "*fair use*" la limitación más emblemática e importante a los derechos exclusivos del *copyright* norteamericano, lo cual entre otras cosas tiene su fundamento en la innovación y productividad de la sociedad.

Esta doctrina de uso justo, puede considerarse como una de las defensas más importantes dentro del *Copyright Act* contra las reclamaciones de infracción de derechos de autor, donde muchos programadores y empresas del rubro de la tecnología dependen de ella para crear obras nuevas como innovaciones tecnológicas.

Causales eximentes para el uso sin retribución económica de los derechos patrimoniales

Según GARCIA, la jurisprudencia norteamericana ha creado tres modelos para explicar el "*fair use"*, el primero es el "*transformation insights"* que implica que la obra posterior le haya incluido un nuevo mensaje, una nueva expresión a la anterior; si eso es así la obra es justa y, por lo tanto, no viola la *Copyright Act*. El segundo modelo, el de la metamorfosis creativa, no requiere un nuevo mensaje, una nueva expresión, sino una suficiente modificación artística. El tercer modelo atiende a si el trabajo posterior

implica una nueva propuesta *(new proposal)* diferente a la del trabajo original sobre la cual se basó la obra.[88]

Observamos entonces, como Estados Unidos ha considerado la importancia de la aplicabilidad del uso de las obras catalogadas como derecho de autor cuando se promueva la originalidad y la creatividad. Es así que, el "*fair use"*, que en un principio fue elaborado por la doctrina judicial estadounidense como un límite al derecho de reproducción, fue recogido de modo expreso en el "*Copyright Act*" como una limitación a todos los derechos exclusivos contemplados en el mismo texto legal, como se evidencia del texto de la Sección 106.

Esta norma establece que "No obstante las previsiones de las secciones 106 y 106 A, el uso leal de una obra protegida por el derecho de autor, incluyendo el uso por reproducción en copias o fonogramas o mediante cualquier otro medio especificado en tal sección, para propósitos tales como la crítica,

[88] GARCIA E., El papel del juez frente a las limitaciones y excepciones al derecho de autor en los países de tradición continental, [en línea] http://propintel.uexternado.edu.co/el-papel-del-juez-frente-a-las-limitaciones-y-excepciones-al-derecho-de-autor-en-los-paises-de-tradicion-continental/

comentarios, reporte de noticias, enseñanza (incluyendo múltiples copias para el uso de la clase), becas o investigación, no es una infracción al derecho de autor"[89].

Estos supuestos se encuentran establecidos en la Sección 107 del *Copyright Act*, los cuales se conforman de cuatro factores que deben ser analizados y aplicados, a los fines de determinar si el uso de una obra en particular puede ser catalogada como de uso justo o caso contrario, pueda considerarse que ha infringido la propiedad intelectual de un tercero. Los factores a considerar son:

"(1) El propósito y carácter del uso, incluyendo si éste es de uso comercial o si tiene un propósito educacional sin fines de lucro;
(2) La naturaleza de la obra objeto de protección por derecho de autor;
(3) La cantidad y sustancia de la porción de la obra usada y protegida mediante derecho de autor como un todo; y

[89] Traducción libre de la autora. Copyright Act de Estados Unidos, OMPI: http://www.wipo.int/wipolex/es/details.jsp?id=5405

(4) El efecto de tal uso sobre el mercado potencial o el valor de la obra protegida mediante derecho de autor.

El hecho de que la obra no sea publicada, no impedirá en sí mismo considerar su uso justo, si tal consideración está hecha sobre los factores anteriores"[90].

Los mencionados supuestos establecen los parámetros del uso justo, los cuales vienen a proporcionar un mecanismo para considerar el impacto de excusar la infracción sobre los derechos exclusivos de los autores. Es el juez quien tendrá la libertad de determinar si hay o no uso justo, por lo que el resultado en cualquier caso es impredecible.

Es así que, para determinar si procede o no el *fair use* se debe observar que éste se aplica después de la conclusión de que el material del demandante está protegido ("*copyrightable*") y que el demandado ha copiado la expresión ("infracción").

[90] Ley citada ut supra.

Asimismo, debe considerarse que si bien la *Copyright Act*. establece estos cuatro factores, no los define, dejando que los tribunales sean quien los apliquen y decidan en base a la regla equitativa de la razón. Esto significa que los resultados pueden variar caso a caso, pues estos factores son las pautas que deben considerar los tribunales, y adaptarla a situaciones particulares.

El Propósito o carácter del uso.

Cuando se habla del primer supuesto, de acuerdo con BEEBE, este factor es uno de los dos más importantes (junto con el cuarto factor) para los jueces norteamericanos, a la hora de determinar si un uso no autorizado de una obra ajena califica o no como "*fair use"*.

Según BEEBE, en 95.3% de los 148 fallos en los que los jueces, al interpretar este factor, sostuvieron que "el propósito y el carácter del uso" no autorizado de la obra no eran favorables a un "*fair use"*, se terminó fallando que no lo hubo. A su vez, en 90.2% de los fallos en los que los jueces sostuvieron que

"el propósito y el carácter del uso" no autorizado eran favorables a un *fair use,* se terminó fallando en ese sentido.[91]

Al analizar el primer factor, se debe revisar la naturaleza y el propósito de la obra, para lo cual se analizará si el uso fue comercial, si éste fue transformador respecto a la obra original y si éste último es considerado como un subfactor del primer factor.

Se puede decir que, cuando hablamos de uso no comercial se debe considerar si se utilizó como "(1) uso personal; (2) si el uso de los demandantes por parte del demandado estaba relacionado con algún tipo de educación formal; (3) si el uso del demandado fue para fines de investigación general y no fue parte de un proceso específico de desarrollo de productos; y (4) las actividades relacionadas al uso, como el intercambio de archivos entre pares que

[91]BARTON B., An Empirical Study of U.S. Copyright Fair Use Opinions, 1978-2005, 156 Pennsylvania, Law Review p. 549, [en línea] http://www.bartonbeebe.com/documents/Beebe%20-%20Empirical%20Study%20of%20FU%20Opinions.pdf

comuniquen al público el trabajo del demandante; todo esto sin beneficio comercial directo o indirecto[92].

Es importante, dejar asentado que el uso comercial a la hora de determinar el uso justo va a pesar menos que el uso transformador de la obra. Así lo ha dejado establecido la jurisprudencia, al señalar que "cuanto más transformador sea el trabajo de un acusado, más factores, como el comercialismo, retrocederán en importancia. Por el contrario, cuanto menos transformador sea el trabajo acusado, más factores como el comercialismo dominarán"[93].

Respecto al otro supuesto, el uso transformador, el Noveno Circuito se refirió al significado de transformador si la obra nueva añade algo nuevo, con otro propósito o carácter diferente, alterando el primer uso con nueva expresión, significado o mensaje en lugar de meramente reemplazar los objetos de la creación original. Puede pasar que un demandado cambie el trabajo protegido

[92] SIG, M., Predicting Fair Use, kb.osu.edu Ohio State Law Journal: Volume 73, Issue 1 (2012) Ohio State Law Journal, vol. 73, no. 1 (2012), p.67, [en línea] http://hdl.handle.net/1811/71532
[93] Ob citada *ut supra*, p. 14

por el autor o, cuando éste los use en un contexto diferente de tal manera que el trabajo original se transforma en una nueva creación. Una obra no es transformadora cuando el usuario hace poca o ninguna alteración al contenido expresivo o mensaje de la obra original y la usa en el mismo o similar contexto[94].

Cuando hablamos entonces de un "uso transformativo", básicamente nos referimos a cuando a la obra original se le agrega algo nuevo, con un propósito distinto o con distintas características, y debe necesariamente alterarse la versión inicial con nueva expresión, sentido o mensaje y cuando la nueva obra tiene una "función totalmente diferente" a la función de la obra original. La transformación de la obra original para la creación de la obra nueva debe ser posible de ser "razonablemente percibida" por el público[95].

