

UNIVERSIDAD AUSTRAL  
MAESTRIA EN PROPIEDAD INTELECTUAL  
Junio 2022

ORACLE AMERICAS INC  
v  
GOOGLE LLC

Gustavo G. Mármol Alioto  
Abogado

Este trabajo se licencia bajo los términos  
CC BY-NC-ND 4.0

Atribución/Reconocimiento-No Comercial-Sin Derivados 4.0 Licencia Pública  
Internacional

El presente trabajo constituye una obra derivada de la segunda edición de la  
obra denominada “El caso judicial Oracle America, Inc. c. Google, LLC. No. C  
10-03561 WHA Copyright © 2022 Gustavo G. Mármol Alioto” y  
de su proyecto de desarrollo colaborativo disponibles en

<https://drive.google.com/drive/folders/1LJrWdizsSCCyyUZHNiHoLFPsSOJMS>

Bh

y

<https://github.com/ggmarmolalioto>

# Temario

Introducción. Principio de división expresión/idea. Programas de Computación. El litigio bajo el prisma del licenciamiento Open Source

Java. Lenguaje de programación y Plataforma Java SE. El principio “*write once, run anywhere*”. Librería de Software Estándar de Java de Sun Microsystems/Oracle.

Licenciamiento de la Plataforma Java SE: Licencia open source GPL v2 + Excepción ClassPath, Licencia Especificación y Licencia Comercial. Test de Compatibilidad de Java TCK

Juicio I **Protegibilidad de las API(s) de Java y su estructura, secuencia y organización (SSO) bajo el Sistema de Derechos de Autor de EE.UU** (Agosto 2010- 26 de Mayo de 2015): Argumentos de Oracle, Argumentos de Google, Sentencia Corte Distrito (DC) y Sentencia Corte Apelación Federal (CAFC)

Juicio II **Limitación a los derechos exclusivos Defensa de “Fair Use” (uso justo o legítimo)** (8 de Junio de 2016-15 de Noviembre de 2019 -aceptación del Writ of Certiorari II de Google). Sentencia Corte Distrito y Sentencia (DC) Sentencia Corte Apelación Federal (CAFC)

Sentencia Corte Suprema de los EE.UU (5 de Abril 2021). Voto de la Mayoría. Voto de la Minoría.

# Expresión/ideas. Programas de Computación.

Programas de computación: código fuente (humano) y código objeto (funcional). Estructura y Organización.

Principio “expresión de las ideas” y “no de las ideas en si mismas”.

¿Cómo debe interpretarse el principio idea/expresión cuanto se trata de líneas de código fuente escritas en cualquier lenguaje de programación?

¿Un método para operar puede ser expresivo?, y por ende protegible por derechos de autor?, o automáticamente quedaría excluido de protección por ser un “método para operar”?

¿Qué sucedería cuando la sintaxis es obligatoria por imposición del lenguaje de programación?

¿Qué sucedería cuando una idea expresada en líneas de código (función requerida) sólo pueden ser expresada (escrita) de una sola manera?

¿Cuándo quedaría fusionada la idea y la expresión de la idea en líneas de código?

# El litigio bajo el prisma del licenciamiento Open Source

- Sun Microsystems/Oracle licenciaron las API(s) de Java bajo la licencia GPLv2 más ClassPath Exception, Especificación (TCK) y Comercial Binaria.
- Google no basó su implementación de las API de Java en la versión open source llamada OpenJDK (GPLv2)
- Google basó su implementación en Apache Harmony, la implementación independiente de la Fundación Apache, licenciada bajo la Licencia Apache v2
- Conforme Sun/Oracle la Fundación Apache nunca obtuvo la aprobación del TCK de Sun/Oracle, y por ende, nunca podía haber distribuido la implementación Apache Harmony
- Google licencia Android bajo la Licencia Apache v2
- Desde la versión 7.0 llamada Android Nougat ([marzo de 2006](#)), Google decidió basar su implementación de las API de Java en la versión open source de Oracle, llamada OpenJDK, bajo licencia GPLv2.
- No constituye el típico caso de infracción de IP, porque Google nunca negó que había copiado el código declarado de los 37 paquetes de las API de Java y su estructura secuencia y organización (SSO)

# Java

- Lenguaje de programación
- Plataforma de desarrollo y ejecución de programas escritos en lenguaje Java
- Por un lado, Java es el **lenguaje de programación** mediante el cual un programador escribe código fuente en lenguaje Java.

```
public abstract void verify (PublicKey key, String sigProvider)
    throws CertificateException, NoSuchAlgorithmException,
    InvalidKeyException, NoSuchProviderException, SignatureException
```

# Plataforma Java SE “*Write Once Run Anywhere*” (WORA). Máquina Virtual de Java (JVM)

Por otro, Java constituye una **plataforma** de software que permite ejecutar programas de software escritos en el lenguaje de programación Java.

Java **agnóstico a la tecnología** sobre la cual se ejecutara el programa de computación escrito en Java

Los programadores no tendrían que escribir versiones distintas de programas de computación para diferentes sistemas operativos o hardware

La plataforma Java a través de la utilización de la máquina virtual de Java permite a los desarrolladores escribir programas que son capaces de ejecutarse en diferentes computadoras o hardware sin tener que reescribir sus programas para cada tipo diferente de computadora.

Esto se conoció como el “write once, run anywhere” (WORA).

En definitiva, un programa escrito en Java podría correr o ejecutarse en cualquier sistema operativo o hardware, siempre que tuviese instalado la máquina virtual de Java

# Plataforma Java SE: Composición. Versiones.

-Herramientas de desarrollo de código (JDK)

-Compilador de Java

-Ambiente de ejecución de Java:

Máquina virtual de Java (JVM),

**Librería de Software Estándar de Java**

-Documentación (para cada versión de Java)

JDK 1.0 (Oak) (Enero de 1996) ([8 paquetes](#)), JDK 1.1. (Febrero de 1997), J2SE 1.2 (Playground) (Diciembre de 1998), J2SE 1.3 (Kestrel) (Mayo de 2000), J2SE 1.4 (Merlin) (Febrero de 2002), [J2SE 5.0 o 1.5 \(Tiger\) \(Sept. de 2004\) \(166 paquetes\)](#), Java SE 6 o 1.6 (Mustang) (Diciembre de 2006), Java SE 7 o 1.7 (Dolphin) (Julio de 2011) ([209 paquetes](#)), Java SE o 1.8 (Marzo de 2014), Java SE 9 (21 de Septiembre de 2017), Java SE 10 (20 de Marzo de 2018), Java SE 11 (25 de Septiembre de 2018), Java SE 12 (19 de Marzo de 2019), Java SE 13 (17 de Septiembre de 2019), Java SE 14 (17 de Marzo de 2020), Java SE 15 (15 de Septiembre de 2020), Java SE 16 (16 de Marzo de 2021), Java SE 17 (14 de Septiembre de 2021), **Java SE 18 (Marzo 22 de 2022)**

# Librería de Software Estándar de Java

- Contiene programas pre-escritos (“*built in packages*”)
- Pueden **ser utilizados o no** por los programadores para escribir sus propios programas (excepto `java.lang`, `java.io` y `java.util`)
- Cada paquete de Java posee **dos (2) tipos relacionados** de código fuente: **código declarado** (*method header*): encabezado del método o declaración: especifica los nombres, parámetros y funcionalidad de los métodos de java y **código implementado** (*method body*): bloque de código que implementa el método instruyendo a una computadora como ejecutar una determinada función.
- Los programadores para reutilizar una función no necesitan conocer el código implementado, sólo deben conocer el comando abreviado que está basado en la sintaxis del código declarado.

