

REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Resolución N° 74051

Ref. Expediente N° NC2019/0011810

Por la cual se deniega una Patente de Invención

EL SUPERINTENDENTE DE INDUSTRIA Y COMERCIO (E)

en ejercicio de sus facultades legales, en especial de las conferidas en el numeral 24 del artículo 3° del Decreto 4886 de 2011, y

CONSIDERANDO:

Que mediante escrito radicado en esta Superintendencia el 24 de octubre de 2019 con el N° NC2019/0011810, por MICROSOFT TECHNOLOGY LICENSING LLC, entró a fase nacional la solicitud de patente de invención titulada “GUÍA DE TOMA DE DECISIONES MEDIANTE APRENDIZAJE AUTOMÁTICO PARA ALERTAS ORIGINARIAS DE SISTEMAS DE MONITORIZACIÓN”, que corresponde a la solicitud internacional PCT/US2018/026380 del 6 de abril de 2018.

Que la solicitud fue publicada en la Gaceta de la Propiedad Industrial N° 883 el 17 de enero de 2020, sin que se hubieran presentado oposiciones por parte de terceros.

Que realizado el examen de fondo mediante Oficio No. 8374 del 3 de junio de 2022, se requirió al solicitante en los términos del artículo 45 de la Decisión 486 de la Comisión de la Comunidad Andina para que presentara respuesta a las observaciones de carácter técnico, relacionadas con la patentabilidad o cumplimiento de los requisitos establecidos por esta Decisión para la concesión de la patente.

Que el solicitante mediante escrito radicado bajo el N° NC2019/0011810 el 30 de agosto de 2022, respondió oportunamente el requerimiento formulado sin presentar nuevo capítulo reivindicatorio. Por lo tanto, se tendrá en cuenta el capítulo reivindicatorio presentado con el radicado N° NC2019/0011810 el 25 de abril de 2022.

Que en virtud de lo dispuesto en el artículo 14 de la Decisión 486 expedida por la Comisión de la Comunidad Andina “*Los países miembros otorgarán patentes para las invenciones, sean de producto o de procedimiento, en todos los campos de la tecnología, siempre que sean nuevas, tengan nivel inventivo y sean susceptibles de aplicación industrial.*”

Que en el presente caso las reivindicaciones 1 a 20 no cumplen los requisitos indicados. Lo anterior, de acuerdo con las siguientes consideraciones:

Objeto de la invención

El problema técnico planteado consiste en la necesidad de suministrar un sistema y un método para determinar si un problema o falla detectada en un sistema de software es el resultado de un error inducido por la programación o debido a alguna otra causa.

Para resolver este problema técnico, la solicitud proporciona un método para determinar automáticamente si un problema detectado en un sistema de cómputo es un error. Este método incluye recibir información sobre un problema detectado y almacenar la información sobre el problema detectado en un dispositivo de almacenamiento. Se genera



Resolución N° 74051

Ref. Expediente N° NC2019/0011810

y analiza un vector de características para el problema detectado con el fin de determinar la probabilidad de que el problema detectado sea un error.

Determinación del Estado de la Técnica

La fecha para determinar el estado de la técnica es el 24 de abril de 2017, 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	US 2015/347212	"Error Classification In A Computing System"	03 de diciembre de 2015
D2	US 2015/019564	"Method and system for reducing instability when upgrading software"	15 de enero de 2015
D3	US 2014/053135	"Predicting Software Build Errors"	20 de febrero de 2014

D1¹ divulga un enfoque para determinar una clasificación de un error en un sistema informático (Pág. 1, Resumen).

D2² divulga un sistema y un método para calificar errores de software utilizando un sistema de seguimiento y puntuación de errores (Pág. 1, Resumen).

D3³ divulga sistemas y métodos para predecir un error de compilación de software (...) (Pág. 1, Resumen).

Examen de Patentabilidad

De acuerdo con lo establecido en el artículo 14 de la Decisión 486 de 2000 de la Comisión de la Comunidad Andina, en el cual se establecen los requisitos de patentabilidad objeto de la solicitud en estudio, esta Oficina presenta a continuación el análisis de los requisitos de patentabilidad.

Nivel Inventivo

Según lo dispuesto en el artículo 18 de la Decisión 486: *"Se considerará que una invención tiene nivel inventivo, si para una persona del oficio normalmente versada en la materia técnica correspondiente, esa invención no hubiese resultado obvia ni se hubiese derivado de manera evidente del estado de la técnica"*.