[94] In the United States District Court for the Northern District of California Notice of Final Charge to the Jury (phase one) and Special Verdict Form, en el caso: Oracle América INC v. Google INC., p. 13, [en línea]
https://es.scribd.com/doc/313560685/Jury-Instruction-Oracle-v-Google
[95] FACUNDO R., Philosophical Foundations of the Fair Use Doctrine, Escuela de Derecho, Universidad Torcuato di Tella. [en línea] http://www.scielo.org.mx/pdf/is/n41/n41a4.pdf

Los tribunales han descrito las nuevas obras como una importante transformación, cuando estas utilizan material protegidos para fines distintos del propósito del material original, es decir, cuando el tercero cambia la obra protegida o utiliza la obra protegida en un contexto diferente, de modo que la obra originaria se transforma en una nueva creación.

La decisión emblemática que adoptó la transformación como el corazón del "*fair use"* fue la decisión Campbell en el año 1994. Aquí, la cuestión clave en Campbell era la extensión de la copia permisible en el contexto de la parodia musical. Según la Corte, la parodia es una forma de uso transformador que provee beneficio social, "al esclarecer una obra anterior y, en el proceso, crear una nueva"[96].

El autor ASAY al definir qué es un uso transformador, lo hace citando el referido caso *Campbell v Acuff - Rose Music*, indicando que por transformador se puede considerar cuando se añade algo nuevo, con otro

[96] SIG, M. Predicting Fair Use, kb.osu.edu Ohio State Law Journal: Volume 73, Issue 1 (2012), Ohio State Law Journal, vol. 73, no. 1 (2012), p. 55, [en línea] http://hdl.handle.net/1811/71532

propósito o carácter diferente, “alterando la obra original con nueva expresión, significado o mensaje”. Es decir, la parte que reclama el uso justo debe haber utilizado el trabajo de una manera diferente a la parte original[97]. Cuando un tribunal considera que el uso de una obra es transformador, a menudo descarta los otros factores o encuentra que la naturaleza transformadora del uso puede influir fuertemente para determinar si es justo o no.

Es así que, un producto no se considera transformador si el usuario "no altera el contenido expresivo o el mensaje de la obra original”[98]. Los tribunales modernos se centran frecuentemente en si el uso es transformador, por lo que, si procede este supuesto, es muy probable que prevalecerá la defensa del uso justo, caso contrario si el uso no se considera transformador, el demandado se enfrentará a una batalla cuesta arriba para justificar una defensa mediante el *fair use*.[99]

[97] CLARK A., Transformative Use of Software, [en línea] https://www.stanfordlawreview.org/online/transformative-use-in-software/
[98] En GEO. L. TECH. REV. 62 (2016) [en línea] https://perma.cc/X6WV-Z28A, p. 67, cita a: (quoting Seltzer v. Green Day, Inc., 725 F.3d 1170, 1177 (9th Cir. 201)
[99] Ob citada “*Transformative Use in Software*”,

Tomando en consideración que el hecho transformador puede variar de un caso a otro, todo dependerá de cuanto mayor sea la transformación, pues más probable será que el uso del infractor califique como un uso justo, y cuanto menos la transformación, menos probable el uso calificará como un uso justo.[100]

En otras palabras, para ser transformadora, una obra nueva no debe reemplazar la obra original en el mercado, es decir, agregar contenido adicional a una obra y usarla para el mismo propósito que la obra protegida por derechos de autor no es transformadora, por lo que, será importante que la obra nueva contenga algo nuevo, que sea con otro propósito o carácter diferente, y debe de haber alteración con nueva expresión, significado o mensaje, en relación a la primera obra.

[100] In The United States District Court for The Northern District of California Notice of Final Charge to the Jury (Phase One)And Special Verdict Form, en el caso Oracle America INC v. Google INC., p. 13 [en línea] https://es.scribd.com/doc/313560685/Jury-Instruction-Oracle-v-Google

Entonces, se habla de un trabajo transformador como aquella obra que impregna al original propósito o carácter diferente, alterando el primero con nueva expresión, significado o mensaje.

La Naturaleza de la obra protegida

El segundo supuesto se centra en la naturaleza de la obra protegida por derechos de autor, en el cual debe evaluarse si su uso no autorizado califica o no como un *fair use.*

Aquí el alcance del uso justo, tiende a ser algo más amplio si el trabajo es funcional que si es artístico o fantasioso. Si el trabajo es publicado o no publicado[101]. Por ejemplo, si la obra original que se ha utilizado para realizar la obra nueva era una obra aún inédita, es más probable que los jueces consideren que no medió *fair use* que si se trataba de una obra ya publicada,

[101] SAMUELSON P., Fair Use for Computer Programs and Other Copyrightable Works in Digital Form: The Implications of Sony, Galoob And Sega, p. 57-58. [en línea] https://www.law.berkeley.edu/php-programs/faculty/facultyPubsPDF.php?facID=346&pubID=147

ya que en dicho supuesto se privaría al autor original de su derecho a ser el primero en publicar su obra[102].

De acuerdo con BEEBE, a pesar de que este factor está explícitamente reconocido en la ley norteamericana, no es muy tomado en cuenta por los jueces norteamericanos: en 17.7% de los 306 fallos jurisprudenciales analizados por el autor, los jueces omitieron analizar este factor, y en 6.5% los jueces, al analizar este factor, sostuvieron que era "irrelevante"[103].

En el caso Campbell v. Acuff-Rose, se dice que dicho factor requiere reconocer que ciertas obras necesitan una mayor protección que otras. Así, para la jurisprudencia norteamericana, mientras más creatividad hay en una obra, merece mayor protección; en consecuencia, se estima que las obras basadas en hechos merecen una menor protección. Sin embargo, dice la

[102] ROJO F., Philosophical Foundations of the Fair Use Doctrine, Escuela de Derecho, Universidad Torcuato di Tella, [en línea] http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-02182014000200004

[103] BEEBE B., University of Pennsylvania Law Review. [en línea] https://www.law.upenn.edu/journals/lawreview/articles/volume156/issue3/Beebe156U.Pa.L.Rev.549(2008).pdf

sentencia, cuando se trata de usos transformativos, el segundo factor no resulta de mucha utilidad para el análisis[104].

Como señaló el Tribunal en el caso Oracle v. Google, el foco de este factor es la importancia del material utilizado para los valores fundamentales de la protección de derechos de autor. Cuanto menos material utilizado implique los valores fundamentales de protección de derechos de autor, más viable será el uso justo y viceversa[105].

Es así que, por ejemplo, en caso de programas informáticos se debe considerar la medida en que los materiales usados fueron creativos versus funcionales. En tal sentido, cuanto más creativo sea el trabajo, tanto más este factor desfavorece el uso justo, y cuanto más funcional sea el trabajo, más se favorece el uso justo bajo este factor. Considerando como original, que la

[104] CETINA R. ¿De la sartén al fuego? La cuestión de la adopción del fair use como solución al sistema de límites al derecho de autor en México, [en línea] file:///C:/Users/Neida%20Zambrano/Downloads/103-401-1-PB.html

[105] The United States District Court for The Northern District of California, Notice of Final Charge to The Jury (Phase One) And Special Verdict Form, en el caso Oracle America INC v. Google INC., p. 15 [en línea] https://es.scribd.com/doc/313560685/Jury-Instruction-Oracle-v-Google

obra fue creada independientemente por el autor (en contraposición a copiada de otras obras) y que posee al menos cierto grado mínimo de creatividad[106].

El tercer factor guarda relación con la obra protegida. Aquí se va tener que examinar la cantidad y la sustancialidad de la obra, es decir, se considera que cuanto menos sustancial sea la porción de la obra original usada para la creación de la obra nueva, más probable será que la obra nueva califique como un *fair use*[107].