# Ejemplo de código declarado y código implementado

código declarado

```
package java.lang;           (declaración del paquete java.lang)  
public class Math {          (declaración de la clase Math)  
    public static int max (int x, int y) { (declaración del método max)
```

código implementado

```
        if (x > y) return x;      (implementación, devuelve x, o)  
        else return y;         (implementación, devuelve y)  
    }                         (cierra el método)  
}
```

(cierra la clase)

# Librería de Software de Java: SSO

Paquete de Java                    **java.lang**

Clase de Java                      **Math**

Método de Java                    **max**

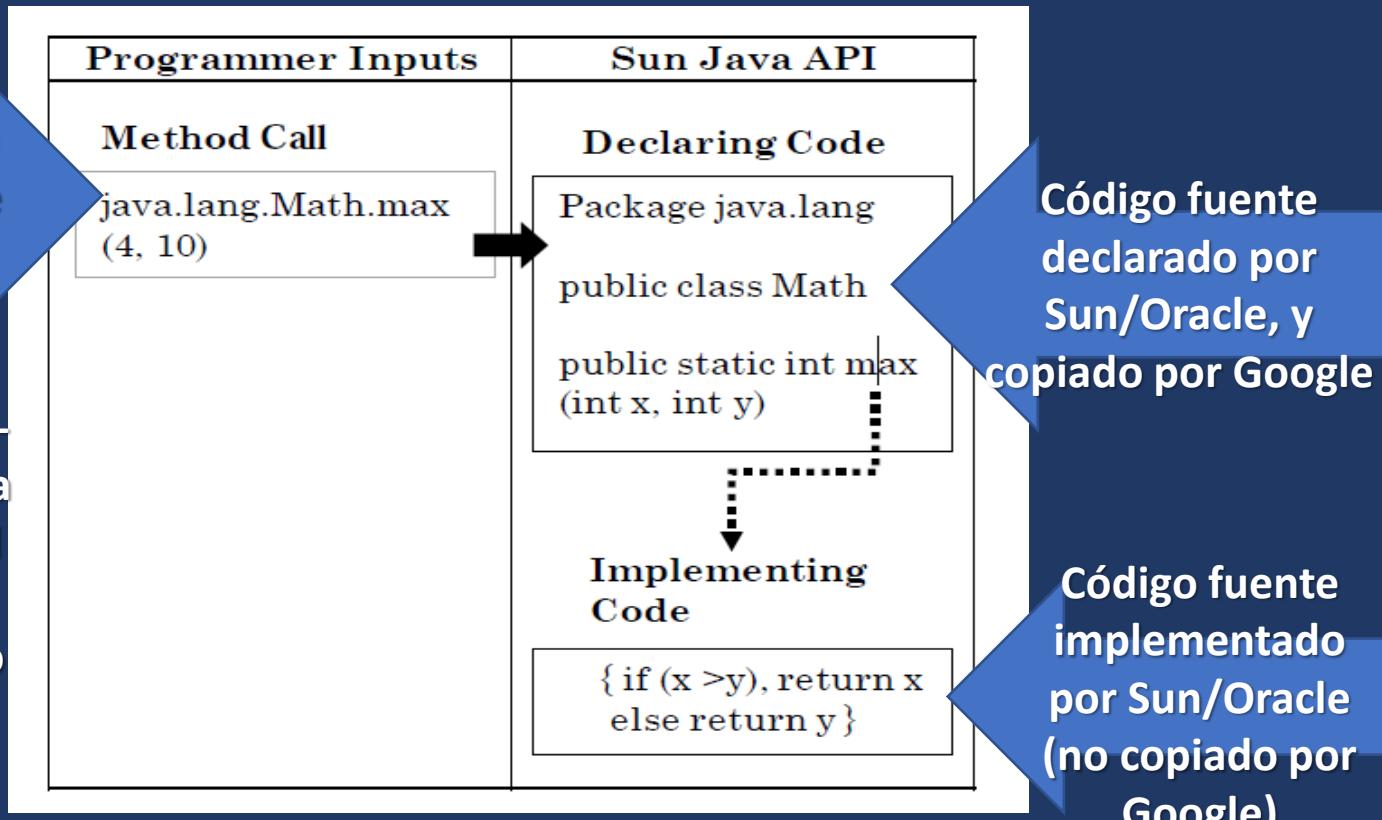
-Los paquetes agrupan a las clases, y las clases agrupan a los métodos de Java

-Si un programador quiere reutilizar en su programa en construcción la función **max** debe utilizar un formato de comando determinado (dándole los parámetros deseados)

**Java.lang.Math.max ()**

# Librería de Software de Java (gráfico según la Corte Suprema de EE.UU. Sentencia Abril 5 de 2021)

Formato de comando -líneas de código- usadas por los programadores para invocar el código declarado



# Licenciamiento de la Plataforma Java SE

- **Licencia de GPLv2 más Excepción ClassPath:** código declarado + código implementado (Sun/Oracle)
- **Licencia de Especificación:** código declarado únicamente
- **Licencia Comercial:** código declarado + código implementado (por Sun/Oracle)

# Test de Compatibilidad de Java SE (TCK). Requisitos.

- Tanto la licencia de especificación como la licencia comercial debían pasar o aprobar el Test de Compatibilidad de Java (TCK) requerido por Sun Microsystems/ Oracle.
- implementación completa de la versión actual de la especificación sin hacer “*subsetting*” o “*supersetting*”
- Implementar todas las interfaces y funcionalidades del paquete requerido por la J2SE sin hacer “*subsetting*” o “*supersetting*”
- No agregar ningún paquete, clase o interface a los paquetes `java.*` o `javax.*` o a los sub-paquetes de estos
- Que la implementación independiente no contenga código fuente u materiales binarios de Sun/Oracle

# Juicio I

Protegibilidad de las API(s) de Java y  
su estructura, secuencia y  
organización (SSO) bajo el Sistema de  
Derechos de Autor de EE.UU

Agosto 2010 a Mayo 2015

# Obra Protegida en Discusión

**37 Paquetes de la Librería de Software Estándar de Java de Java2 SE 5.0 (Tiger):**

- ✓ La **copia literal** (idéntica) de **elementos literales** (código fuente declarado, -o funciones de prototipos, o *header file* o declaraciones de interface- o API(s) de los 37 paquetes de Java2 SE 5.0)
- ✓ La **copia no literal** (no idéntica) de los **elementos no literales** de los 37 paquetes de Java SE 5.0 (estructura, secuencia y organización) (SSO)

# Los 37 Paquetes de la Librería de Clases de Java (incluidos en la versión 5.0 de Java SE)

java.awt.font,      java.beans,      java.io,      java.lang,  
java.lang.annotation,    java.lang.ref,    java.lang.reflect,  
java.net,                java.nio,                java.nio.channels,  
java.nio.channels.spi,                            java.nio.charset,  
java.nio.charset.spi,    java.security,    java.security.acl,  
java.security.cert,                            java.security.interfaces,  
java.security.spec,    java.sql,      java.text,      java.util,  
java.util.jar,            java.util.logging,    java.util.prefs,  
java.util.regex,          java.util.zip,        javax.crypto,  
javax.crypto.interfaces,    javax.crypto.spec,    javax.net,  
javax.net.ssl,            javax.security.auth,    javax.  
security.auth.callback,                            javax.security.auth.login,  
javax.security.auth.x500, javax.security.cert, and javax.sql

# Ejemplo de código fuente declarado (denominación utilizada en el juicio)

## **java.security.cert.package**

```
public abstract void verify (PublicKey key, String sigProvider)  
throws CertificateException, NoSuchAlgorithmException,  
InvalidKeyException, NoSuchProviderException  
SignatureException
```

# Objeto de la Disputa: Aspecto “Literal”

- ✓ Aspecto “Literal” (+ 7000 líneas de código *per se* en Java2 SE 5.0)
- ✓ **CENTRADO SOLAMENTE** en las declaraciones de interface o API(s) de Java (código fuente declarado) de 37 (de un total de 166) paquetes que conforman la Librería Estándar de Software de Java
- ✓ **NO ESTA EN DISCUSIÓN** la implementación de las API(s) de Java realizada por Sun Microsystems -código fuente implementado de Sun/Oracle-, ya que Google realizó su propia implementación)

# Objeto de la Disputa: Aspecto “Literal” (líneas de código *per se*)

## código fuente declarado

package java.lang; (declaración del paquete java.lang)

public class Math { (declaración de la clase Math)

public static int max (int x, int y) { (declaración del método max)

## código fuente implementado

if (x > y) return x; (implementación, devuelve x, o)

else return y; (implementación, devuelve y)

