

REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Oficio N° 790 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013186

Señor(a)
LLC MICROSOFT TECHNOLOGY LICENSING

TÍTULO DE LA SOLICITUD: **“OPTIMIZADOR DE CONSULTAS PARA UTILIZACIÓN DE CPU Y REFACTORIZACIÓN DE CÓDIGO”**

NÚMERO DE SOLICITUD INTERNACIONAL: PCT/US2017/035085

FECHA DE PRESENTACIÓN INTERNACIONAL: 31 de mayo de 2017.

PRIORIDAD(ES): 06/06/2016, US (ESTADOS UNIDOS DE AMÉRICA), 15/174,688

Como resultado del examen de patentabilidad realizado a la solicitud indicada en la referencia, atendiendo a lo consagrado en el artículo 45 de la Decisión 486 de 2000, se le comunica que esta Dirección ha encontrado que la materia objeto de solicitud no cumple con algunos requisitos establecidos en la Decisión, con fundamento en los siguientes aspectos:

1. Objeto de la invención

De acuerdo con lo establecido en el numeral 1.2.2.2 del Capítulo Primero del Título X de la Circular Única para efectos del examen de patentabilidad, cuando la solicitud contenga más de diez (10) reivindicaciones y no se haya pagado la tasa complementaria, sólo se estudiarán las cubiertas por la tasa pagada al momento de realizar el examen de patentabilidad consagrado en el artículo 45 de la Decisión 486. Por lo tanto, se estudiarán las reivindicaciones 1 a 15 del radicado N° NC2018/0013186 del 5 de diciembre de 2018.

El problema planteado en esta solicitud consiste en la necesidad de diseñar e implementar un método que permita aumentar la eficiencia de las consultas en códigos de programación.

Para resolver este problema técnico, la solicitud en estudio proporciona métodos, sistemas, aparatos y productos de programas informáticos para aumentar la eficiencia de las consultas en el código de programa. Una pluralidad de consultas se detecta en el código de programa. Se extiende una evaluación perezosa mediante la cual las consultas se evalúan en el código de programa. Las consultas se descomponen en una pluralidad de componentes de consulta. Un conjunto de reglas que incluye una pluralidad de reglas se aplica a los componentes de consulta para generar un conjunto de consultas funcionalmente equivalente a la pluralidad de consultas que evalúa de manera más eficiente con relación a la pluralidad de consultas.

2. No es una invención (Art. 15 Decisión 486 literal(e))

El objeto de la reivindicación 15 no se considera una invención, de acuerdo con el artículo 15, literal (e) de la Decisión 486: *“Artículo 15.- No se considerarán invenciones: (...) e) los programas de ordenadores o el soporte lógico, como tales; y, (...)”*, dicha reivindicación se refiere de manera explícita y literal a un software o programa informático en la cual menciona: *“programa informático”*, razón por la cual bajo los lineamientos de la Decisión 486 no se considera materia susceptible de protección.

3. De la solicitud de patente



Oficio N° 790 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013186

3.1. Reivindicaciones (Art. 30 Decisión 486)

- El preámbulo de las reivindicaciones 1 a 14 no permiten identificar con claridad el objeto a proteger y las características que lo definen porque mencionan: *“Un método en un dispositivo informático”* (Reiv. 1), *“El método de acuerdo con la reivindicación”* (Reivs. 2 a 10), *“Un dispositivo informático, que comprende”* (Reiv. 11) y *“El dispositivo informático de acuerdo con la reivindicación”* (Reiv. 12 a 14). Se recomienda presentar el preámbulo indicando cuál es el objeto de la invención, el cual suele coincidir con el título de la invención y las características técnicas conocidas en el estado de la técnica.
- Las reivindicaciones 1 a 15 mencionan las frases: *“extender la evaluación perezosa”* (Reivs. 1 y 11), *“aplicar un conjunto de reglas, que incluye una pluralidad de reglas, a las consultas detectadas”* (Reivs. 1 y 11), *“consultas funcionalmente equivalente”* (Reivs. 1 a 3, 6 a 9 y 11 a 13), *“refactorice automáticamente el código de programa”* (Reivs. 11 y 12), *“determinar un contexto lógico común entre múltiples componentes de consulta”* (Reiv. 5), *“aplicar una regla del conjunto de reglas”* (Reiv. 5), *“pluralidad de consultas”* (Reivs. 1 a 3, 6 a 8 y 11 a 13) y *“pluralidad de combinaciones”* (Reivs. 7 y 8), son frases de carácter general que sugieren de forma imprecisa que la protección se extendería a posibles variaciones o modificaciones. Se sugiere sustituirlos por un término preciso y presentar una mayor especificidad en cada una de las mismas.
- Las reivindicaciones 1, 6 y 11 no son claras porque los términos: *“más eficiente”* (Reivs. 1, 6 y 11) y *“mayor ganancia de eficiencia”* (Reivs. 7 y 8) son relativos por lo cual, no permiten distinguir la invención del estado de la técnica. Se sugiere sustituirlos por un término o rango concreto o presentar las evidencias correspondientes que demuestren dicho incremento en la eficiencia.
- La reivindicación 10 menciona: *“utilizar el aprendizaje automático para generar al menos una regla para agregarse al conjunto de reglas”* la cual indica el uso que puede darse al aprendizaje automático, el cual no es una característica técnica que defina a la invención. Se sugiere eliminarlo.
- La reivindicación 15 menciona: *“programa informático”*, el cual hace referencia explícita a un software. Se sugiere reemplazar dicha frase por: medio legible por computador para ejecutar una serie de pasos almacenados en este.