Asimismo, este factor se deberá analizar con base a la obra protegida por derechos de autor, no el trabajo infractor. En efecto, "ningún plagiario puede excusar el error mostrando cuánto de su trabajo no es pirata. En contraste, el hecho de que una porción sustancial del trabajo infractor fue copiada literalmente es una prueba del valor cualitativo de la copia material, tanto para el titular de la obra como para el plagiario que busca beneficiarse de la comercialización"*[108]*

[106] Ob citada p. 15-16
[107] Ob. citada: Facundo Rojo, Philosophical.
[108] [en línea] p. 20: https://cyber.harvard.edu/people/tfisher/cx/2014_Oracle.pdf

Otra cosa que se debe tomar en cuenta es: 1.si la persona que usó la obra protegida sólo copia tanto como es necesario para su uso previsto, entonces este factor no pesará contra éste; 2. la justificación de la copia particular realizada; y 3. si es por motivo de investigación, se debe analizar en conjunto con el primero de los factores estatutarios, pues la extensión de la copia permisible varía con el propósito y carácter del uso[109].

De acuerdo con BEEBE, este factor está fuertemente vinculado al primer factor y al cuarto factor: mientras menor sea la porción que se toma de la obra original, resultará más probable que el uso sea transformativo (primer factor) y que la aparición de la nueva obra no afecte la posición en el mercado del titular de la obra original (cuarto factor).[110]

El efecto del uso de la obra en el mercado potencial de la obra original.

En cuanto al cuarto factor, este es, el efecto del uso en el potencial mercado o en el valor de la obra original, es otro factor importante junto con el

[109] [en línea] p. 20: https://cyber.harvard.edu/people/tfisher/cx/2014_Oracle.pdf
[110] Ob. citada BEEBE B, University of Pennsylvania…

primero, aquí debe considerarse si el uso de la obra priva al propietario de la comercialización o de los ingresos, para lo cual se debe considerar si la obra nueva creada que utilizó otra obra (original) compiten directamente en el mercado.

En este factor, la Corte Suprema de Estados Unidos ha dejado asentado que es uno de los elementos más importantes del uso justo. Exige que los tribunales consideren no sólo el alcance del daño al mercado causada por las acciones concretas del presunto infractor, sino también por la conducta irrestricta y generalizada del tipo en que haya intervenido el acusado.[111]

Es así que éste va a involucrar dos aspectos, por un lado, si el uso en cuestión actúa directamente como un substituto en el mismo mercado de la obra original; y, por otro lado, si existe un daño potencial en el mercado que trasciende al directo, en la existencia potencial de un mercado de licencias.[112]

---

[111] **ORACLE AMERICA, INC. v. GOOGLE INC.,** United States Court of Appeals, Federal Circuit. Decided: May 9, 2014,

[en línea] p. 20: https://cyber.harvard.edu/people/tfisher/cx/2014_Oracle.pdf

[112] IBAÑEZ C., Fair Use como Modelo de Flexibilización del Derecho de Autor, Memoria para optar al grado de Licenciado en Ciencias Jurídicas y Sociales, p. 44, [en línea] http://repositorio.uchile.cl/bitstream/handle/2250/113335/de-ibanez_c.pdf?sequence=1

Asimismo, señala que el daño al mercado es una cuestión de grado, y la importancia de este factor variará, no sólo con la cantidad de daño, sino también con la fuerza relativa de la demostración en los otros factores. Aquí debe considerarse si se ha afectado la comercialización de la obra y si afectó las ventas debido a la competencia de la nueva obra.

Entonces, el análisis comienza con considerar cualquier daño real o potencial a un mercado, el producto copiado debe estar en el mismo mercado que el producto original. Por lo tanto, aquí se analizará si los tribunales deben revisar si el uso de un programa informático existente, puede dañar sustancialmente o no el mercado donde éste se desenvuelve.

Ahora bien, de acuerdo con los supuestos antes mencionados, podemos de manera general señalar que con el "*fair use"*, en algunos casos podemos utilizar para la creación de nuevas obras, la propiedad intelectual de un tercero sin autorización de su titular. Incluso, si la obra nueva contiene una parte de una obra original preexistente, el titular de los derechos intelectuales de la obra original no ostentará ningún derecho de propiedad intelectual sobre

la obra nueva, la cual será exclusivamente de quien tomó la obra original y la utilizó de manera transformadora para la creación de la obra nueva.

Así pues, la doctrina de "*fair use"* tiene una incidencia muy importante dentro del contenido del *copyright* recaído en los programas informáticos, por lo que en principio se podría considera que copiar un programa con la finalidad de alcanzar interoperabilidad de aplicaciones constituye un uso justo y, por tanto, no infringe el *copyright* del programa.

Se reafirma que esta doctrina tiene su basamento en la jurisprudencia norteamericana, la cual ha delineado los supuestos de esta figura, por lo cual se considera importante adentrarnos en las opiniones sobre uso justo de algunas decisiones de tribunales estadounidenses, las cuales han variado año tras año, reflejándose de las disputas relacionadas con programas informáticos.

Decisiones emblemáticas norteamericana sobre las interfaces de los programas informáticos.

Dentro de uno de los precedentes importantes para muchas situaciones en las que las defensas de uso justo podrían ser afirmadas en casos de *copyright* de *software*, podemos encontrar al caso *Sony vs. Universal City Studios, Inc* (1984), en el que la Corte Suprema sostuvo que las grabaciones en formato Betamax de Sony de programas de televisión previamente difundidos constituían un uso justo, poniendo el énfasis más en el primer factor, seguido respectivamente por el cuarto (efecto de mercado) y tercero cantidad.

El caso se basa en que Sony produjo y vendió a Betamax una grabadora de vídeo casero, que permitía a los clientes grabar programas de televisión. Universal Studios demandó a Sony por infracción de derechos de autor basada en los clientes de Sony que usaban Betamax para grabar las transmisiones televisadas con derechos de autor de Universal Studios.

Sony alegó un uso justo y el Tribunal de Distrito se declaró a su favor, pero el Tribunal de apelación revocó el fallo.

La Corte Suprema de Estados Unidos concedió *certiorari* y revocó la decisión de la Corte de apelación, que se pronunció contra Sony. Esta Corte resolvió que las copias realizadas de forma privada y no comercial estaban amparadas por la doctrina del "*fair use"*, pero que las copias que se realizaban con fines comerciales debían presumirse violatorias de los derechos de propiedad intelectual de los creadores originales. La Corte sostuvo que, en caso de que las copias se realizaran con fines comerciales, para dejar sin efecto la presunción de que eran violatorias de los derechos de propiedad intelectual de los creadores originales, quien realizaba las copias debía demostrar que la comercialización de tales copias no perjudicaba o perjudicaría al creador de la obra original en el mercado[113].

En particular, la Corte se refirió al hecho de que el uso primario de las cintas Betamax era el uso doméstico privado, que debe caracterizarse como una actividad no comercial y sin fines de lucro. Debido a que el propósito era la visualización privada y personal en el hogar en un momento posterior, la Corte consideró que había una presunción de lealtad en el uso.

[113] Ob citada: ROJO F.

En el caso Altai Inc, se desprende que éste y Computer Associates eran competidores en el mercado de *software* de programación diseñado para funcionar en dos tipos de máquinas IBM.

Altai desarrolló una nueva versión del programa Zeke, construyendo un nuevo componente de compatibilidad. Para esto, contrató a un programador de nombre Arney, el cual le propuso que la mejor manera de rediseñar al programa Zeke para que fuera compatible con dos sistemas operativos diferentes de IBM, era construir un nuevo componente de compatibilidad para Zeke. Es decir, Arney planeó construir un subprograma (finalmente llamado Oscar), que transpondría los comandos de Zeke para tareas específicas en el formato apropiado para que cada sistema operativo pudiera, a su vez, instruir correctamente al hardware de IBM para llevar a cabo los comandos del programa de programación de Zeke[114].