En el presente caso, el método reclamado en la solicitud no tiene nivel inventivo, teniendo en cuenta que la reivindicación 1 está referida a: *"Un método para determinar*

¹https://worldwide.espacenet.com/publicationDetails/biblio?CC=US&NR=2015347212A1&KC=A1&FT=D&ND=3&date=20151203&DB=&locale=en_EP
²https://worldwide.espacenet.com/publicationDetails/biblio?CC=US&NR=2015019564A1&KC=A1&FT=D&ND=3&date=20150115&DB=&locale=en_EP
³https://worldwide.espacenet.com/publicationDetails/biblio?CC=US&NR=2014053135A1&KC=A1&FT=D&ND=3&date=20140220&DB=&locale=en_EP

Resolución N° 74051

Ref. Expediente N° NC2019/0011810

automáticamente si un problema detectado en un sistema de cómputo es un error, que comprende (...)” (Radicado N° NC2019/0011810 el 25 de abril de 2022).

La siguiente tabla presenta la comparación entre las características técnicas esenciales de la solicitud estudiada, mencionadas en la reivindicación 1, y las reveladas en los documentos D1 y D3 que representan el estado de la técnica:

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

CARACTERÍSTICAS ESENCIALES	D3	D1
Un método para determinar automáticamente si un problema detectado en un sistema de cómputo es un error, que comprende:	(...) un método para predecir errores de construcción de software (Pág. 1, Párr. 003).	(...) un método, un producto de programa informático y un sistema informático para determinar una clasificación de un error en un sistema informático (Pág. 1, Resumen).
Recibir (210) información sobre un problema detectado.	El módulo de interrupción de compilación (122) puede enviar los cambios que causan errores de compilación de software al módulo de actualización (124) (Pág. 3, Párr. 025).	El programa de informes (134) recibe una notificación de error durante una ejecución de prueba (Pág. 3, Párr. 025).
Almacenar la información sobre el problema detectado en un dispositivo de almacenamiento.	El módulo de actualización (124) puede almacenar información histórica para cambios de código de software y errores de construcción correspondientes (Pág. 3, Párr. 025).	Cada instancia de la base de datos (122) almacena archivos de registro generados por una aplicación de software u otro componente dentro de cada dispositivo informático cliente (120) respectivo (Pág. 1, Párr. 012).
Generar (220) un vector de características para el problema detectado.	(...) las probabilidades de errores de creación de software se pueden calcular mediante un tipo de regresión o utilizando modelos de aprendizaje automático que incluyen, entre otros, máquinas de vectores de soporte (...) (Pág. 4, Párr. 040). Un vector de características hace parte del conocimiento del técnico medianamente versado en la materia en una máquina de vectores de soporte (SVM) ⁴ , puesto que, para el entrenamiento y clasificación de la SVM, es necesario un vector de características que define el conjunto de propiedades por las cuales serán diferenciados los datos.	(...) las entradas de registro se categorizan utilizando métodos conocidos en la técnica, por ejemplo, algoritmos de aprendizaje automático tales como aprendizaje automático supervisado por texto que incluye, por ejemplo, máquinas de vectores de soporte ("SVM") (Pág. 2, Párr. 020). Un vector de características hace parte del conocimiento del técnico medianamente versado en la materia en una máquina de vectores de soporte (SVM) ⁴ , puesto que, para el entrenamiento y clasificación de la SVM, es necesario un vector de características que define el conjunto de propiedades por las cuales serán diferenciados los datos.
Analizar (230) el vector de características para determinar una probabilidad de que el problema detectado sea un error	(...) el módulo de análisis de predicción (120) informa las probabilidades de que cada cambio en el código de software	No se menciona

⁴ <https://www.ugpti.org/smartse/resources/downloads/support-vector-machines.pdf>