4. Determinación del Estado de la Técnica

La fecha para determinar el estado de la técnica es el 6 de junio de 2016 que corresponde a la fecha de presentación de la solicitud de prioridad.

Consultadas las fuentes de información con que se cuenta, se encontraron los siguientes documentos que anticipan el objeto de esta solicitud:

Ítem	Documento	Título	Fecha de Publicación
D1	US20040210882	“Program converting apparatus, method, and program”	21 de octubre de 2004
D2	US20120078878	“Optimized lazy query operators”	29 de marzo de 2012

Ref. Expediente N° NC2018/0013186

El documento D1¹ divulga métodos, aparatos y sistemas de conversión de programas que incluyen una unidad de análisis de código para realizar análisis léxicos y sintácticos de un código fuente de un programa de ejecución, una unidad de optimización para transformar este programa de ejecución y una unidad de generación de código de salida para convertir el programa de ejecución transformado en un código de lenguaje de máquina. Además, esta unidad de optimización detecta un procedimiento de llamada y un procedimiento llamado en una llamada a procedimiento de este programa de ejecución analizado por la unidad de análisis de código, protege una evaluación de un argumento descrito en el procedimiento llamado bajo una condición de evaluación predeterminada y transforma la ejecución programa para que la evaluación se realice al referirse a este argumento (Resumen).

El documento D2² divulga una implementación eficiente de operadores de consulta. Más específicamente, los operadores de consultas, como, entre otros, los que proporcionan la funcionalidad de agrupación, pueden implementarse para ejecutarse de manera perezosa o bajo demanda, en lugar de con entusiasmo como se hace convencionalmente. A modo de ejemplo y no de limitación, uno o más grupos pueden crearse y/o rellenarse perezosamente con uno o más elementos de una secuencia fuente en respuesta a una solicitud de un grupo o elemento de un grupo. Además, una implementación de operador diferido puede optimizarse en función del contexto que rodea una consulta. Por ejemplo, la creación y la población de grupos pueden restringirse, entre otras cosas (Pág. 1, Párr. 0005).

5. Examen de Patentabilidad (Art. 14 Decisión 486)

5.1. Novedad (Art 16 Decisión 486)

La reivindicación 1 consiste en: “Un método en un dispositivo informático que incluye al menos un procesador, que comprende: (...)” (Radicado N° NC2018/0013186 del 5 de diciembre de 2018).

Tabla 1. Comparación de las características técnicas esenciales, con las del estado de la técnica

Características esenciales	D1	D2
Un método en un dispositivo informático que incluye al menos un procesador, que comprende.	Además, otro aspecto de la presente invención que logra los aspectos anteriores es un método de conversión de programa, que controla una computadora y transforma un programa para ser procesado (Pág. 1, Párr. 0012, líneas 1 a 4).	El operador o método "Take" se muestra en (620) (Pág. 5, Párr. 0053, líneas 8 y 9).
Detectar una pluralidad de consultas en el código de programa.	Todavía otro aparato de conversión de programas de la presente invención comprende: una unidad de análisis de código; una unidad de optimización; y una unidad de generación de código. Este aparato se caracteriza porque la unidad de optimización incluye: primeros medios de transformación para detectar un procedimiento de llamada y un procedimiento llamado en una llamada de procedimiento	Descrito brevemente, la divulgación del tema generalmente se refiere a la implementación eficiente de operadores de consulta. Más específicamente, los operadores de consulta, como, entre otros, los que proporcionan la funcionalidad de agrupación, se pueden implementar para ejecutar de forma perezosa o bajo demanda, en lugar de con entusiasmo como se hace