Este nuevo diseño evitaría la necesidad de personalizar cada módulo de Zeke para los dos sistemas operativos. Sin saberlo, Altai, Computer Associates había adoptado el mismo enfoque en la última versión de un programa de su

[114] SAMUELSON, P., *The Uneasy Case for Software Copyrights Revisited*, p. 1767-1768

propiedad, un proyecto en el que Arney había trabajado cuando había estado en el empleo de Computer Associates[115].

Altai ganó la controversia ante el Tribunal de Apelaciones del Segundo Circuito en Nueva York, estableciéndose un importante precedente. La decisión de Altai no sólo rechazó las reclamaciones de protección de derechos de autor en las interfaces, sino que también adoptó una prueba de tres pasos ampliamente utilizada para evaluar las reclamaciones de infracción de derechos de autor en los programas informáticos.

El primer paso consiste en construir una jerarquía de abstracciones, desde las más abstractas hasta las más detalladas, para el programa del demandante. El segundo paso consiste en "filtrar" los distintos elementos del programa que están fuera del alcance de la protección de derechos de autor. El tercer paso consiste en comparar las "pepitas doradas" restantes de expresión con el programa del demandado para determinar si el acusado copió cantidades sustanciales de expresión del programa del demandante[116].

[115] Ob citada *ut supra.*
SAMUELSON P., The Strange Odyssey of Software Interfaces and Intellectual Property Law UC Berkeley, p. 8, en: http://repositories.cdlib.org/bclt/lts/59

El mencionado Tribunal afirmó que las similitudes entre los programas de Computer Associates y Altai eran "*dictadas por las demandas funcionales*" de los programas en cuestión o que de otro modo estaban en el dominio público, Altai necesitaba tener listas de servicios y listas de parámetros debido a que tanto Computer Associates, como los programas de programación de Altai, fueron diseñados para proporcionar los mismos servicios y cumplir con los procedimientos de interfaz necesarios para interoperar con los programas de IBM OS[117]

En este caso, el Tribunal estableció que los elementos de los programas "dictados por factores externos", tales como "requisitos de compatibilidad de otros programas con los que un programa está diseñado para funcionar conjuntamente", están fuera del alcance de la protección que el derecho de autor otorga a los programas.

Como el derecho de autor parecía inadecuado para proteger las innovaciones de los programas, el tribunal de Altai sugirió 1. que debía considerarse si los

[117] Ver: Sega Enters. v. Accolade, Inc., Corte de Apelaciones Noveno Circuito, 1992, [en línea] https://www.law.berkeley.edu/files/sega.pdf

programas necesitaban protección adicional de la propiedad intelectual; y 2. que las patentes podría ser una forma más adecuada de protección de la propiedad intelectual para el programa SSO.

Por otro lado, otro caso que tocó el tema del uso justo de las interfaces fue el caso Sega, en el que influenciado por la decisión del *copyright* del *software* de Altai, Inc., se determinó que los requisitos funcionales para alcanzar interoperabilidad no eran protegibles por la ley de *copyright*.

El caso se planteó por cuanto Accolade INC estaba tratando de fabricar videojuegos para varias consolas de juego, pero querían compatibilidad con el sistema Sega [118]. Para determinar los requisitos de compatibilidad, Accolade hizo ingeniera inversa del *software* de videojuegos Sega para encontrar el código necesario a los fines de lograr la compatibilidad, no obstante, éste escribió sus propios procedimientos y desarrolló sus propios juegos.

[118] Ver: Sega Enters. v. Accolade, Inc., Corte de Apelaciones Noveno Circuito, 1992, [en línea] https://www.law.berkeley.edu/files/sega.pdf

Aquí, el Tribunal determinó que las ideas centrales del programa del código copiado, fueron sobre todo elementos funcionales, ponderando a favor un uso justo de éste.

Así pues, se puede señalar que Sega fue importante en la odisea IP-in-interfaces por al menos cuatro razones: 1. consideró el enfoque retórico de Altai para conceptualizar los programas de computadora como obras utilitarias elegibles para un ámbito limitado de protección de derechos de autor; 2. Sega siguió a Altai en cuanto a determinar que las interfaces eran elementos de programas que la ley de derechos de autor no protegía; en realidad, Sega habló de la información de interfaz como "requisitos funcionales para lograr la compatibilidad con otros programas; 3. El tribunal dictaminó que la copia de código de programa en el curso de la ingeniería inversa para un propósito legítimo, como la extracción de información de interfaz para hacer un programa compatible No infringe ningún derecho de autor en ese código; y 4. indicó que incluso copiar un código exacto de otro programa no sería infracción en la medida en que ese código era esencial para lograr la interoperabilidad"[119].

[119] Ob. citada *ut supra*, p. 12

En este caso, el Tribunal determinó que, si la copia/reproducción/ingeniería inversa de un código objeto protegido por derechos de autor es *per se* un uso injusto, el titular de los derechos de autor gana un monopolio de facto sobre los aspectos funcionales de su trabajo, lo cual el Congreso negó expresamente. Para poder disfrutar de un monopolio legal sobre la idea o el principio funcional que subyace a una obra, el creador de la obra debe satisface las normas más estrictas impuestas por las leyes de patentes.

De los cuatro factores estatutarios, la Corte concluyó que el uso del Accolade fue justo, afirmando que el primer, el segundo y el cuarto de los factores legales pesaban a favor de Accolade, y que sólo el tercer factor pesaba ligeramente a favor de Sega.

Respecto al primer factor del *fair use*, el tribunal señaló que la presunción contra el uso justo cuando se trata de un propósito comercial, puede ser refutada examinando la característica del uso particular. En cuanto al segundo, el tribunal declaró que debido a que Accolade copió el código objeto sólo para revelar las ideas y elementos funcionales del programa sin

copiarlos, la naturaleza comercial de la copia era sólo "*indirecta o derivada*"[120].

En cuanto al cuarto factor, señaló que todo intento de monopolizar el mercado al hacer imposible que otros compitan, es contrario al propósito estatutario de promover la expresión creativa y de innovación, con lo cual no se podrá constituir una base sólida y fuerte para resistir la invocación de la doctrina del uso justo. Por lo tanto, concluyó ese Tribunal que este factor legal pesa en el favor de Accolade, no de Sega, a pesar de la pérdida económica (menor) que Sega podía sufrir"[121].

En el referido caso se señaló que, los únicos medios confiables para proteger los requisitos funcionales para lograr la interoperabilidad era patentarlos. Las patentes tenían por lo menos una ventaja sobre la ley de derechos de autor en la protección de interfaces, porque el derecho de patentes no tiene una doctrina de "fusión".

[120] PETERS R., Ninth Circuit Holds That Intermediate Copying of Object Code Is Fair Use, Santa Clara High Tech. L.J. 579 (1993)., p. 584, [en línea] http://digitalcommons.law.scu.edu/chtlj/vol9/iss2/8
[121] Sentencia citada en la obra ut supra, p. 586

Por lo tanto, si sólo hay una forma de lograr una función particular y un desarrollador ha patentado de una manera, puede ejercer sus derechos de patente para detener los usos sin licencia.

Otro caso en esta materia, es el caso Lotus Development Corp. v. Borland International, Inc[122], en el cual se desarrolló en virtud de que Borland fue demandado por Lotus por la comercialización de una hoja de cálculo (Lotus 1-2-3), por cuanto el Borland había duplicado toda su interfaz de usuario, violando así su copyright menú 1-2-3, es decir éste, replicó 469 comandos del programa Lotus (como, por ejemplo: copiar/imprimir, etc.) y su organización, sin copiar el código fuente del programa.