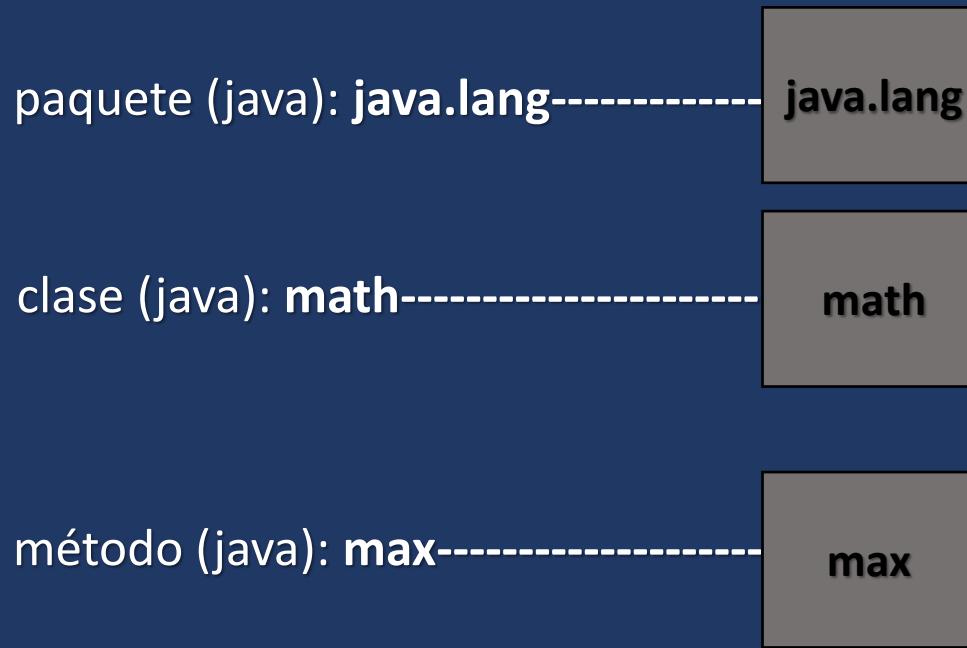
}

(cierra el método)

}

(cierra la clase)

# Objeto de la Disputa: Aspecto “No Literal” (SSO de 37 paquetes de Java)



# Objeto de la Disputa: Aspecto “No Literal” (SSO de los paquetes de Java)

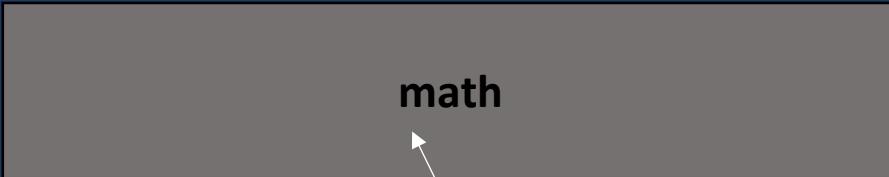
Clases en  
**Java.lang**

**java.lang**

Boolean, Byte, Character, Character.Subset,  
Character.UnicodeBlock, Class, ClassLoader, ClassValue,  
Compiler, Double, Enum, Float, InheritableThreadLocal,  
Integer, Long, Math, Number, Object, Package, Process,  
ProcessBuilder, ProcessBuilder.Redirect, Runtime,  
RuntimePermission, SecurityManager, Short, StackTraceElement,  
StrictMath, String, StringBuffer, StringBuilder, System, Thread,  
ThreadGroup, ThreadLocal, Throwable, Void

# Objeto de la Disputa: Aspecto “No Literal” (SSO de los paquetes de Java)

Métodos de  
Java  
dentro de la  
clase  
**math**



abs(double a), abs(float a), abs(int a), abs(long a), acos(double a), atan(double a), atan2(double y, double x), asin(double a) cbrt(double a), cos(double a) y **max** (int a, int b)

# Argumentos principales de Oracle

En **Agosto de 2010** Oracle demanda a Google por infracción a sus derechos de autor y patentes de invención

Oracle alega que al copiar Google los 37 paquetes de la Librería de Software Estándar de Java de Java2 SE 5.0 (Tiger) violó los derechos de autor y patentes de invención de Oracle

La construcción de las API de Java constituye un “**proceso creativo**” y por ende **protegible bajo derechos de autor (EEUU)**

- Comienza con la identificación de un área de necesidad o de nuevas funciones por parte de los desarrolladores
- Parte de la creatividad esta en “que incluir” en cada paquete de Java y como organizar el código declarado de Java para que el uso de parte de los programadores sea “intuitivo”
- Las líneas de código declarado **están expresadas** en palabras, números u otros símbolos verbales o numéricos, y es **original** (i) porque fue creada por Sun/Oracle de forma independiente, y (ii) porque Sun/Oracle podrían haberlas expresado de una manera diferente.

# Argumentos principales Oracle

- Oracle argumenta que la Fundación Apache no podría haber distribuido su implementación independiente de la especificación de Java SE 5.0 llamada Apache Harmony, teniendo en cuenta que la Fundación Apache no había obtenido la aprobación del TCK de Sun/Oracle.
- Término API (aplicación de programación de interface): Oracle sostiene que el término API es un “camaleón” ya que puede describir un protocolo de comunicación trivial, como pasar información entre dos programas de computación, o puede describir programas sofisticados como los escritos por Sun Microsystems objeto del juicio.
- Al copiar Google en forma literal el código declarado también copio la estructura, secuencia y organización de los 37 paquetes de Java.

# Argumentos principales de Google

-Google reconoce que copió los 37 paquetes de las API(s) de Java y su SSO pero desarrollando su propia implementación independiente, y afirma que tanto las declaraciones de interface como su SSO constituyen un “método o procedimiento de operación” excluidas de protección bajo el artículo 102 inciso b) de la ley de derechos de autor de los EE.UU (Título 17, Copyright Act 1976).

-Término API. Google sostiene que la terminología “código declarado” es incorrecta, que no constituye un término en las ciencias de computación, y que lo correcto sería hablar de “declaraciones”, en parte porque las declaraciones no constituyen “código”, ya que no pueden ejecutarse por una computadora, y su única función está en establecer como dos programas se pueden comunicar.

-Google sostiene que las declaraciones de interface API(s) constituyen la “idea”, y el código implementado la “expresión” de la idea.

-Google expresa que su uso en la reimplementación de las declaraciones de interface de Java no se basan en ninguna licencia de ningún tipo de Sun/Oracle, porque su uso se basa en el *fair use*, y por lo tanto, no se requiere ningún tipo de licencia o autorización de Sun/Oracle.

# Argumentos principales de Google

- Google basó su implementación de las API(s) de Java en la implementación e las API(s), de la Fundación Apache, Harmony.
- Google sostiene que la Fundación Apache podía –como lo hizo– distribuir su propia implementación independiente, ya que era su propio código fuente implementado (y no era de Sun/Oracle), lo que no podía hacer la Fundación Apache era denominar a Apache Harmony como “Java compatible”.
- Google sostiene que para que los programadores en Java pudieran usar el conocimiento adquirido en Java SE (el sistema de comandos) en la Plataforma Android, Google debió copiar las declaraciones de interface usadas por Sun/Oracle, de lo contrario, aquel sistema de comandos aprendido no podría ser utilizado por los programadores.

# Sentencia Corte de Distrito (Mayo 2012)

-Reverso el veredicto del jurado y **declaro** que los paquetes de las API(s) de Java y su SSO no son protegibles bajo el sistema de derecho de autor de EEUU

## Argumentos:

-Implementación independiente de Google

-Estructura, Secuencia y Organización: No importa que sea creativa si constituye un método de operación (art 102, inc. b)

-Declaraciones de Interfaces impuestas por reglas del Lenguaje Java

-Doctrina de la Fusión (respecto del código declarado únicamente, no de la SSO) (expresión/idea): una sola forma de expresar.