¹<https://patentimages.storage.googleapis.com/84/1e/55/50dcaa8fc961a1/US20040210882A1.pdf>

²<https://patentimages.storage.googleapis.com/ae/6d/45/1fb8100029cc8a/US20120078878A1.pdf>

Oficio N° 790 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013186

	dentro del programa de ejecución (Pág. 1, Párr. 0011, líneas 1 a 7).	convencionalmente (Pág. 1, Párr. 0005, líneas 1 a 6).
Extender la evaluación perezosa mediante la cual las consultas se evalúan en el código de programa.	De la manera descrita hasta ahora, en esta realización, el compilador (100) puede insertar automáticamente la condición (condición de evaluación) derivada de la condición (condición de referencia) para referirse al argumento del destinatario de la llamada cuando el programa se compila. A diferencia del cambio en la evaluación perezosa, que se puede establecer en el Esquema, no se requiere un trabajo problemático de reescritura del código fuente. Además, la evaluación intensiva se utiliza para la evaluación de los argumentos en el caso de que la condición de evaluación sea Satisfecha, lo que permite mantener la ventaja en la reducción sustancial de la latencia (Pág. 5, Párr. 0060). Transformando el programa para que la evaluación se realice cuando haciendo referencia a este argumento y almacenando el programa transformado en un dispositivo de almacenamiento predeterminado (Pág. 2, Párr. 0012).	Los detalles a continuación generalmente están dirigidos a operadores de consulta diferidos y optimizaciones de los mismos (Pág. 1, Párr. 0021, líneas 1 a 2). Debe apreciarse que, por razones de brevedad y simplicidad, se han descrito aspectos de la divulgación con respecto al operador / método "GroupBy". Sin embargo, dichos aspectos no se limitan a los mismos y, de hecho, se extienden fácilmente a varios otros operadores/métodos, como "Seleccionar muchos" y "OrderBy", entre otros, a la luz de "Take". "TakeWhile". "TakeUntil" y "Skip", por ejemplo (Pág. 5, Párr. 0052).
Aplicar un conjunto de reglas, que incluye una pluralidad de reglas, a las consultas detectadas para generar un conjunto de consultas funcionalmente equivalente que evalúe de manera más eficiente con relación a las consultas detectadas.	En general, el llamador puede insertar un cálculo arbitrario entre la evaluación del argumento y la llamada al procedimiento utilizando los argumentos evaluados (Pág. 3, Párr. 0039, líneas 1 a 3). Además, en aras de la conveniencia de sincronizar el tamaño de las condiciones de evaluación a partir de la condición de referencia que es un código de la persona que llama, se requiere que se determine un conjunto de la persona que llama en un punto de llamada durante una compilación de la persona que llama. Esta condición se cumple en una llamada estática en la que el destinatario se determina de forma única, una llamada dinámica estáticamente vinculada por un código de prueba y una llamada dinámica como en una función virtual de C++, que no se acompaña de una sobrecarga durante la ejecución de un método (Pág. 4, Párr. 0041, líneas 1 a 9).	Además, como se apreciará, varias partes de los sistemas descritos anteriormente y los métodos a continuación pueden incluir o consistir en inteligencia artificial, aprendizaje automático o conocimiento o componentes basados en reglas, subcomponentes, procesos, medios, metodologías o mecanismos (p. Ej., máquinas de vectores de soporte, redes neuronales, sistemas expertos, redes de creencias bayesianas, lógica difusa, motores de fusión de datos, clasificadores ...) (Pág. 5, Párr. 0056, líneas 1 a 8).

De acuerdo con la tabla anterior, las características esenciales del método de la reivindicación 1 habían sido divulgadas previamente en los documentos D1 y D2, por lo tanto, la materia reivindicada no es nueva.

Oficio N° 790 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013186

Asimismo, el documento D1 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 3: Es otro aspecto de la presente invención proporcionar un compilador que optimice un programa en el programa eficiente para compilar el programa (Pág. 1, Párr. 0009).