Lotus es el creador de "Lotus 1-2-3", un programa de hoja de cálculo que permite a los usuarios realizar funciones de contabilidad electrónicamente en una computadora". Una parte integral del programa Lotus 1-2-3 es la

[122] Lotus Development Corp. v. Borland Intern., Inc., 49 F.3d 807 (1995), párrafo 819, [en línea]

https://scholar.google.com/scholar_case?case=9888762079230732186&q=Lotus+v.+Borland,+49+F.3d+807+(1995)&hl=en&as_sdt=40000006

estructura de comandos de menú, que tiene 469 comandos dispuestos en más de 50 menús y submenús[123].

Borland, parte demandada en este caso, había creado un programa denominado "Quattro", el cual es una hoja de cálculo que compite con el programa de Lotus 1-2-3. El primero tiene su propia estructura de comando de menú, no obstante, Borland incluyó una copia de la estructura de comandos de Lotus 1-2-3 en el programa Quattro para que los usuarios ya familiarizados con la estructura de comandos de Lotus pudieran usar Quattro sin tener que aprender una nueva estructura de comandos.

Alega Borland que, al copiar la estructura de comandos de Lotus, Borland no copió ningún código fuente de Lotus u otros elementos no literales del programa Lotus, como las pantallas[124].

El tribunal de Distrito determinó en primera instancia que, debido a que había varias maneras diferentes de producir una interfaz, la interfaz de Lotus 1-2-3

[123] Ob. citada ut supra
[124] Ob citada ut supra

era una expresión y, por tanto, protegida por derechos de autor, prohibiendo a Borland la distribución de los programas de computadora Quattro o Quattro Pro que contenían una copia de los comandos de Lotus 1-2-3menu y / o la estructura del menú, en cualquier forma[125].

En apelación, la Corte del Primer Circuito revocó la decisión del tribunal de distrito, encontrando que la estructura de comando del menú era un "método de operación", por lo que era susceptible de protección bajo el17 USC § 102 (b).

Señala el Juez que, la estructura de comando de Lotus fue escrita para que la gente pudiera aprender su contenido y la usaran. Encontró que la estructura de comandos de Lotus era análoga a los botones utilizados para controlar un grabador de videocasete ("VCR"). Ejemplificó señalando que cuando un usuario de Lotus elige un comando, como "Imprimir", resaltándolo o

[125] GOT J., Lotus Development Corporation V. Borland International: The United States Court of Appeals for the Firs, Circuit Takes a Step Backward for the Copyright Protection of Computer Programs, p.1353. [en línea] http://dspace.creighton.edu:8080/xmlui/bitstream/handle/10504/40205/49_30CreightonLRev1349(1996-1997).pdf?sequence=1

escribiendo la letra "P", el usuario empuja efectivamente un botón de la misma manera que presiona "Reproducir" en la videograbadora.[126]

También se señaló que, si bien el Tribunal del Primer Circuito aplicó la prueba de extracciones / filtraciones / comparación para determinar la capacidad de protección de elementos no literales de un programa informático, establecido en Computer Assoc. Int. v. Altai, Inc., no resultaría de ayuda para evaluar si la copia literal de una jerarquía de comandos de menú constituye una infracción del derecho de autor, pues extraer las jerarquías de comandos de menú a sus niveles individuales de palabra y menú y luego filtrar la idea de la expresión en esa etapa, como requieren dicha prueba, ocultan la cuestión fundamental de si una jerarquía de comandos de menú puede estar protegida por derechos de autor.

La consulta inicial no debería ser si los componentes individuales de una jerarquía de comandos de menú son expresivos, sino más bien si la jerarquía de comandos de menú como un todo puede estar protegida por derechos de autor.

[126] Ob citada.

La Corte de los Estados Unidos sostuvo la misma postura antes comentada, ésta es que la estructura de comando de menú 1-2-3 no infringía la propiedad intelectual de Lotus porque era un método de operación bajo 17 U.S.C. § 102 (b). La Corte dictaminó además que el uso por parte de Borland de las Lotus 1-2-3 fue justo. También declaró que este resultado contradecía fundamentalmente la intención del Congreso que subyace al artículo 102 (b) de la Ley de Derecho de Autor.

Las anteriores decisiones proporcionan análisis sobre las implicaciones del derecho de autor sobre la reimplementación de un segundo ingrediente de procedimientos de interfaz que son necesarios para interoperabilidad.

CONCLUSIONES

A la luz de lo expuesto en los capítulos anteriores, es importante señalar que las decisiones recaídas en el caso Oracle v. Google, nos deja claro que en el campo tecnológico aún no existe certeza en lo relacionado con el alcance de la protección jurídica de un programa informático, dada la particularidad que reviste el *software* en la práctica destacando su carácter funcional, se puede decir que no existe hasta el momento un criterio uniforme respecto a qué es lo que se protege y lo que no.

En virtud de la aludida naturaleza funcional que caracteriza al *software*, es muy importante que exista un equilibrio entre el derecho del autor (respecto a la protección de las expresiones originales) y la garantía de que esa protección no impida las innovaciones a nivel tecnológico por parte de terceros que utilizan como base ideas o funciones de obras preexistentes.

Pese a la ausencia de uniformidad de criterio antes referida, la cual ha quedado en evidencia en la sentencia de la Corte Federal en el caso Oracle v. Google, prevalece la posición que sostiene que las ideas y las funciones de un programa informático están excluidos del alcance del derecho de autor.

Se pudo evidenciar además, que aun en caso de que se inclinara la balanza a favor de señalar que las interfaces sí están protegidas por el derecho de autor, bajo ese supuesto pasaría a ser protagonista la figura del "*fair use"* o uso justo, tal y como lo determinó la Corte Federal en el caso Oracle v. Google.

De esta manera, para determinar si se aplica o no el *fair use* lo primero que debe precisarse es si la obra se encuentra protegida o no por el *copyright*, lo cual será un asunto un poco complejo cuando se trate de funciones de programas informáticos específicas como las interfaces, en virtud de que las mismas están destinadas mediante su funcionalidad a ser interoperables con un determinado lenguaje de programación. No obstante, como observamos, ha quedado evidenciado que el *fair use* sí tiene aplicabilidad como limitación al *copyright* de los programas informáticos.

En el derecho norteamericano existe un sistema abierto para determinar si el uso es justo o no. Aun cuando existen cuatro factores referenciales para determinar un uso de tales características, los mismos no restringen el empleo de otros elementos, debiéndose realizar un análisis caso por caso, contrario

al sistema continental que es cerrado, en el cual las limitantes al derecho de autor son reguladas y aplicadas en cada legislación de manera taxativa.

Así pues, dentro del contexto de un sistema abierto de *fair use* como el estadounidense, observamos que la legislación no ofrece un catálogo de limitantes sino criterios para establecer el uso justo de una obra preexistente, tal como se ha indicado. Notamos que estos se aplican mediante la regla de la razón y un criterio de equidad o beneficio social que deberá aplicar el juez según el caso en particular, por lo que ha sido la jurisprudencia la encargada de desarrollar y delinear lo que puede considerarse como un uso justo o no respecto a los elementos de un *software*.

Es así que, pudimos notar que la teoría del *fair use* es una herramienta flexible a la limitación esencial del *copyright*, la cual otorga la posibilidad de que los programadores puedan usar y/o reimplementar programas informáticos y que dicho uso no infrinja la propiedad del titular de la obra originaria, siempre y cuando -esto es importante resaltarlo- el uso haya sido de manera justa, según el análisis que se haga de los criterios previstos en el 107 de la Copyright Act,

los cuales, como lo ha reconocido la jurisprudencia, no son exclusivos, pues otras consideraciones pueden ser tomadas en cuenta.