-No hay protección bajo derechos de autor de EEUU para nombres y frases cortas

-Interoperabilidad: hay necesidad de duplicar la estructura de comandos de Java

-Fragmentación. Los casos SONY y SEGA (ambas del 9no Circuito de Apelación)

-Johnson Control v. Phoenix Control Sys.: en el Circuito 9no la SSO es protegible según los hechos particulares del caso (es decir no siempre y en todos los casos)

-Porcentaje: el 97% de las líneas de código de los 37 paquetes fueron escritas por Google, y el 3% restante eran replicables bajo la teoría de la fusión y de nombres cortos

-Elementos particulares del caso

# Apelación de ORACLE ante la CAFC

- Axioma 1:** La ley de derechos de autor de EEUU protege la “expresión original”
- Axioma 2:** La protección de derechos de autor se extiende a los programas de computación, de la misma manera que para cualquier otra obra literaria.
- Se protege a las obras literarias expresadas en: **palabras, números, símbolos verbales o numéricos.**
- Los programas de computación cumplen con dicha definición
- Lo expresado por el Juez de la Corte de Distrito que existe un tipo de software (código fuente declarado) que esta por fuera de estas reglas constituye una violación a los principios de protección bajo derechos de autor de EEUU.
- El código declarado es protegible porque esta “expresado”.
- No existen convenciones de programación o reglas dictadas por el lenguaje Java acerca de lo que el código declarado debe o no ser.
- Frases cortas. No se debe diseccionar la obra, tampoco poner foco en la línea individual de código, sino en su arreglo del que forma parte, de lo contrario ningún programa de computación estaría protegido.

# Apelación de ORACLE ante la CAFC

- La SSO fue reconocida por la Corte de Distrito como original y creativa, por tanto es protegible en dos sentidos a) código declarado incorporado en ella, y b) por su propia SSO, independiente// de que el código declarado incorporado a ella lo sea.
- Como existieron infinitas opciones para Oracle (i) de como organizar y estructurar las API(s) de Java, y (ii) de como denominar a cada línea individual de código declarado (en lugar de **java.lang.Math.max** podría haber elegido **Arith.bigger**, **Math.maximum**), “la idea de como organizar cada paquete de Java o de como llamar a cada línea individual de código declarado no se fusiona con su expresión”, por tanto el art. 102 inciso b) no es aplicable.
- Google reconoció que la SSO de Android es sustancialmente la misma a la SSO de las 37 API(s) de Java copiada.
- La doctrina de la fusión (“opciones de expresar” para que idea y expresión no se fusionen) se debe evaluar al momento de la creación de la obra intelectual y no al momento de la copia.
- Una vez que el autor eligió crear y denominar a un método de Java como **java.lang.Math.max** lógicamente que quien quiera usar la funcionalidad debe llamarlo por su nombre.

# Estructura de la Corte de Apelación de los EE.UU.

La Corte de Apelación de los EE.UU es la Corte más influyente después de la Corte Suprema de EE.UU.

Se divide en 11 “Circuitos de Apelaciones”. Cada Circuito abarca varios Estados.

Cada Circuito de Apelación recibe las apelaciones de las Cortes de Distrito de su propio Circuito.

Las opiniones de cada Circuito de Apelación se aplica sólo en “su” Circuito.

El **Círculo 9no (San Francisco)** incluye el Estado de California (del Oeste, Este, Norte, Central y Sur), Montana, Nevada, Arizona, Alaska, Idaho, Hawái, Oregon, Washington (del Este y Oeste) entre otros.

**Corte de Apelaciones Federal:** recibe todas las apelaciones de todos los Distritos de todos los circuitos en **materia federal** (patentes, marcas, veteranos de guerra, contratos con gobierno, etc.)

Debido a que la demanda de Oracle incluía asuntos de infracción de patentes, quedó establecido procesalmente que la Corte Federal de Apelación (CAFC) sería la instancia de apelación, y no así la Corte de Apelación del Circuito Noveno. La CAFC no suele revisar asuntos de infracción de derechos de autor de EEUU. En el caso, la CAFC **debió aplicar** el “derecho establecido” -conforme casos anteriores- por la Corte de Apelación del Circuito Noveno.

# Sentencia I de la Corte Federal de Apelaciones (CAFC) (9 de Mayo de 2014)

-Determinó que las API(s) de Java y su SSO son protegibles bajo el sistema de derecho de autor de EE.UU.

-El derecho de autor de EE.UU protege a las obras originales que hayan sido fijadas en un medio de expresión tangible.

-los programas de computación se definen como: *"el set de declaraciones o instrucciones para ser usados directa o indirectamente por una computadora con el objeto de producir un resultado cierto"*.

-Los programas de computación se protegen bajo derechos de autor de EEUU como obras literarias.

-La obra debe ser original, y la originalidad no es exigente.

-Originalidad tiene un sentido propio: que haya sido creada en forma independiente, que no sea copia de otra.

-La protección sólo se extiende a la expresión de las ideas, y no a las ideas en si mismas.

# Sentencia I de la Corte Federal de Apelaciones (CAFC) (9 de Mayo de 2014)

-La protección por derechos de autor se extiende a los **elementos literales** (código fuente y código objeto), y a los **elementos no literales** (la estructura secuencia y organización del programa de computación)

- Cuando Google copió el código declarado copio la estructura, secuencia y organización de los 37 paquetes de las API(s) de Java

-Año 2005: Google adquirió Android Inc. con la idea de desarrollar una plataforma para teléfonos móviles inteligentes, Google comenzó a negociar con Sun una licencia para usar y adaptar Java para dispositivos móviles.

-Google no quería hacer su implementación compatible con Java. Como eso era contrario al WORA, Sun decidió no otorgar una licencia a Google sobre las paquetes de las API de Java. Google desarrollo su Plataforma Android con 168 paquetes, 37 de los cuales están en juicio.

- Google creyó que los desarrolladores esperarían encontrar el mismo set de funcionalidades con el mismo formato de llamada de Java en Android, por ello copió el código declarado de los 37 paquetes de las API de Java y lo introdujo en Android. Al hacer ello, también copió la SSO de los 37 paquetes de Java.

# Sentencia I de la Corte Federal de Apelaciones (CAFC) (9 de Mayo de 2014)

-La protección por derechos de autor se extiende a los **elementos literales** (código fuente y código objeto), y a los **elementos no literales** (la estructura secuencia y organización del programa de computación)

-Juez de Distrito y Google reconocieron que los 37 paquetes de las API(s) de Java y su SSO era creativo y original, por tanto Google y Oracle estaban de acuerdo en se cumplía con el requisito del art. 102 inciso a) originalidad, pero no estaban de acuerdo en la interpretación del artículo 102, inciso b).

-Artículo 102 inciso b): “En ningún caso el derecho de autor para la autoría de una obra original se extiende a las **ideas, procedimientos, procesos, sistemas, métodos de operación, conceptos, principios o descubrimientos**, sin importar la forma en las cuales estas sean descriptos, explicados ilustrados o incorporados en la obra en cuestión”

-Google sostiene que hay dos (2) pasos en el análisis de protección bajo derechos de autor: en donde el artículo 102 inciso a) concede protección a las obras originales, y el b) quita la protección a la obra si ésta posee elementos funcionales.

-CAFC expresa que la postura de Google es incorrecta, contrario a lo dispuesto por el Congreso de los EE.UU.

# Sentencia I de la Corte Federal de Apelaciones (CAFC) (9 de Mayo de 2014)

-Conforme la interpretación del Congreso de EEUU el art 102 inciso b) “de ninguna manera extiende o contrae el ámbito de protección conferido por el derecho de autor”...”su propósito es reafirmar...que el principio básico entre la división de idea y expresión se mantiene inalterado”

-El art 102 inc. b) no extingue la protección acordada a la expresión de una idea por la sola circunstancia que dicha expresión se haya incorporado en un método de operación.

-Afirma la CAFC que conforme el derecho establecido en el Circuito Noveno a los fines de determinarse si los elementos no literales de un programa de computación constituyen una expresión protegible se debe utilizar el test desarrollado por el Circuito Segundo, llamado abstracción, filtración y comparación.

-Este examen (AFC) descarta que por el solo hecho de que cualquier cosa ejecute una función no pueda ser protegido, es decir, aunque un elemento de una obra pueda ser caracterizado como un método de operación, ese elemento, puede sin embargo contener un grado de expresión que lo haga elegible para ser protegido por derecho de autor.

-CAFC expresa que la Corte de Distrito se equivoca en aplicar la teoría de la Fusión

# Sentencia I de la Corte Federal de Apelaciones (CAFC) (9 de Mayo de 2014)

-Frases cortas y nombres: al diseccionarse en forma individual las líneas de código declarado en frases cortas, el CD erró en reconocer una combinación original de elementos que puede ser protegido.