Asimismo, el documento D2 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 3: Es otro aspecto de la presente invención proporcionar un compilador que optimice un programa en el programa eficiente para compilar el programa (Pág. 1, Párr. 0009).
- Reivindicación 5: La figura 10 es un diagrama de flujo de un método para optimizar la ejecución de operadores de consulta diferidos (1000). En el número de referencia (1010), se adquiere una política. Una política es como una regla en el sentido de que define una acción a tomar en un contexto dado. Por ejemplo, si un operador "GroupBy es seguido por un operador" Take ", entonces la implementación del operador" GroupBy puede verse restringida de modo que algunos grupos no se creen y/o llenen (Pág. 6, párr. 0061, líneas 1 a 8).

Las reivindicaciones dependientes 3 y 5 tampoco cumplen con el requisito de novedad.

La reivindicación 11 consiste en: *“Un dispositivo informático, que comprende: al menos un circuito procesador y (...)”* (Radicado N° NC2018/0013186 del 5 de diciembre de 2018).

Tabla 2. Comparación de las características técnicas esenciales, con las del estado de la técnica

Características esenciales	D1	D2
Un dispositivo informático, que comprende: al menos un circuito procesador y al menos una memoria que almacena el código de programa configurado para ejecutarse por al menos un circuito procesador, el código de programa está configurado para realizar operaciones que comprenden:	El método se caracteriza por comprender: un primer Paso para leer el programa a procesar desde un dispositivo de Almacenamiento predeterminado, y detectar un procedimiento de llamada y un procedimiento llamado en una llamada de procedimiento dentro del programa (Pág. 2, Párr. 0028, líneas 4 a 8).	Los aspectos también se pueden practicar en entornos informáticos distribuidos donde las tareas son realizadas por dispositivos de procesamiento remoto que están vinculados a través de una red de comunicaciones. Sin embargo, algunos, si no todos los aspectos de la materia reclamada se pueden practicar en computadoras independientes. En un entorno informático distribuido, los módulos de programa pueden estar ubicados en uno o ambos dispositivos de almacenamiento de memoria local y remota (Pág. 8, Párr. 0073, líneas 6 a 13).
Detectar una pluralidad de consultas en el código de programa.	Todavía otro aparato de conversión de programas de la presente invención comprende: una unidad de análisis de código; una unidad de optimización; y una unidad de generación de código. Este aparato se caracteriza porque la unidad de optimización incluye: primeros medios de transformación para	Descrito brevemente, la divulgación del tema generalmente se refiere a la implementación eficiente de operadores de consulta. Más específicamente, los operadores de consulta, como, entre otros, los que proporcionan la funcionalidad de agrupación, se pueden implementar para ejecutar

Oficio N° 790 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013186

	detectar un procedimiento de llamada y un procedimiento llamado en una llamada de procedimiento dentro del programa de ejecución (Pág. 1, Párr. 0011, líneas 1 a 7).	de forma perezosa o bajo demanda, en lugar de con entusiasmo como se hace convencionalmente (Pág. 1, Párr. 0005, líneas 1 a 6).
Extender la evaluación perezosa mediante la cual las consultas se evalúan en el código de programa.	De la manera descrita hasta ahora, en esta realización, el compilador (100) puede insertar automáticamente la condición (condición de evaluación) derivada de la condición (condición de referencia) para referirse al argumento del destinatario de la llamada cuando el programa se compila. A diferencia del cambio en la evaluación perezosa, que se puede establecer en el Esquema, no se requiere un trabajo problemático de reescritura del código fuente. Además, la evaluación intensiva se utiliza para la evaluación de los argumentos en el caso de que la condición de evaluación sea Satisfecha, lo que permite mantener la ventaja en la reducción sustancial de la latencia (Pág. 5, Párr. 0060). Transformando el programa para que la evaluación se realice cuando haciendo referencia a este argumento y almacenando el programa transformado en un dispositivo de almacenamiento predeterminado (Pág. 2, Párr. 0012).	Los detalles a continuación generalmente están dirigidos a operadores de consulta diferidos y optimizaciones de los mismos (Pág. 1, Párr. 0021, líneas 1 a 2). Debe apreciarse que, por razones de brevedad y simplicidad, se han descrito aspectos de la divulgación con respecto al operador / método "GroupBy". Sin embargo, dichos aspectos no se limitan a los mismos y, de hecho, se extienden fácilmente a varios otros operadores/métodos, como "Seleccionar muchos" y "OrderBy", entre otros, a la luz de "Take". "TakeWhile". "TakeUntil" y "Skip", por ejemplo (Pág. 5, Párr. 0052).
Aplicar un conjunto de reglas que incluye una pluralidad de reglas a las consultas detectadas para generar un conjunto de consultas funcionalmente equivalente que evalúe de manera más eficiente con relación a las consultas detectadas.	En general, el llamador puede insertar un cálculo arbitrario entre la evaluación del argumento y la llamada al procedimiento utilizando los argumentos evaluados (Pág. 3, Párr. 0039, líneas 1 a 3). Además, en aras de la conveniencia de sincronizar el tamaño de las condiciones de evaluación a partir de la condición de referencia que es un código de la persona que llama, se requiere que se determine un conjunto de la persona que llama en un punto de llamada durante una compilación de la persona que llama. Esta condición se cumple en una llamada estática en la que el destinatario se determina de forma única, una llamada dinámica estáticamente vinculada por un código de prueba y una llamada dinámica como en una función virtual de C ++, que no se acompaña de una sobrecarga durante la ejecución de un método (Pág. 4, Párr. 0041, líneas 1 a 9).	Además, como se apreciará, varias partes de los sistemas descritos anteriormente y los métodos a continuación pueden incluir o consistir en inteligencia artificial, aprendizaje automático o conocimiento o componentes basados en reglas, subcomponentes, procesos, medios, metodologías o mecanismos (p. Ej. , máquinas de vectores de soporte, redes neuronales, sistemas expertos, redes de creencias bayesianas, lógica difusa, motores de fusión de datos, clasificadores ...) (Pág. 5, Párr. 0056, líneas 1 a 8).