Resolución N° 74051

Ref. Expediente N° NC2019/0011810

provocado por una alteración de al menos una porción del código fuente utilizado por el sistema de cómputo, en donde el análisis se basa, al menos en parte, en el vector de características.	pueda causar un error de construcción de software (Pág. 4, Párr. 036). Los ejemplos adicionales de características de la lista de cambios pueden incluir cualquier actividad realizada por otros desarrolladores en archivos de código fuente o líneas de archivos de código fuente que se hayan modificado (...). (Pág. 3, Párr. 031). (...) las probabilidades de errores de creación de software se pueden calcular mediante un tipo de regresión o utilizando modelos de aprendizaje automático que incluyen, entre otros, máquinas de vectores de soporte (...) (Pág. 4, Párr. 040).	
Cuando la probabilidad está por encima de un umbral, informar el problema detectado a un dispositivo cliente.	No se menciona	Si la puntuación de confianza alcanza o supera el valor umbral establecido, por ejemplo, 75% o más (bloque de decisión (312), rama "sí"), los resultados se informarán a un usuario (...) (Pág. 3, Párr. 030).
Recibir una entrada correspondiente al problema detectado, la entrada indicando si el problema detectado es un error.	El componente de informes (408) también puede enviar las listas de cambios y los errores de compilación del software a un componente de control de versiones (410) (Pág. 5, Párr. 050).	El programa de informes (134) recibe una notificación de error durante una ejecución de prueba (Pág. 3, Párr. 025).
Utilizar la entrada para actualizar la información sobre el problema detectado en el dispositivo de almacenamiento.	El módulo de actualización (122) puede entonces actualizar los datos históricos para reflejar si la lista de cambios provocó un error de construcción de software (Pág. 5, Párr. 044).	No se menciona
Proporcionar la información actualizada a un sistema que identificó el problema detectado para entrenar automáticamente el sistema.	El módulo predictor de actualización (506) puede entrenar un modelo de predicción basado en datos históricos (Pág. 6, Párr. 054).	No se menciona

D3 es considerado como el estado de la técnica más cercano a la invención definida en la reivindicación 1, ya que divulga un enfoque para determinar una clasificación de un error en un sistema informático (Pág. 1, Resumen).

La diferencia entre la invención definida entre la reivindicación 1 y el método divulgado en D3, consiste en que cuando la probabilidad está por encima de un umbral, informar el problema detectado a un dispositivo cliente.

El efecto que se logra al incluir específicamente que cuando la probabilidad está por encima de un umbral, informar el problema detectado a un dispositivo cliente es indicar si

Resolución N° 74051

Ref. Expediente N° NC2019/0011810

un problema es un error introducido por un desarrollador o algún otro problema intermitente.

Por lo tanto, el problema técnico objetivo que pretende resolver esta invención se puede formular así: ¿cómo modificar el método divulgado en D3 para indicar si un problema es un error introducido por un desarrollador o algún otro problema intermitente?

Sin embargo, el incluir que cuando la probabilidad está por encima de un umbral, informar el problema detectado a un dispositivo cliente para indicar si un problema es un error introducido por un desarrollador o algún otro problema intermitente, ya está divulgado en D1 que se refiere a que, si la puntuación de confianza alcanza o supera el valor umbral establecido, por ejemplo, 75% o más (bloque de decisión (312), rama "sí"), los resultados se informarán a un usuario (...) (Pág. 3, Párr. 030).

En consecuencia, la persona normalmente versada en la materia estaría motivada a incluir que cuando la probabilidad está por encima de un umbral, informar el problema detectado a un dispositivo cliente del D1 en el método de D3, para así llegar al objeto de la reivindicación 1. Por lo cual se considera obvia.

Así mismo, D1 divulga las características de las reivindicaciones dependientes de la siguiente manera:

- Reivindicaciones 6 y 7: El programa de entrenamiento (132) luego normaliza cada entrada en el archivo de registro y categoriza las entradas para crear identificadores (Pág. 2, Párr. 014). (...) las entradas de registro se categorizan utilizando métodos conocidos en la técnica, por ejemplo, algoritmos de aprendizaje automático tales como aprendizaje automático supervisado por texto que incluye, por ejemplo, máquinas de vectores de soporte ("SVM") (Pág. 2, Párr. 020).

Las reivindicaciones dependientes 6 y 7 no mencionan alguna característica que, junto con las características de la reivindicación independiente, aporte nivel inventivo al método de la reivindicación 1, por lo cual se considera que carecen de nivel inventivo.

En el presente caso, el sistema reclamado en la solicitud no tiene nivel inventivo, teniendo en cuenta que la reivindicación 8 está referida a: *“Un sistema para determinar automáticamente si un problema detectado en un sistema de cómputo es un error (...)”* (Radicado N° NC2019/0011810 el 25 de abril de 2022).