-Doctrina Scene a Faire: No se demostró por Google que agrupar a los métodos de java dentro de las clases, y estas en paquetes constituyen una práctica común y esperada.

-Para determinar que la SSO no era protegible, más allá de que había reconocido su creatividad, el DC sostuvo que la SSO era un sistema para operar o método de operación conforme art. 102 inc. b), y basó su decisión en el caso Lotus Development v Borland International.

-CAFC menciona que el CD se equivoca al fundar el caso OvG en Lotus, porque Lotus es distinto en cuanto a los hechos al caso OvG: (i) Borland no había copiado código de Lotus, y en OvG Google reconoció que copio el código declarado de Oracle en forma idéntica, (ii) los comandos de Lotus no eran creativos, y en OvG se dijo que tanto las API(s) como la SSO de los paquetes de Java eran creativos y originales, (iii) en Lotus los comandos eran esenciales para operar el sistema de Lotus 1-2-3, pero en OvG Google no necesitaba copiar la SSO de las API de Java para escribir programas en Java.

-La CD debería haber aplicado el derecho establecido por el Circuito Noveno (Johnson Controls), y no Lotus, que es inconsistente con el derecho establecido en el Circuito Noveno.

# Sentencia I de la Corte Federal de Apelaciones (CAFC) (9 de Mayo de 2014)

-CAFC sostiene que en el caso Lotus se sostuvo que la expresión que es parte de un método de operación no es protegible por el derecho de autor, y que el derecho establecido en el Circuito Noveno que es el aplicable a OvG se estableció lo opuesto, ya que el derecho de autor protege la expresión de una método de operación.

-CAFC sostiene que una obra original –incluso aquella que su propósito sea una función- se encuentra bajo la protección del derecho de autor siempre que el autor posea multiples formas de expresar la idea.

-**Interoperabilidad:** la CD sostuvo que Google debió copiar el código declarado y sus estructura para garantizar la interoperabilidad con el sistema de llamada de comandos **java.package.Class.method()**. Para llegar a esta conclusión el Juez de la CD se basó en los casos del Circuito Noveno: Sega Enterprises v Accolade (1992) y Sony Computer v. Connectix (2000). Interpretó que estas dos decisiones sostuvieron que los procesos de interface que son necesarios copiar con el objeto de lograr compatibilidad son aspectos funcionales que no están protegidos por el derecho de autor conforme art. 102, inc. b).

-CAFC sostiene que ambos casos (Sega y Sony) se relacionan con uso justo y no con la protección de la obra.

# Sentencia I de la Corte Federal de Apelaciones (CAFC) (9 de Mayo de 2014)

-CAFC sostiene que utilizar los casos Sony y Sega para revisar el asunto de protección bajo derechos de autor de un determinado programa de computación constituye un error

-CAFC expresa que en ambos casos Sony y Sega se estableció que los programas de software tenían aspectos funcionales que no estaban protegidos por el derecho de autor, pero por ello no puede afirmarse que todo el programa de computación no sea protegible por derechos de autor.

-CAFC sostiene que el foco debería estar puesto por ejemplo en las especificaciones de la computadora sobre el cual el software se ejecutaría, o los requerimientos de compatibilidad de otros programas con lo cual el programa en desarrollo debería operar en forma conjunta

-Con lo cual teniendo en cuenta que la ley de derechos de autor de EE.UU se focaliza en las opciones disponibles que tenía el autor al momento de crear el programa de computación, la pregunta relevante en materia de compatibilidad sería si las opciones del creador fueron dictadas o no por la necesidad de asegurar que su programa funcione con los programas existentes de terceros.

-Es la interoperabilidad de Oracle la que importa y no la de Google, y el requisito de la compatibilidad debe ser analizado al momento del creador y no del que copia.

# Sentencia I de la Corte Federal de Apelaciones (CAFC) (9 de Mayo de 2014)

-CAFC expresa que la compatibilidad buscada por Google no era con la Plataforma Java SE o con la maquina virtual de Java, sino que Google quería capitalizar el hecho de que los desarrolladores ya estaban entrenados y con experiencia en utilizar los paquetes de las API de la Librería Estándar de Java.

-Google quería acelerar el proceso de desarrollo de su Plataforma Android aprovechando a Java con su base existente de desarrolladores.

-Google tuvo la libertad de desarrollar sus propias paquetes de API(s), y cabildear por su adopción, pero no lo hizo, y por contrario copio el código declarado por Oracle y la SSO de las API(s) de Java para capitalizar la comunidad de programadores preexistentes quienes ya conocían y estaban acostumbrados a usar los paquetes de Java.

-CAFC entendió que no había suficientes constancias en relación a los cuatro (4) factores del uso justo, y por ende mando a llevar a cabo un segundo juicio para que se determinase si el uso por parte de Google del código declarado y la SSO podría ser considerado como uso justo.

# Recurso de Apelación de Google (*Writ of Certiorari I*)

-Google interpuso recurso de apelación contra la decisión de la CAFC para revisión de la Corte Suprema de EEUU.

-El recurso no fue aceptado SCOTUS.

CAFC no aplica en forma correcta el derecho establecido por el Circuito Noveno.

-Google sostuvo que la decisión de CAFC es contraria a lo dispuesto en el artículo 102, inc. b)

-Para poder hacer uso de la funcionalidad que proporcionan los métodos de Java, los programadores sólo deben conocer los comandos abreviados: **java.package.Class.method(input)**.

-Cada comando abreviado deriva directamente del código declarado.

-Por tanto para que los comandos abreviados conocidos por los programadores funcionaran en la Plataforma Android, Google tuvo que copiar el código declarado usado por Oracle en los 37 paquetes de las API de Java, ya que cualquier cambio que Google hubiese realizado hubiera impedido que esos comandos abreviados funcionan correctamente

- Antes de la publicación de Android, existían millones de líneas de código escritas en Java las cuales necesariamente utilizaban el formato de comando **java.package.Class.method()** para invocar y así poder utilizar algunas de las funcionalidades incluidas dentro de los 37 paquetes de las API de Java en discusión.

# Recurso de Apelación de Google (*Writ of Certiorari I*)

-Por tanto para que parte de esas líneas de código escritas (conocimiento) por los programadores independientes en lenguaje Java durante años pudieran ser reutilizadas en la Plataforma Android , Google se encontró obligado a utilizar el mismo sistema de comandos de `java.package.Class.method()`, con los mismos nombres y la misma estructura y organización.

-Google sostiene que la interpretación del art 102 inciso b) por parte de la CAFC es absolutamente incorrecto y contrario al derecho de EE.UU y a la historia legislativa de los incisos a) y b) del articulo 102 del titulo 17 del USC.

-Google manifiesta que bajo el art. 102 inc. a) la autoría de una obra original es “generalmente” protegible, y bajo el 102 inc. b) , que va sobre lo específico se establece que en ningún caso la protección sobre obras originales se extiende a los sistemas, métodos de operación, etc., sin importar la forma en la cual estos [métodos, sistemas, etc] sean descriptos, explicados, ilustrados o incorporados a la obra.

-La exclusión legal es explicita y absoluta cuando la norma se refiere “sin importar la forma” en la cual [el sistema o método de operación] sea descripta, explicada, ilustrada o incorporada a la obra.

# Recurso de Apelación de Google (*Writ of Certiorari I*)

-Google sostiene que las declaraciones de interface constituyen interfaces de software, y por lo tanto no son protegibles por el derecho de autor de los EE.UU

-La necesidad y existencia de las interfaces de software son esenciales en un mundo interconectado.