Oficio N° 790 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013186

De acuerdo con la tabla anterior, las características esenciales del dispositivo de la reivindicación 11 habían sido divulgadas previamente en los documentos D1 y D2, por lo tanto, la materia reivindicada no es nueva.

Asimismo, el documento D1 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 3: Es otro aspecto de la presente invención proporcionar un compilador que optimice un programa en el programa eficiente para compilar el programa (Pág. 1, Párr. 0009).

Asimismo, el documento D2 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 13: Es otro aspecto de la presente invención proporcionar un compilador que optimice un programa en el programa eficiente para compilar el programa (Pág. 1, Párr. 0009).

Las reivindicación dependientes 13 tampoco cumplen con el requisito de novedad.

6. Conclusión con respecto a novedad y/o nivel inventivo

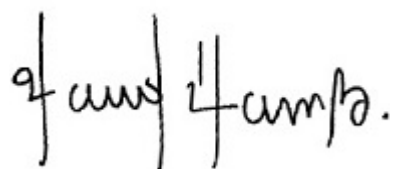
Los requisitos de Patentabilidad de la presente solicitud están afectados así:

Ítem	Documento	Reivindicaciones afectadas	Requisito que afecta
D1	US20040210882	1, 3, 11 y 13	Novedad
D2	US20120078878	1, 3, 5, 11 y 13	Novedad

En cumplimiento del artículo 45 de la Decisión 486 de 2000, se le indica que debe dar respuesta dentro del término de sesenta (60) días contados a partir de la fecha de notificación de la presente comunicación. En caso de no dar respuesta al presente requerimiento, o si a pesar de la respuesta subsistieran los impedimentos para la concesión, se denegará la patente.

Si como respuesta a este requerimiento, el solicitante modifica: i) el capítulo reivindicatorio, ii) la descripción de la invención a proteger, iii) o presenta un nuevo capítulo reivindicatorio deberá observar las instrucciones contenidas en el numeral 1.2.2.1.1., del Capítulo Primero del Título X de la Circular Única y la Dirección de Nuevas Creaciones seguirá el procedimiento descrito en el numeral 1.2.2.5.1.

Notifíquese el presente oficio conforme a lo dispuesto en el numeral 6.2 del Capítulo Sexto del Título I de la Circular Única, informándoles que contra el mismo no procede recurso alguno en actuación administrativa.



MARÍA JOSÉ LAMUS BECERRA
DIRECTORA DE NUEVAS CREACIONES



Oficio N° 790 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013186

INFORME DE BÚSQUEDA
EXAMEN DE PATENTABILIDAD

Solicitud No.		Examen de patentabilidad							
NC2018/0013186		1°	X	2°		3°		Revoca	
Solicitud Internacional No.		Fecha solicitud internacional: (DD/MM/AÑO)							
PCT/US2017/035085		31 de mayo de 2017							
Fecha de determinación estado de la técnica (DD/MM/AÑO)				Prioridad		Presentación			
6/6/2016				X					
Solicitante									
LLC MICROSOFT TECHNOLOGY LICENSING									
Reivindicaciones que no son objeto de búsqueda por:									
Art. 14	-	Art. 15	15	Art. 20	-	Art. 25 (Falta de unidad de invención)		-	