La siguiente tabla presenta la comparación entre las características técnicas esenciales de la solicitud estudiada, mencionadas en la reivindicación 8, y las reveladas en los documentos D2 y D3 que representan el estado de la técnica:

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

CARACTERÍSTICAS ESENCIALES	D3	D2
Un sistema para determinar automáticamente si un problema detectado en un sistema de cómputo es un error, el sistema comprende:	(...) un sistema para predecir errores de compilación de software (Pág. 1, Párr. 004).	(..) un sistema y un método para calificar errores de software utilizando un sistema de seguimiento y puntuación de errores (Pág. 1, Resumen).
Al menos una unidad de procesamiento (510; 685).	El sistema informático (100) puede incluir un procesador (102) que	(...) una memoria comunicativamente acoplada

Resolución N° 74051

Ref. Expediente N° NC2019/0011810

Una memoria (515; 645) que almacena instrucciones ejecutables por computadora que, cuando se ejecutan por el al menos una unidad de procesamiento (510; 685), provocan que el sistema realice un método para detectar ocurrencias de problemas en un sistema de cómputo, el método comprende.	está adaptado para ejecutar instrucciones almacenadas, así como un dispositivo de memoria (104) que almacena instrucciones que son ejecutables por el procesador (102) (Pág. 2, Párr. 022).	y legible por el uno o más procesadores y que tiene almacenadas en él una serie de instrucciones que, cuando son ejecutadas por el uno o más procesadores, provocan el uno o más procesadores para priorizar un error de software (Pág. 1, Párr. 005).
Recibir (210) información sobre un problema detectado.	El módulo de interrupción de compilación (122) puede enviar los cambios que causan errores de compilación de software al módulo de actualización (124) (Pág. 3, Párr. 025).	El motor de puntuación (113) puede recibir y recopilar datos de las fuentes de datos internas y externas relacionadas con el error (Pág. 3, Párr. 026).
Comparar una propiedad distintiva del problema detectado con una o más propiedades distintivas de problemas detectados previamente para determinar si el problema detectado es nuevo.	(...) el módulo de análisis de predicción (120) puede detectar los cambios en el código de software comparando el código de software con una versión anterior del código de software (Pág. 3, Párr. 028). (...) el módulo de análisis de predicción (120) informa las probabilidades de que cada cambio en el código de software pueda causar un error de construcción de software (Pág. 4, Párr. 036).	No se menciona
Cuando se determina que el problema detectado es nuevo: almacenar la información sobre el problema detectado en un dispositivo de almacenamiento.	El módulo de actualización (124) puede almacenar información histórica para cambios de código de software y errores de construcción correspondientes (Pág. 3, Párr. 025).	Los puntajes de los parámetros recibidos o calculados relacionados con cada error pueden almacenarse en una memoria (...) (Pág. 4, Párr. 040).
Generar (220) un vector de características para el problema detectado.	(...) las probabilidades de errores de creación de software se pueden calcular mediante un tipo de regresión o utilizando modelos de aprendizaje automático que incluyen, entre otros, máquinas de vectores de soporte (...) (Pág. 4, Párr. 040). Un vector de características hace parte del conocimiento del técnico medianamente versado en la materia en una máquina de vectores de soporte (SVM) ⁴ , puesto que, para el entrenamiento y clasificación de la SVM, es necesario un vector de características que define el conjunto de propiedades por las cuales serán diferenciados los datos.	No se menciona
Analizar (230) el vector de características para determinar una probabilidad de que el	(...) el módulo de análisis de predicción (120) informa las probabilidades de que cada	No se menciona