Entonces, al titular de la obra que considere que su programa informático fue infringido le corresponderá demandar ante el tribunal correspondiente, quien deberá analizar las características del caso a fin de determinar si el uso fue justo. En efecto, es allí donde el Juez debe valorar dichos criterios a los fines de determinar si el uso de elementos de un programa informático puede constituir infracción al *copyright* o, al contrario, puede considerarse que fue un uso justo, caso en el que resultará improcedente la demanda.

Pudimos observar que, aunque la comercialización pesa en gran medida en contra del uso justo, en el caso aquí estudiado se pudo evidenciar que Google mediante el uso de las APIs de Java, si bien pudo generar ganancias, por ejemplo, a través de publicidad, no es menos cierto que, es evidente que Android es de uso libre (disponible para todos los usuarios que quieran usarlo), es decir, Google no cobra dinero por su utilización, motivo importante que pesa en contra de un hallazgo de uso justo.

En la misma línea de lo anterior, además de considerar la comerciabilidad de un uso, se verificó si el uso era transformador o no. Es importante mencionar que éste fue el factor que predominó en la decisión para considerar que el uso que le dio Google a las APIs de Java fue de manera justa.

En cuanto a las 37 APIs de Java, es un hecho público que para el momento en que Google sacó al mercado un nuevo sistema operativo para dispositivos móviles y tabletas, éstas no se utilizaban en teléfonos. Es decir, se pudo constatar que Android si fue transformador porque implementó con éxito el lenguaje Java y sus APIs en un contexto diferente, solo conservó el código declarativo y la SSO de las 37 APIs para asegurar la interoperación y basándose en la costumbre de la industria tecnológica respecto al acceso abierto de reimplementacion de las interfaces.

Para este factor, se desprende que el supuesto transformador pesó más que el uso comercial sobre las APIs de Java por parte de Google, y esto se debió a la importancia que tiene la innovación dentro de la industria del software y cómo el público puede beneficiarse de los avances tecnológicos que pueden utilizar elementos como los códigos declarativos, en este caso en específico,

las APIs de Java que fueron utilizadas para desarrollar Android, sistema éste que hasta el momento es el que domina el mercado en *smartphones*. Por lo que se considera que, este uso transformador compensó las finalidades del derecho de autor, usando Google las APIs de Java de forma justa y, por lo tanto, en lo que respecta a éste factor no se habría incurrido en violación del *copyright*.

Una consideración primordial que se observó en la evaluación de la naturaleza de las interfaces fue el carácter funcional que estas revisten, lo cual hace que este tipo de programas informáticos se considere menos creativos que otros, lo cual jugará a favor de los desarrolladores al poder calificarse como un uso justo cuando implementen el código declarativo y sus SSO de una interfaz o parte de estos.

Unas de las razones en las que se fundamentó Google para hacer uso de estas APIs fue, por un lado que, éstas fueron liberadas al público en general para su uso siguiendo una licencia de código abierto sin restricciones, como lo estimó Sun Microsystems en su oportunidad basándose en colaborar con la industria de *software* y, por otro lado, partiendo de esto, los programadores

invirtieron tiempo en aprender los lenguaje funcionales de Java para aplicarlos en la elaboración de programas informáticos, no quedando duda que en el mercado de tecnología las interfaces de Java son sustancialmente funcionales y unas de las más comunes y utilizadas por éstos.

Por lo que la reimplementación del código declarativo y las SSO de las 37 APIs de Java en el sistema operativo fue un uso justo, por cuanto con esto se facilitaba y resguardaba que otros desarrolladores programaran sus softwares sin tener que aprender otro lenguaje de programación.

Hay que considerar que es esencial que un programa informático nuevo que sale al mercado sea interoperable con otros programas, pues ello incidirá en su funcionalidad, utilidad y comercialización, lo que hace necesario que los desarrolladores utilicen una o varias interfaces para lograr dicha interoperabilidad. Esto se logra siempre y cuando se adhieran exactamente a las mismas especificaciones de la interfaz del primer programa, siendo ésta la que va a establecer las reglas acerca de cómo los programas van a trabajar juntos para efectuar tareas específicas, las cuales deberán ser idénticas y compatibles.

Lo anterior hace que las APIs sean primordiales dentro de la interoperabilidad de los programas informáticos y, en general, dentro del mundo globalizado, donde todo está intercomunicado, lo cual sin duda beneficia a los consumidores finales que adquieren o necesitan de la tecnología.

Si consideramos la asimilación de las APIs a una suerte de enchufe que permite la interoperabilidad, podemos decir que existen diversos tipos de enchufes a nivel mundial, como existen diversos tipos lenguajes de programación. Por ejemplo, podemos decir que en Chile el tipo de enchufe que se utiliza es el dominado "L"[127] por lo que los aparatos que se quieran utilizar en Chile deben usar este tipo de enchufes. En el caso de las interfaces de Java, si un programador quiere programar bajo el lenguaje Java debe usar la misma secuencia, estructura y organización de esas APIs, pues si no se copia línea por línea de este código no se puede utilizar aquella API y tendría que crear una nueva que nadie conocería, con lo que no existiría un estándar de interoperabilidad, es decir, cada sistema tendría un APIs diferente y no sería posible o resultaría muy dificultosa la comunicación en general.

[127] Según la Comisión Electrónica Internacional (CEI)

Dentro de la industria del *software*, es un hecho reconocido que el lenguaje Java es unos de los lenguajes más importantes y usados dentro de la programación. Es así que no se puede pasar por alto todo el tiempo que han invertido los programadores en el aprendizaje de este lenguaje, el cual brinda un estándar dentro la programación.

Observamos entonces, que, por la naturaleza funcional de las APIs, estas se materializan a través de una estructura, organización y secuencia de las líneas y son escritas mediante un lenguaje computacional, como es el caso de Java, lo cual exige la estandarización del lenguaje de programación de las APIs, y es esencial tanto para el conocimiento y aplicación de los programadores, como para que la tarea de la interoperabilidad resulte efectiva.

Ésta es una de las razones fundamentales para que Google haya utilizado las 37 APIs de Java, en virtud de que los desarrolladores que programan bajo el lenguaje Java ya conocen estas APIs, por cuanto estas conceden funciones básicas de uso cotidiano para la programación en la mayoría de los idiomas, por ejemplo, operaciones matemáticas (suma, resta, raíz cuadrada), que permiten la interoperabilidad entre diferentes programas, y para que estos

funcionen entre sí deben estar escritos en el mismo lenguaje de programación. Una decisión contraria a esta podría quebrantar la costumbre que se ha venido ejerciendo en la industria del software respecto a las interfaces.

En cuanto a la cantidad o sustancia que puede ser copiada de una interfaz, es importante señalar que cuando se utiliza el código declarativo y SSO un desarrollador implementa lo justo y necesario para lograr la compatibilidad con el otro programa, tal y como lo va a hacer el otro desarrollador del otro programa que va a interactuar con el primero, por lo que si se compara con todos los demás elementos o capas que conforman al programa nuevo, el porcentaje copiado siempre va hacer mínimo.

Partiendo entonces de que el código declarativo y las secuencias, organización y estructura de la interfaces son un método de operación, puede concluirse que si bien la decisión emitida sobre el uso justo a favor de Google de las 37 interfaces de Java fue positiva para la industria del software, no aplacó del todo la incertidumbre, pues, al fin y al cabo, la decisión que crearía estabilidad será aquella que considere a las interfaces como no protegidas por el

copyright, pues lo contrario generará demandas futuras, creando inseguridad dentro de esta industria a la hora de que los desarrollares implementen los códigos declarativos y sus SSO, que como se ha reiterado constituye el medio para lograr la interoperabilidad, teniendo en cuenta que la solución de cada caso dependerá del criterio de los jueces acerca del alcance la protección por el derecho de autor que, como se vio, no es uniforme.