CRITERIOS DE BÚSQUEDA	
Reivindicaciones objeto de búsqueda	1 a 14
Palabras clave utilizadas	Lazy, optimization, evaluation, code, program, cpu, utilization, optimizer, refactoring.
Clasificaciones utilizadas para la búsqueda	CIP: G06F 9/44

DOCUMENTOS CONSIDERADOS RELEVANTES (Documentos patente, Literatura no patente (LNP): Artículos, catálogos, etc.)					
Informes de búsqueda / Bases de datos consultadas	N° de solicitud de patente / N° de patente / Autor LNP(¹)	Fecha de Publicación /Presentación (DD/MM/AAAA)	Reivindicaciones /Secuencias afectadas	Citas relevantes del estado de la técnica	Categoría del documento citado(²)
SIPI	04088586	8/9/2004	-	-	A
ISA	US2004/210882	21/10/2004	1, 3, 11 y 13	1 Pág. 1, Párr. 0012, líneas 1 a 4; Pág. 1, Párr. 0011, líneas 1 a 7; Pág. 2, Párr. 0012; Pág. 3, Párr. 0039, líneas 1 a 3; Pág. 5, Párr. 0060. 3 Pág. 1, Párr. 0009. 11 Pág. 2, Párr. 0028, líneas 4 a 8; Pág. 1, Párr. 0011, líneas 1 a 7; Pág. 2, Párr. 0012; Pág. 3, Párr. 0039, líneas 1 a 3; Pág. 5, Párr. 0060. 13 Pág. 1, Párr. 0009.	X
Patbase	US2007038618	15/02/2007	-	-	A
Orbit	EP2510464	17/10/2012	-	-	A
Googlepatent	US20120078878	29/03/2012	1, 3, 5, 11 y 13	1 Pág. 5, Párr. 0053, líneas 8 y 9; Pág. 1, Párr. 0005, líneas 1 a 6; Pág. 1, Párr. 0021, líneas 1 a 2; Pág. 5, Párr. 0052; Pág. 5, Párr. 0056, líneas 1 a 8.	X

Oficio N° 790 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013186

				3 Pág. 1, Párr. 0009. 5 Pág. 6, párr. 0061, líneas 1 a 8 11 Pág. 8, Párr. 0073, líneas 6 a 13; Pág. 1, Párr. 0005, líneas 1 a 6; Pág. 1, Párr. 0021, líneas 1 a 2; Pág. 5, Párr. 0052; Pág. 5, Párr. 0056, líneas 1 a 8. 13 Pág. 1, Párr. 0009.	
(1) Cita completa literatura no patente (LNP)					
Apellido, A. A. (Fecha). Título del artículo. <i>Nombre de la revista. Volumen</i> (Número), pp-pp.					
O					
Apellido, A. A. (Fecha). Título del artículo. Nombre de la revista. Recuperado de http://xxxxxxx .					
(2) Categorías de documentos citados					
“A” Documento que define el estado general de la técnica no considerado como particularmente relevante. “E” Solicitud de patente o patente anterior pero publicada en la fecha de presentación internacional o en fecha posterior. “L” Documento citado para llamar la atención sobre por ejemplo, dudas acerca de una reivindicación de prioridad o para determinar la fecha de publicación de otra cita, siempre van acompañados de una explicación, por la cual se citan. “O” Documento del tipo actas de conferencia, este tipo de documentos siempre estará acompañado por otra letra que indique la pertinencia del documento, por ejemplo, O,X, O,Y o bien O,A. “P” Documento publicado antes de la fecha de presentación internacional pero con posterioridad a la fecha de prioridad reivindicada. “T” Documento publicado en fecha posterior a la de presentación o la de prioridad de la solicitud internacional y no entra en conflicto con dicha solicitud, pero puede ser útil para la mejor comprensión del principio o teoría en que se basa la invención, o para demostrar que el razonamiento o los hechos en los que se basa dicha invención son incorrectos. “X” Documento particularmente relevante; la invención reivindicada no puede considerarse nueva o que implique una actividad inventiva por referencia al documento aisladamente considerado. “Y” Documento particularmente relevante; la invención reivindicada no puede considerarse que implique una actividad inventiva cuando el documento se asocia a otro u otros documentos de la misma naturaleza, cuya combinación resulta evidente para un experto en la materia.					
Fecha del reporte de búsqueda: 15/11/2019					
Examinador: FREDDY ALEXANDER VELANDIA VIVAS					