Resolución N° 74051

Ref. Expediente N° NC2019/0011810

problema detectado sea un error provocado por una alteración de al menos una porción del código fuente utilizado por el sistema de cómputo.	cambio en el código de software pueda causar un error de construcción de software (Pág. 4, Párr. 036). Los ejemplos adicionales de características de la lista de cambios pueden incluir cualquier actividad realizada por otros desarrolladores en archivos de código fuente o líneas de archivos de código fuente que se hayan modificado (...). (Pág. 3, Párr. 031) (...) las probabilidades de errores de creación de software se pueden calcular mediante un tipo de regresión o utilizando modelos de aprendizaje automático que incluyen, entre otros, máquinas de vectores de soporte (...) (Pág. 4, Párr. 040).	
Proporcionar una lista ordenada de problemas detectados a un dispositivo de cliente.	No se menciona	(...) los errores se pueden ordenar en función de su puntuación de errores y se puede asignar a los 20 errores principales una prioridad alta (Pág. 6, Párr. 059).
Recibir una entrada correspondiente a uno o más de los problemas detectados en la lista ordenada de problemas detectados, la entrada indicando si el problema detectado es un error, o un falso positivo.	El componente de informes (408) también puede enviar las listas de cambios y los errores de compilación del software a un componente de control de versiones (410) (Pág. 5, Párr. 050). No se menciona la lista ordenada	(...) recibir un informe de error de software (Pág. 1, Párr. 005). (...) los errores se pueden ordenar en función de su puntuación de errores y se puede asignar a los 20 errores principales una prioridad alta (Pág. 6, Párr. 059).
Utilizar la entrada para actualizar la información sobre el problema detectado.	El módulo de actualización (122) puede entonces actualizar los datos históricos para reflejar si la lista de cambios provocó un error de construcción de software (Pág. 5, Párr. 044).	No se menciona

D3 es considerado como el estado de la técnica más cercano a la invención definida en la reivindicación 8, ya que divulga un enfoque para determinar una clasificación de un error en un sistema informático (Pág. 1, Resumen).

La diferencia entre la invención definida entre la reivindicación 8 y el sistema divulgado en D3, consiste en proporcionar una lista ordenada de problemas detectados a un dispositivo de cliente.

El efecto que se logra al incluir específicamente el proporcionar una lista ordenada de problemas detectados a un dispositivo de cliente es priorizar el problema detectado con respecto a uno o más problemas detectados adicionales.

Resolución N° 74051

Ref. Expediente N° NC2019/0011810

Por lo tanto, el problema técnico objetivo que pretende resolver esta invención se puede formular así: ¿cómo modificar el sistema divulgado en D3 para priorizar el problema detectado con respecto a uno o más problemas detectados adicionales?

Sin embargo, el incluir proporcionar una lista ordenada de problemas detectados a un dispositivo de cliente para priorizar el problema detectado con respecto a uno o más problemas detectados adicionales, ya está divulgado en D2 que se refiere a que los errores se pueden ordenar en función de su puntuación de errores y se puede asignar a los 20 errores principales una prioridad alta (Pág. 6, Párr. 059).

En consecuencia, la persona normalmente versada en la materia estaría motivada a incluir el proporcionar una lista ordenada de problemas detectados a un dispositivo de cliente del D2 en el sistema de D3, para así llegar al objeto de la reivindicación 8. Por lo cual se considera obvia.

Así mismo, D3 divulga las características de las reivindicaciones dependientes de la siguiente manera:

- Reivindicación 9: (...) las probabilidades de errores de creación de software se pueden calcular mediante un tipo de regresión o utilizando modelos de aprendizaje automático que incluyen, entre otros, máquinas de vectores de soporte (...) (Pág. 4, Párr. 040). Un vector de características hace parte del conocimiento del técnico medianamente versado en la materia en una máquina de vectores de soporte (SVM)⁴, puesto que, para el entrenamiento y clasificación de la SVM, es necesario un vector de características que define el conjunto de propiedades por las cuales serán diferenciados los datos.

- Reivindicación 15: (...) el componente de activación (308) también puede enviar la lista de cambios a un módulo de actualización (312) si el componente de construcción (310) no devuelve un error de construcción de software. El módulo de actualización (122) puede almacenar datos históricos con respecto a las listas de cambios y el éxito o fracaso de las construcciones de software basadas en las listas de cambios (Pág. 5, Párr. 044).

De igual manera, D2 divulga las características de las reivindicaciones dependientes así:

- Reivindicaciones 10 y 11: El parche puede incluir información sobre los errores que corrige el parche (Pág. 7, Párr. 066).

- Reivindicación 12: Los parches de software relacionados con errores de alta prioridad clasificados y priorizados según las preferencias del cliente pueden agruparse en una o más actualizaciones periódicas de software (Pág. 7, Párr. 065). El parche puede incluir información sobre los errores que corrige el parche (Pág. 7, Párr. 066).

- Reivindicación 14: (...) los errores se pueden ordenar en función de su puntuación de errores y se puede asignar a los 20 errores principales una prioridad alta (Pág. 6, Párr. 059).