Observamos entonces, que las interfaces son claves dentro de la programación para que se dé la interoperabilidad entre los programas informáticos, por lo que se debe tener mucho cuidado de no obstruirla, por cuanto esto al fin y al cabo pudiera afectar los intereses y el bienestar social y limitar la posibilidad de fomentar la innovación y el desarrollo tecnológico. Con ello lo que se lograría sería afectar a los consumidores finales de los programas y encerrarlos en un determinado estándar.

Por tal razón, mientras se mantenga el criterio de la Corte de Apelaciones Federal de que las interfaces están protegidas por el copyright y se tenga que decidir si su implementación es un uso justo o no, la futura decisión de esta Corte con ocasión al caso pendiente por resolver, será de suma importancia

tanto para todos los desarrolladores que trabajan a diario para crear nuevas tecnologías e innovaciones como para los usuarios de la industria del software. Una decisión distinta a la posición del jurado sostenida en la decisión aquí analizada, se insiste, ocasionaría un obstáculo a la innovación y estancamiento en los avances tecnológicos.

BIBLIOGRAFIA

1. AGUIRRE C., *Does The Fair Use Doctrine Constitute A Proper Balance Between The Rights Of Copyright Holders And The Rights Of The General Public, In Cases Of Appropriation Of Characters?*, University of Washington School of Law Seattle, Washington, p. 12. **[en línea]** <http://repositorio.educacionsuperior.gob.ec/bitstream/28000/1594/1/T-SENESCYT-00712.pdf>

2. ANTEQUERA Ricardo, "*Las Limitaciones y Excepciones al Derecho de Autor y los Derechos Conexos en el Entorno Digital*, "OMPI" [en línea] <http://www.wipo.int/edocs/mdocs/lac/es/ompi_sgae_da_asu_05/ompi_sgae_da_asu_05_2.pdf>

3. AMAYA J., *Sistema de Información Gerenciales*, pág. 31, [en línea] <http://www.rchdt.uchile.cl/index.php/RCHDT/article/viewFile/45900/48408>

4. ARIAS A. y otro, Curso de Programación y Análisis de Software, 2da Edición, p. 48 [en línea] <https://books.google.cl/books?id=2Wj0DAAAQBAJ&pg=PA52&dq=software+instrucciones&hl=es-419&sa=X&ved=0ahUKEwjz4Pe-gonVAhWHlJAKHfZtCyo4ChDoAQhCMAY#v=onepage&q=software%20instrucciones&f=false>

5. ASAY C., Transformative Use of Software, [en línea] <https://www.stanfordlawreview.org/online/transformative-use-in-software/>

6. BEEBE B., "An Empirical Study of U.S. Copyright Fair Use Opinions, 1978-2005, 156 Pennsylvania", Law Review. [en línea] <http://www.bartonbeebe.com/documents/Beebe%20-%20Empirical%20Study%20of%20FU%20Opinions.pdf>

7. BROCCA Juan y CASAMIQUELA René, Revista Pilquen Sección Ciencias Sociales año *VII No. 7, 2005, Las Licencias de Software desde la Perspectiva del Usuario Final,* [en línea] <http://www.scielo.org.ar/pdf/spilquen/n7/n7a12.pdf>

8. BROWN Franklin, "*The Incompatibility of Copyright and Computer Software: An Economic Evaluation and a Proposal for a Marketplace Solution*", North Carolina Law Review, [en línea] <http://scholarship.law.unc.edu/cgi/viewcontent.cgi?article=3166&context=nclr>

9. CANALES M., "*Oracle versus Google: Protección de derecho de autor sobre elementos funcionales de programas computacionales que ponen en riesgo la interoperabilidad y la innovación*", Revista Chilena de Derecho y Tecnología, Universidad de Chile. [en línea] <http://www.rchdt.uchile.cl/index.php/RCHDT/article/viewFile/37361/40376>

10. CETINA R., ¿De la sartén al fuego? La cuestión de la adopción del fair use como solución al sistema de límites al derecho de autor en México, [en línea] <http://aplicaciones.ccm.itesm.mx/virtualis/index.php/virtualis/article/viewFile/103/90>

11. CULEBRO JUÁREZ, M., GÓMEZ HERRERA, W. y TORRES SÁNCHEZ, S. *Software libre vs software propietario*, [en línea] <http://www.rebelion.org/docs/32693.pdf>

12. DEMASKY T., *North Carolina Journal of Law & Technology Volume 18, Issue on.: December* 2016 1 Oracle v. Google: *¿setting a standard or handicapping an industry?* [en línea] <file:///C:/Users/Neida%20Zambrano/Downloads/Demasky_Final%20(1).pdf>

13. FERNÁNDEZ J., *El Futuro de los **Límites** a los Derechos de Autor que Benefician y Apoyan la Investigación*, [en línea]

<http://enancib.ibict.br/index.php/enancib/vienancib/paper/viewFile/1801/942>

14. GARCIA E., *El papel del juez frente a las limitaciones y excepciones al derecho de autor en los países de tradición continental,* [en línea] <http://propintel.uexternado.edu.co/el-papel-del-juez-frente-a-las-limitaciones-y-excepciones-al-derecho-de-autor-en-los-paises-de-tradicion-continental/>

15. GONZÁLEZ B., 2013, El Debate Sobre La Interoperabilidad Informática En El Derecho De Autor Comunitario, Tesis doctoral, mención europea, [en línea] <file:///C:/Users/Neida%20Zambrano/Downloads/Dialnet-ElDebateSobreLaInteroperabilidadInformaticaEnElDer-46836%20(6).pdf>

16. GRIJALVA A., *Internet y derechos de autor*, [en línea] <http://www.flacso.org.ec/docs/sfintgrijalva.pdf

17. GOTT J., Lotus Development Corporation V. Borland International: The United States Court Of Appeals For The First Circuit Takes A Step Backward For The Copyright Protection Of Computer Programs, [en línea] <http://dspace.creighton.edu:8080/xmlui/bitstream/handle/10504/40205/49_30CreightonLRev1349(1996-1997).pdf?sequence=1>

18. GUERRERO G., *Aspectos Globales de la Patentabilidad de las Invenciones Implementadas por Ordenador*, *Estado actual y Nuevas Perspectivas*, [en línea] <http://revistas.uexternado.edu.co/index.php/propin/article/view/904/858>

19. IBÁÑEZ K. *Orígenes Del Fair Use,* Revista Chilena de Derecho y Tecnología, VOL. 2 NÚM. 2 (2013), [en línea] file:///C:/Users/Neida%20Zambrano/Downloads/30312-100879-2-PB%20(1).pdf>

20. MAZELLA M. y otro. 1 GEO. L., REV. 62 (2016), p. 66, [en línea] <https://perma.cc/X6WV-Z28A>

21. MONROY J., Organización Mundial de la Propiedad Intelectual (OMPI), Estudio sobre las Limitaciones o Excepciones al Derecho de Autor y los Derechos Conexos en Beneficio de las Actividades Educativas y de Investigación en América Latina y El Caribe, [en línea] <http://www.wipo.int/edocs/mdocs/copyright/es/sccr_19/sccr_19_4.pdf, >

22. MULLIN J., Google wins crucial API ruling, Oracle's case decimated, ARS Technica, [en línea] <https://arstechnica.com/tech-policy/2012/05/google-wins-crucial-api-ruling-oracles-case-decimated/>

23. NORTON P., *Introducción a la computación*, 6ta. Edición, [en línea] <https://alkedua.files.wordpress.com/2015/07/218982559-introduccion-a-la-computacion-peter-norton.pdf>

24. ODDI Samuel, "*An Uneasier Case for Copyright Than for Patent Protection of Computer Programs*", 1993, Nebraska Law Review [en línea] <http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1636&context=nlr>