-Congreso de los EEUU: posibilidad de realizar ingeniería inversa para identificar elementos de un programa de computación que son necesarios para lograr la interoperabilidad de otro programa creado en forma independiente con otros programas

-Directiva Europea 91/250 de mayo de 1992

-SAS Institute Inc. v. World Programming Ltd (2012)

# Juicio II.

*Limitación a los derechos exclusivos  
Defensa de “Fair Use” (uso justo o  
legítimo)*

8 de Junio de 2016 al 15 de Noviembre de  
2019

# Defensa de *Fair Use*

-El Jurado emitió veredicto unánime a favor de Google y conforme a ello la Corte de Distrito (CD) emitió fallo definitivo el 8 de junio de 2016 expresando que el uso realizado por Google se consideraba uso justo o legítimo, posteriormente Oracle en Febrero de 2017 apeló la sentencia afirmando que el uso de Google en ningún sentido podía constituir uso justo

-artículo 107: “Limitaciones de los derechos exclusivos. Uso Justo. “Sin perjuicio de lo establecido en los artículos 106 y 106A, el uso justo de una obra protegida por derecho de autor, incluido el uso por reproducción en copias o registros fonográficos o por cualquier otro medio especificado en esa sección, con fines tales como críticas, comentarios, informes de noticias, enseñanza (incluyendo copias múltiples para uso en el aula), beca o investigación, no constituye una infracción de los derechos de autor. Para determinar si el uso que se hace de una obra en un caso particular es un “uso justo”, los factores a ser considerados incluirán: (1) el propósito y el carácter del uso, incluyendo si dicho uso es de naturaleza comercial o para propósitos educativos sin fines de lucro; (2) la naturaleza de la obra protegida por derechos de autor; (3) la cantidad y la sustancialidad de la parte utilizada en relación a la obra protegida por derechos de autor en su conjunto; y (4) el efecto del uso sobre el mercado potencial o el valor de la obra protegida por derechos de autor. El hecho de que una obra sea inédita no impedirá en sí mismo un hallazgo de uso justo o legítimo si dicho hallazgo se realiza considerando todos los factores anteriores”

# Defensa de *Fair Use*

-La teoría de la defensa de uso justo comenzó como una creación de la doctrina judicial de los EE.UU y luego fue codificado en el art. 107 de la ley de derechos de autor de EE.UU.

-Funciona como una limitación de excepción a los derechos exclusivos del autor de la obra protegida, y permite el uso para fines como críticas, comentarios, informes de noticias, educación, investigación, entre otros.

-La enumeración del art.107 no es taxativa

-La norma requiere una determinación caso por caso sobre si en una determinada circunstancia existe o no un uso legítimo, y para ello se debe realizar una análisis de cuatro (4) factores.

Factor 1. el propósito y el carácter del uso, incluyendo si dicho uso es de naturaleza comercial o para propósitos educativos sin fines de lucro;

Factor 2. la naturaleza de la obra protegido por derechos de autor;

Factor 3: la cantidad y la sustancialidad de la parte utilizada en relación a la obra protegida por derechos de autor en su conjunto; y

Factor 4: el efecto del uso sobre el mercado potencial o el valor de la obra protegida por derechos de autor.

El análisis a realizarse por los jueces implicar realizar un balance de los cuatro (4) factores, en el cual los jueces deben considerar si el objetivo de las leyes de derechos de autor de EE.UU de promover el progreso de las ciencias y las artes están mejor representada (i) permitiéndose dicho uso o (ii) impidiéndolo.

# Sentencia II de la Corte Federal de Apelaciones (CAFC) (9 de Marzo 2018)

**Factor 1. el propósito y el carácter del uso, incluyendo si dicho uso es de naturaleza comercial o para propósitos educativos sin fines de lucro**

-propósitos educacionales no lucrativos o de interés público **vs.** naturaleza comercial

-carácter transformativo del uso **vs.** suplantación de la obra original: la Corte Suprema de EE.UU ha establecido que el propósito central del primer factor es determinar si la nueva obra es transformativa, y en su caso en qué grado y extensión.

**-Transformativo:** agrega algo nuevo, con propósitos y carácter distinto, que alteran el primer uso con una nueva expresión, significado o mensaje. La segunda obra debe alterar la obra original con una nueva expresión, significado o mensaje, y servir con un propósito diferente del tenido por la primera obra.

El uso de los paquetes de las API(s) de Java por parte de Google no fue transformativo: (i) no se encuentra listado dentro del artículo 107, y tampoco Google mencionó o sugirió otros, (ii) el propósito de los paquetes de las API en Android era el mismo que el propósito de los paquetes en la Plataforma Java SE, (iii) Google no había efectuado una alteración de la expresión del contenido, (iv) los teléfonos inteligentes no eran un nuevo contexto (Savaje, Nokia y Danger) (v) lo copiado por Google es cualitativamente significativo.

La copia idéntica, para una idéntica función y propósito, y un mero cambio en su formato (de PCs a tabletas y teléfonos móviles inteligentes) es insuficiente, y por tanto no ha existido transformación.

**-Factor 1 a favor de Oracle**

# Sentencia II de la Corte Federal de Apelaciones (CAFC) (9 de Marzo 2018)

## Factor 2: la naturaleza de la obra protegido por derechos de autor

-Este factor se relaciona con el reconocimiento que existe al respecto de que ciertas obras protegidas por el derecho de autor se encuentran más cerca que otras del núcleo de protección, por tanto, es más difícil establecer la existencia de uso justo cuando se trata de obras que se encuentran alejadas del núcleo de protección.

-Este factor se trata de si la obra es de carácter informativo o creativo.

-La CAFC en 2014 reconoció que las API y su SSO era creativa en forma suficiente para calificarla como protegida por derechos de autor, pero también dijo que los aspectos funcionales podían ser relevantes para la defensa de uso justo de Google.

-Oracle menciono que 3 (de 37) paquetes de Java eran necesarios para utilizar el lenguaje de programación Java, pero no demostró como se distinguía la funcionalidad y la creatividad de esas declaraciones (**java.lang, java.io, java.util**) al respecto del resto de los 34 paquetes de las Java API(s)

-Factor 2 a favor de Google.

# Sentencia II de la Corte Federal de Apelaciones (CAFC) (9 de Marzo 2018)

## **Factor 3- la cantidad y la sustancialidad de la parte utilizada en relación a la obra protegida por derechos de autor en su conjunto**

-Este factor focaliza en la cantidad y sustancialidad utilizada de la obra protegida (Java SE 5.0), y no de la obra en infracción (Android)

-El porcentaje del material copiado no es dispositivo cuando la porción copiada es cualitativamente significativa.

-Quedo establecido en el juicio II que Google copió 11.500 líneas de código declarado, pero sólo 170 líneas de código se establecieron como obligatorias para poder utilizar y programar en lenguaje Java, es decir que Google copio 11.330 más que las necesarias.

-Google argumenta que copió una pequeña parte de Java, unas 11.500 líneas de código declarado de un total de 2.86 millones de líneas de código de la Librería de Software Estándar de Java, ahora Google copió el total de los 37 paquetes y la SSO de las API(s) de Java.

-Google busco capitalizar que los desarrolladores de software ya estaban entrenados en programar Java utilizando los formatos de comandos de la Librería de Software de Java. Ahora, no existe ningún derecho inherente de copiar con el propósito de capitalizar la popularidad de una obra protegida.

-Asumiéndose de que lo copiado fuera una pequeña cantidad, no es razonable pensar que el material copiado es cualitativamente insignificante, sobre todo cuando el material copiado era importante para crear la plataforma Android.

-Factor 3: en el mejor de los casos es neutral o favorece a Oracle.

# Sentencia II de la Corte Federal de Apelaciones (CAFC) (9 de Marzo 2018)

## -Factor 4 el efecto del uso sobre el mercado potencial o el valor de la obra protegida por derechos de autor

-Este factor se centra en que la copia realizada por terceros no perjudique materialmente la comerciabilidad de la obra protegida.

-Se considera “el daño actual o potencial del mercado para la obra protegida”, pero también “los mercados para los potenciales usos derivados”, en donde se incluyen los que el creador de la obra protegida podría desarrollar o los que licenciaría para que terceros desarrollen.

-El autor o titular decide cuando, como y de que forma lanzar el material protegido a nuevos mercados sea por el mismo o por terceros autorizados.

-Android fue usado como un sustituto para Java SE, y ello tuvo un impacto directo en el mercado

-Google sostienen que Java SE y Android no compitieron en el mismo mercado porque Oracle no era un fabricante de dispositivos, y además no había construido aún su propia plataforma para teléfonos móviles inteligentes: CAFC rechazo estos argumentos: la definición de mercado potencial incluye en autorizar a otros para que desarrollen productos derivados.