Las reivindicaciones dependientes 9 a 12, 14 y 15 no mencionan alguna característica que, junto con las características de la reivindicación independiente, aporte nivel inventivo al sistema de la reivindicación 8, por lo cual se considera que carecen de nivel inventivo.

En el presente caso, el sistema reclamado en la solicitud no tiene nivel inventivo, teniendo en cuenta que la reivindicación 16 está referida a: *“Un medio de almacenamiento legible*



Resolución N° 74051

Ref. Expediente N° NC2019/0011810

por computadora que almacena (...)” (Radicado N° NC2019/0011810 el 25 de abril de 2022).

La siguiente tabla presenta la comparación entre las características técnicas esenciales de la solicitud estudiada, mencionadas en la reivindicación 16, y las reveladas en los documentos D2 y D3 que representan el estado de la técnica:

Tabla 3. Comparación de las características técnicas esenciales, con las del estado de la técnica:

CARACTERÍSTICAS ESENCIALES	D3	D2
Un medio de almacenamiento legible por computadora que almacena instrucciones ejecutables por computadora que, cuando se ejecutan por una unidad de procesamiento (510; 685), provocan que la unidad de procesamiento (510; 685) realice un método, el método comprende:	(...) medios de almacenamiento legibles por computadora (600) que predicen errores de construcción de software. Un procesador (602) puede acceder a los medios de almacenamiento legibles por computadora tangibles (600) (...) (Pág. 6, Párr. 056).	Un conjunto de estas instrucciones y/o código puede almacenarse en un medio de almacenamiento no transitorio legible por computadora (Pág. 8, Párr. 075).
Generar (220) un vector de características para un problema detectado utilizando información asociada con el problema detectado.	(...) las probabilidades de errores de creación de software se pueden calcular mediante un tipo de regresión o utilizando modelos de aprendizaje automático que incluyen, entre otros, máquinas de vectores de soporte (...) (Pág. 4, Párr. 040). Un vector de características hace parte del conocimiento del técnico medianamente versado en la materia en una máquina de vectores de soporte (SVM) ⁴ , puesto que, para el entrenamiento y clasificación de la SVM, es necesario un vector de características que define el conjunto de propiedades por las cuales serán diferenciados los datos.	No se menciona
Analizar (230) el vector de características por medio de uno o más modelos para determinar la probabilidad de que el problema detectado sea un error provocado por una alteración de al menos una porción del código fuente utilizado por el sistema de cómputo.	(...) el módulo de análisis de predicción (120) informa las probabilidades de que cada cambio en el código de software pueda causar un error de construcción de software (Pág. 4, Párr. 036). Los ejemplos adicionales de características de la lista de cambios pueden incluir cualquier actividad realizada por otros desarrolladores en archivos de código fuente o líneas de archivos de código fuente que se hayan modificado (...). (Pág. 3, Párr. 031) (...) las probabilidades de errores de creación de software se pueden	No se menciona

Resolución N° 74051

Ref. Expediente N° NC2019/0011810

	calcular mediante un tipo de regresión o utilizando modelos de aprendizaje automático que incluyen, entre otros, máquinas de vectores de soporte (...) (Pág. 4, Párr. 040).	
Informar la ocurrencia del problema detectado.	El componente de informes (408) también puede enviar las listas de cambios y los errores de compilación del software a un componente de control de versiones (410) (Pág. 5, Párr. 050).	(...) los informes de errores se suministran a la herramienta (Pág. 3, Párr. 024)
Recibir una entrada que indica si el problema detectado es un falso positivo o un error.	El módulo de interrupción de compilación (122) puede enviar los cambios que causan errores de compilación de software al módulo de actualización (124) (Pág. 3, Párr. 025).	(...) recibir un informe de error de software (Pág. 1, Párr. 005).
Recibir una solución al problema detectado cuando la entrada indica que el problema detectado es un error.	No se menciona	El parche puede incluir información sobre los errores que corrige el parche (Pág. 7, Párr. 066).
Actualizar al menos uno de uno o más modelos utilizando la solución al problema detectado.	El módulo de actualización (122) puede entonces actualizar los datos históricos para reflejar si la lista de cambios provocó un error de construcción de software (Pág. 5, Párr. 044).	Los parches de software relacionados con errores de alta prioridad clasificados y priorizados según las preferencias del cliente pueden agruparse en una o más actualizaciones periódicas de software (Pág. 7, Párr. 065).