25. ORGANIZACIÓN MUNDIAL DE PROPIEDAD INTELECTUAL, *La Protección Internacional del Derecho de Autor y de los Derechos Conexos*, [en línea] <http://www.wipo.int/export/sites/www/copyright/es/activities/pdf/international_protection.pdf>

26. ORGANIZACIÓN MUNDIAL DE PROPIEDAD INTELECTUAL (OMPI), *El derecho de autor y los derechos conexos en el entorno digital*, [en línea] <http://www.wipo.int/edocs/mdocs/lac/es/ompi_sgae_da_asu_05/ompi_sgae_da_asu_05_2.pdf>

27. ORGANIZACIÓN MUNDIAL DE PROPIEDAD INTELECTUAL. Tratado de la OMPI sobre Derecho de Autor, [en línea] http://www.wipo.int/treaties/es/text.jsp?file_id=295167#P74_6370>

28. ORGANIZACIÓN MUNDIAL DE PROPIEDAD INTELECTUAL, *Glossary Of Terms Of The Law Of Copyright And Neighboring Rights*. [en línea] <ftp://ftp.wipo.int/pub/library/ebooks/wipopublications/wipo_pub_816_efs-ocr-sp-image.pdf>

29. OWEN D., Interfaces and Interoperability in Lotus v. Borland: A Market-Oriented Approach to the Fair Use Doctrine, 64 Fordham L. Rev. 2381 (1996). [en línea] <http://ir.lawnet.fordham.edu/flr/vol64/iss5/8>

30. PETERS R., Ninth Circuit Holds That Intermediate Copying of Object Code Is Fair Use, 9 Santa Clara High Tech. L.J. 579 (1993)., p. 584 [en línea] <http://digitalcommons.law.scu.edu/chtlj/vol9/iss2/8>

31. REGINFO E. o "El papel del juez frente a las limitaciones y excepciones al derecho de autor en los países de tradición continental [en línea] <http://propintel.uexternado.edu.co/el-papel-del-juez-frente-a-las-limitaciones-y-excepciones-al-derecho-de-autor-en-los-paises-de-tradicion-continental/>

32. ROJO F., *Philosophical Foundations of the Fair Use Doctrine*, Escuela de Derecho, Universidad Torcuato di Tella, Argentina, [en línea] <file:///C:/Users/Neida%20Zambrano/Desktop/Carpeta%20provisional%20RED/Fair%20Use/Fundamentos%20filos%C3%B3ficos%20de%20la%20doctrina%20del%20fair%20use.html>

33. SAMUELSON P., "Three Fundamental Flaws in CAFC's Oracle v Google Decision" University of California, Berkerly, European Intellectual Propietary Review". [en línea]

<file:///Users/neidazambrano/Downloads/SSRNid2643840%20(2).pdf>

34. SAMUELSON P., The Strange Odyssey of Software Interfaces and Intellectual Property Law UC Berkeley, p [en línea] <http://repositories.cdlib.org/bclt/lts/59>

35. SAMUELSON P., The Uneasy Case for Software Copyrights Revisited, 79 Geo. Wash. L. Rev. 1746 (2010). [en línea] <http://scholarship.law.berkeley.edu/facpubs/1545>

36. SAMUELSON P., *Fair Use For Computer Programs And Other Copyrightable Works In Digital Form: The Implications Of Sony, Galoob And Sega*, p. 57-58, [en línea] https://www.law.berkeley.edu/php-programs/faculty/facultyPubsPDF.php?facID=346&pubID=147>

37. *Stanford University Libraries, Copyright and fair use*, [en línea] <http://fairuse.stanford.edu/overview/fair-use/what-is-fair-use/>

38. SIG M., *Predicting Fair Use, kb.osu.edu Ohio State Law Journal*: Volume 73, Issue 1 (2012), Ohio State Law Journal, vol. 73, no. 1 (2012), p.67, [en línea] <http://hdl.handle.net/1811/71532>

39. TREVOR M, J*ury Returns Verdict for Google In Question of Fair Use of Oracles Code, Journal of Intellectual Property Law & Practice* 2016; [en línea] <*doi: 10.1093/jiplp/jpw112*>

Leyes y decisiones

40. BIBLIOTECA DEL CONGRESO NACIONAL DE CHILE (BCN), [en línea] https://www.leychile.cl/Navegar?idNorma=28933

41. CONVENIO DE BERNA, [en línea] la página de la OMPI: <http://www.wipo.int/export/sites/www/treaties/en/documents/pdf/berne.pdf

42. LOTUS DEVELOPMENT CORP. V. BORLAND INTERN., Inc., 49 F.3d 807 (1995), párrafo 819, [en línea] <https://scholar.google.com/scholar_case?case=9888762079230732186&q=Lotus+v.+Borland,+49+F.3d+807+(1995)&hl=en&as_sdt=40000006

43. ORGANIZACIÓN MUNDIAL DE PROPIEDAD INTELECTUAL (OMPI), *Copyright Act de los Estados Unidos.* [en línea] <http://www.wipo.int/wipolex/es/details.jsp?id=5405

44. SEGA ENTERS. V. ACCOLADE, INC., CORTE DE APELACIONES NOVENO CIRCUITO, 1992, [en línea] <https://www.law.berkeley.edu/files/sega.pdf

45. SONY CORPORATION OF AMERICA V. UNIVERSAL CITY STUDIOS, INC, [en línea] <http://www.law.sc.edu/orientation/2015/case.pdf y/o <http://picker.uchicago.edu/seminar/Class02.pdf

46. *THE UNITED STATES DISTRICT COURT FOR THE NORTHERN DISTRICT OF CALIFORNIA NOTICE OF FINAL CHARGE TO THE JURY (PHASE ONE) AND SPECIAL VERDICT FORm*, en el caso Oracle America INC v. Google INC., p. 15 [en línea] <https://es.scribd.com/doc/313560685/Jury-Instruction-Oracle-v-Google

47. *THE UNITED STATES DISTRICT COURT FOR THE NORTHERN DISTRICT OF CALIFORNIA*, caso: Oracle v. Google, [en línea] <https://www.eff.org/files/2014/11/10/judge_alsup_ruling_that_java_apis_uncopyrightable.pdf (Traducción libre de la autora)

48. *THE UNITED STATES DISTRICT COURT FOR THE NORTHERN DISTRICT OF CALIFORNIA*, caso: Oracle v. Google, [en línea] <https://assets.documentcloud.org/documents/2856497/06-08-16-Order-Denying-Rule-50-Motions.pdf

49. *THE UNITED STATES DISTRICT COURT FOR THE NORTHERN DISTRICT OF CALIFORNIA, ORACLE AMERICA, INC., PLAINTIFF, V. GOOGLE INC., DEFENDANT.* / No. C 10-03561 WHA, [en línea] <https://www.eff.org/files/alsup_api_ruling.pdf

50. TRATADO DE LA OMPI SOBRE INTERPRETACIÓN O EJECUCIÓN Y FONOGRAMAS. [en línea] la página de la OMPI: <http://www.wipo.int/export/sites/www/treaties/en/documents/pdf/wppt.pdf

51. *UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT, BRIEF OF COMPUTER SCIENTISTS AS AMICI CURIAE IN SUPPORT OF DEFENDANT-APPELLEE*, Caso Oracle v. Google. Nos. 2017-1118, 2017-1202, [en línea] <https://www.eff.org/files/2017/05/31/2017.05.30_computer-scientists-fair-use-amicus-brief_oracle_v_google.pdf

52. *UNITED STATES COURT OF APPEALS FOR THE FEDERAL CIRCUIT* BRIEF OF AMICUS CURIAE MOZILLA URGING AFFIRMANCE OF THE JUDGMENT, [en línea] < https://www.eff.org/es/document/mozilla-amicus-brief