# Sentencia II de la Corte Federal de Apelaciones (CAFC) (9 de Marzo 2018)

-No hay nada justo o legítimo en tomar una obra protegida, copiarla exacta y usarla para los mismos propósitos y función que la obra original en una plataforma en competencia

Del estudio individual de los cuatro factores, la CAFC concluyó que permitir que Google explotara comercialmente la obra protegida de Oracle no contribuía con el avance de los propósitos del sistema de derecho de autor de los EE.UU

- Factor 4 a favor de Oracle

Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021)

Voto de la Mayoría

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Mayoría (1)

## **Factor 2: la naturaleza de la obra protegido por derechos de autor**

- El código declarado (que es inseparable de los métodos de llamadas de los programadores) incorporan un diferente tipo de creatividad.
- Sun/Oracle en la creación de Java, trataron de encontrar nombres para su código declarado que provocara intuitivamente una manera fácil de ser recordado.
- La idea era atraer a los programadores para que pudieran aprender el sistema de Java, y así, lograr su desarrollo.
- El código declarado está orientado al usuario (en inglés “user facing”) y debe estar diseñado y organizado de forma tal que sea intuitivo y entendible para que los desarrolladores los puedan invocar
- Estas características significan, como la de la interface de usuario, que el código declarado difiere de alguna manera con otros programas de computadora.

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Mayoría (2)

A diferencia de otros programas, (i) su valor deriva del valor de aquellos que no poseen derechos de autor, es decir, de los programadores, que invierten su propio tiempo y esfuerzo en aprender el sistema de las API, (ii) en sus esfuerzos para alentar a los programadores para aprender y usar el sistema de manera tal que estos usarán programas relacionados implementados por Sun Microsystems, que Google no ha copiado.

-Aunque el derecho de autor protege varias formas de escritos, también se ha mencionado la necesidad de reconocer que existen ciertas obras protegidas que están más cerca que otras del núcleo de protección del derecho de autor.

-El código declarado, como protegible, se encuentra más lejos que otros programas de computación -como el código implementado- del núcleo de protección por los derechos de autor.

**Factor 2 a favor del uso justo (Google)**

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Mayoría (3)

## **Factor 1: el propósito y el carácter del uso, incluyendo si dicho uso es de naturaleza comercial o para propósitos educativos sin fines de lucro**

-Google copió porciones de las API de Java de Sun/Oracle en forma precisa, y lo hizo por las mismas razones que tuvo Sun/Oracle cuando las creó: permitir a los programadores invocar o llamar a los programas implementados para que cumplirían con determinadas tareas

-El uso de Google de las API de Java buscó crear un nuevo producto: busco expandir el uso y la utilidad de los teléfonos móviles inteligentes basados en Android.

-En el sentido de que Google usó partes de las API de Java para crear una nueva plataforma que permitiera ser rápidamente usada por los programadores, este uso es consistente con el progreso “creativo” que es el objetivo constitucional del derecho de autor en sí mismo.

-Google copió sólo las API de Java necesarias para incluir las tareas que serían útiles en los programas para teléfonos móviles inteligentes, y lo hizo solo en la medida necesaria para permitir a los programadores invocar aquellas tareas, sin descartar la porción del lenguaje de programación con la cual ya estaban familiarizados, evitándose de que tuvieran que aprender uno nuevo.

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Mayoría (4)

-El objetivo primario del sistema de derecho de autor no es compensar a los autores por su trabajo, sino “Promover el Progreso de la Ciencia y las Artes aplicadas”.

-Aunque el uso de Google sea comercial ello no es dispositivo del primer factor, y ello particularmente a la luz del rol inherentemente transformativo que jugó la reimplementación del nuevo sistema de Android

El VMA concluye que el propósito y carácter de la copia realizada por Google fue “transformativo”, y por lo tanto que el factor primero sopesa en favor del uso justo.

## **Factor 1 a favor del uso justo (Google)**

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021). Voto Mayoría (5)

## **Factor 3- la cantidad y la sustancialidad de la parte utilizada en relación a la obra protegida por derechos de autor en su conjunto**

-Si se considerase el código declarado en forma aislada la cantidad copiada por Google sería relevante.

-Google copió el código declarado de 37 paquetes de las API de Java en un total aproximado de 11.500 líneas de código.

-Estas líneas de código, representa todo el código declarado necesario para llamar o invocar los cientos de diferentes funciones o tareas.

-Si se mirase la totalidad de los paquetes de las API de Java que incluye el código declarado más el código implementado por Sun Microsystems que suman un total de 2.86 millones de líneas de código, la cantidad de líneas copiada por Google sería poco representativo esas 11.500 líneas un 0,4%.

-Lo copiado por Google sugiere que la manera de enfocar el asunto es mirando las líneas de código que Google no copió

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Mayoría (6)

-Google lo copió porque los programadores ya habían aprendido a cómo usar las API de Java, y sería muy difícil, e incluso tal vez prohibitivo, atraer a los programadores para que construyeran o desarrollaran en la plataforma Android sin ellas

-El objetivo de Google no era solamente hacer que el lenguaje de programación de Java sea el usado en la plataforma Android, sino además permitir a los desarrolladores hacer uso de su conocimiento y experiencia ya ganada utilizando las API de Java cuando estos escribieran nuevos programas para teléfonos móviles inteligentes en la plataforma Android.

-Es cierto que Google podría haber creado su propio código declarado, ahora no es menos cierto que podría pensarse que con ello Google no podría haber logrado su objetivo básico: el código declarado era el necesario para desbloquear las “energías creativas de los programadores”, y Google necesitaba de esas energías para crear su propio sistema innovador Android.

Factor 3 a favor del uso justo (Google)

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Mayoría (7)

## **Factor 4 el efecto del uso sobre el mercado potencial o el valor de la obra protegida por derechos de autor**

-En relación a la probabilidad de pérdida de ingresos Android no causó daño en el mercado actual o potencial para Java SE

-Sun Microsystems (ahora, Oracle) no habría sido capaz de entrar a estos mercado- actual y potenciales- con éxito, y ello con independencia de si Google hubiese copiado o no las API de Java

Sun Microsystems estaba muy mal posicionado para tener éxito en el mercado de los teléfonos móviles, ademas el mercado primario de Sun Microsystems era el de computadoras de escritorios y el de las computadoras portátiles o notebooks.

-Los dispositivos que utilizan la plataforma Android son distintos de aquellos que están licenciados con la tecnología de Sun/Oracle.

-la industria en general distingue entre teléfonos móviles inteligentes y “teléfonos móviles con funciones”.

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Mayoría (8)

-Los dispositivos que utilizan la tecnología creada por Sun/Oracle, surge que uno de esos dispositivos no poseía una pantalla táctil, mientras que otros no poseían un teclado QWERTY, y otros como Kindle, si bien utilizaba Java, sus modelos más avanzados como Kindle Fire fueron construidos con el sistema operativo de Android.

- En lugar de reutilizarse código de computadoras de escritorio y portátiles, la plataforma Android de Google fue parte de un mercado distinto y más avanzado que el mercado del software Java SE.

-La fuente rentable de ingresos de Android está relacionado con la inversión de “terceras partes” (los programadores) en los programas de Java de Sun/Oracle. En consecuencia, no tiene que ver con la inversión de Sun/Oracle en la creación de las API de Java

-No hay razón para pensar que el sistema de derechos de autor de EE.UU. busca proteger la inversión de terceras partes por el aprendizaje logrado de como operar una obra creada

-Entonces, permitir la prosecución de los derechos de autor de Oracle podría ir en contra del interés general.

-teniéndose en cuenta los costos y las dificultades de producir API alternativas, y de que estas posean un similar atractivo para los programadores, permitir la prosecución en este caso haría que el código declarado de las API de Java opere como un “cerrojo” limitando el futuro creativo de nuevos programas. Sólo Oracle tendría la llave.

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Mayoría (9)

- En este sentido, el “cerrojo” interferiría con los principios básicos del derecho de autor.
- El intento de monopolizar el mercado haciendo imposible para otros de competir va en contra de los propósitos de la ley de derechos de autor de promover las expresiones creativas
- el derecho de autor provee del incentivo económico para crear y diseminar las ideas, y la reimplementación de una interface de usuario, permite la creación de nuevo código fuente para lograr entrar al mercado de una manera más accesible.

Factor 4 a favor del uso Justo (Google)

Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021)

## Voto de la Minoría

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Minoría (1).

## **Factor 2: la naturaleza de la obra protegido por derechos de autor**

-Este factor requiere una evaluación respecto del nivel de “creatividad o funcionalidad” en la obra original.

-De modo general se suele decir que este factor favorece el uso justo (i) cuando la obra protegida posee carácter “informativo o funcional”, en lugar de creativo, (ii) y como el código de computadora es predominantemente funcional este factor también suele favorecer la copia cuando la obra original trata de código de computadora.

-Cuando El Congreso de los EE.UU. decidió acerca de la protección bajo derechos de autor y definió al código de computadora como lo hizo, rechazó cualquier distinción o categorización entre código declarado y código implementado.

-El código implementado ordena a una computadora una operación de forma directa, y el código declarado lo hace de manera indirecta.

-La definición legal incluyó al código fuente declarado como conjunto de declaraciones que indirectamente ejecutan funciones de cómputo mediante el código implementado.

-El valor del código implementado es directamente proporcional a como los programadores valoran el código declarado asociado

## **Factor 2 a favor de uso justo (Google)**

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Minoría (2).

## **Factor 4 el efecto del uso sobre el mercado potencial o el valor de la obra protegida por derechos de autor**

- El cuarto factor constituye el elemento individual más importante en el análisis de uso justo.
- la evidencia en el juicio que surge del daño actual y potencial a causa de lo copiado por Google, es abrumadora.
- Al copiar el código de Oracle y luego desarrollar y lanzar Android, Google arruinó el mercado potencial de Oracle en menos de dos años. Google eliminó las razones por las cuales los fabricantes podrían querer pagar por instalar Java en sus dispositivos.
- Mientras que los ingresos de Oracle se generan por cobrarles a los fabricantes de dispositivos por instalar Java en sus aparatos fabricados, Google obtiene sus ingresos principalmente a través de sus ventas de publicidad.
- La estrategia al lanzar Android fue para ofrecerlo sin costo a los fabricantes de dispositivos y así utilizar a Android como vehículo para recolectar datos de consumidores y así entregar publicidad basada en comportamiento.

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Minoría (3)

-Teniendo presente que había disponible un producto sin costo, “gratis”, que incluía el código de Oracle, los fabricantes de dispositivos no vieron muchas razones para seguir pagándole a Oracle por embeber Java SE en sus dispositivos

-Amazon pagaba a Oracle por una licencia por incluir Java SE en sus dispositivos Kindle. Luego de que Google liberaría Android al mercado, Amazon utilizó el argumento de que Android era gratuito para obtener un descuento de Oracle del 97, 5%.

-Samsung tenía un contrato con Oracle de USD 40 millones de dólares. Después de Android ese mismo contrato fue de USD 1 millón de dólares.

-Java Micro Edition es un conjunto de Java SE. Google copió código de ambas plataformas.

-Google interfirió con las oportunidades de Oracle de licenciar la plataforma Java para los desarrolladores de sistemas operativos para dispositivos móviles. Antes de que Google copiara el código de Oracle, en casi todos los teléfonos móviles del mercado se encontraba Java SE.

-Cuando se analiza el mercado potencial que los creadores de una obra original probablemente desarrollarían, no sólo se mira lo que ellos podrían desarrollar por sí mismos, sino también aquello que podría desarrollarse a través del modelo de licenciamiento a terceros.

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Minoría (4)

- Las acciones de Google causaron un efecto desastroso en el mercado potencial de Oracle.
  - Este caso sólo involucra las versiones de Android hasta noviembre de 2014, y desde esa fecha Google ha publicado seis versiones, y sólo el 7,7% de los dispositivos activos de Android contienen las versiones anteriores al 2014.
  - La preocupación del voto de la mayoría acerca del posible vendor lock-in de parte de Oracle tendría más sentido si estuviera discutiendo acerca de las versiones actuales en uso de Android o de las que estarían en uso. No hace demasiado sentido cuando se trata de versiones que están a punto de ser obsoletas.
  - El voto de la mayoría expresa preocupación en relación a que Oracle pueda abusar de sus derechos de autor (en las versiones obsoletas de Android) y realizar un intento de monopolizar el mercado. Ahora, fue Google quién recientemente fue condenado a una multa de USD 5 billones por cometer prácticas ilegales en relación con Android y violar leyes antimonopólicas.
- Entonces, es Google quién controla el sistema operativo para teléfonos móviles más usado en el mundo. Si el voto de la mayoría está preocupado en relación a las conductas monopólicas, debería considerar si Google es la gran amenaza.

## **Factor 4: en contra del uso justo (Oracle)**

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Minoría (5)

## **Factor 1. el propósito y el carácter del uso, incluyendo si dicho uso es de naturaleza comercial o para propósitos educativos sin fines de lucro**

-Este factor constituye el segundo factor más importante.

-Sólo en el año 2015, Google obtuvo una ganancia de USD 18 Billones (o USD 18.000 millones lo que es lo mismo) en relación a Android. Si este uso comercial no es decisivo por lo menos pesa muy fuerte en contra del uso justo.

-El voto de la mayoría intentar minimizar este abrumador uso comercial sosteniendo que el uso comercial no necesariamente juega en contra del uso justo. Es cierto, el uso comercial a veces se ve superado por un uso que es lo suficientemente transformativo. Ahora, no se puede ignorar el propósito de Google de suplantar la valiosa plataforma comercial de Oracle con la suya propia.

-No podría encontrarse uso justo si el uso del copista no es transformativo. Una obra es transformativa si agrega algo nuevo, con un propósito diferente, alterando la primera expresión, significado o mensaje.

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Minoría (6)

-Si una obra es transformativa o no es guiada por los ejemplos provistos en el artículo 107, Título 17 USC, cuando incluye como ejemplos, y de manera simplemente enunciativa, a las críticas, enseñanza, educación, becas, investigación, etc.

-Google, en su reutilización del código de Oracle no efectuó ninguno de ellos, como tampoco uso el código de Oracle para enseñar, o efectuar ingeniería inversa o para asegurar la compatibilidad entre sistemas. Por el contrario, Google utilizó el código declarado para los mismos exactos propósitos que Oracle.

Factor 1 a favor de Oracle

## **Factor 3- la cantidad y la sustancialidad de la parte utilizada en relación a la obra protegida por derechos de autor en su conjunto**

-A mayor cantidad en uso o copiada, menor la probabilidad de uso justo.

-Aún, si el copista toma una pequeña parte, copiando el “corazón” o “puntos centrales” de una obra, ello pesa en contra de un uso justo, a menos que no se haya tomado más de lo necesario por el copista para lograr su uso transformativo.

# Sentencia de la Corte Suprema de EE.UU. (5 de abril 2021) Voto Minoría (7)

-Google copio el corazón o la parte principal de la obra protegida. El código declarado es lo que atraída a los programadores a la plataforma Java y por ello Google estaba tan interesado en ello.

-El Voto Mayoria expresa que Google tomo no más que lo necesario para crear nuevos productos. Esta conclusión es errónea, porque el uso de Google no es transformativo.

-Aunque el uso de Google fuera transformativo, el Voto Mayoria concluye que Google sólo copió una pequeña porción de la obra original. Señala así el Voto Mayoria que las 11.500 líneas de código declarado copiado por Google, constituye solamente “una fracción” del código de la plataforma Java.

-El problema con esta visión es que el denominador correcto no es “todo el código” de la plataforma Java SE, sino “sólo” el código declarado.

## **-factor 3 a favor de Oracle**

-Tres de los cuatro factores pesan decididamente en contra de Google.

-La naturaleza de la obra protegida, el único factor que tal vez podría favorecer a Google, por sí sólo no puede lograr una determinación de uso justo, porque sostener ello sería contrariar los principios que el Congreso de los EE.UU. tomó en cuenta al determinar la protección bajo derechos de autor del código declarado

# Preguntas

Los alumnos pueden enviar sus preguntas a la siguiente dirección de correo electrónico:

[ggma@ggmalaw.com](mailto:ggma@ggmalaw.com)