D3 es considerado como el estado de la técnica más cercano a la invención definida en la reivindicación 16, ya que divulga un enfoque para determinar una clasificación de un error en un sistema informático (Pág. 1, Resumen).

La diferencia entre la invención definida entre la reivindicación 16 y el medio de almacenamiento divulgado en D3, consiste en recibir una solución al problema detectado.

El efecto que se logra al incluir específicamente el recibir una solución al problema detectado es proporcionar una resolución final al problema detectado.

Por lo tanto, el problema técnico objetivo que pretende resolver esta invención se puede formular así: ¿cómo modificar el medio de almacenamiento divulgado en D3 para proporcionar una resolución final al problema detectado?

Sin embargo, el incluir recibir una solución al problema detectado para proporcionar una resolución final al problema detectado, ya está divulgado en D2 que se refiere a que el parche puede incluir información sobre los errores que corrige el parche (Pág. 7, Párr. 066).

En consecuencia, la persona normalmente versada en la materia estaría motivada a incluir recibir una solución al problema detectado del D2 en el medio de almacenamiento de D3, para así llegar al objeto de la reivindicación 16. Por lo cual se considera obvia.

Resolución N° 74051

Ref. Expediente N° NC2019/0011810

Así mismo, D3 divulga las características de las reivindicaciones dependientes de la siguiente manera:

- Reivindicación 20: (...) las probabilidades de errores de creación de software se pueden calcular mediante un tipo de regresión o utilizando modelos de aprendizaje automático que incluyen, entre otros, máquinas de vectores de soporte (...) (Pág. 4, Párr. 040). Un vector de características hace parte del conocimiento del técnico medianamente versado en la materia en una máquina de vectores de soporte (SVM)⁴, puesto que, para el entrenamiento y clasificación de la SVM, es necesario un vector de características que define el conjunto de propiedades por las cuales serán diferenciados los datos.

De igual manera, D2 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 17: (...) los parches de software relacionados con errores de alta prioridad clasificados y priorizados según las preferencias del cliente pueden agruparse en una o más actualizaciones periódicas de software (Pág. 7, Párr. 065). Los datos pueden almacenarse, analizarse y mostrarse utilizando el portal (506). Los informes de errores, informes de parches, listas de errores y/o parches con puntuaciones y valoraciones pueden mostrarse y estar disponibles para el cliente a través del portal (506) (Pág. 8, Párr. 068).

- Reivindicación 18: El parche puede incluir información sobre los errores que corrige el parche (Pág. 7, Párr. 066).

- Reivindicación 19: Los puntajes de los parámetros recibidos o calculados relacionados con cada error pueden almacenarse en una memoria (...) (Pág. 4, Párr. 040).

Las reivindicaciones dependientes 17 a 20 no mencionan alguna característica que, junto con las características de la reivindicación independiente, aporte nivel inventivo al medio de almacenamiento de la reivindicación 16, por lo cual se considera que carecen de nivel inventivo.

Respuesta a los argumentos del solicitante

El solicitante argumenta que los documentos D1, D2 y D3 no mencionan “(...) *determinar una probabilidad de que el problema detectado sea un error provocado por una alteración de al menos una porción del código fuente utilizado por el sistema de cómputo* (...)”.

Al respecto, resulta conveniente aclarar que el documento D3 hace referencia a que (...) el módulo de análisis de predicción (120) informa las probabilidades de que cada cambio en el código de software pueda causar un error de construcción de software (Pág. 4, Párr. 036). Los ejemplos adicionales de características de la lista de cambios pueden incluir cualquier actividad realizada por otros desarrolladores en archivos de código fuente o líneas de archivos de código fuente que se hayan modificado (...) (Pág. 3, Párr. 031). En estos párrafos se puede apreciar que el módulo de análisis de predicción calcula la probabilidad de un código en el código fuente utilizado en el sistema que pueda causar un error de construcción de software, es decir, la probabilidad de un problema detectado correspondiente a un error de alteración del código, por lo que esta característica se anticipa por el documento D3 tal como se aprecia en el análisis de nivel inventivo realizado.



Resolución N° 74051

Ref. Expediente N° NC2019/0011810

A continuación, el solicitante argumenta que los documentos D1, D2 y D3 no mencionan *“(...) analizar (230) el vector de características para determinar una probabilidad de que el problema detectado sea un error provocado por una alteración de al menos una porción del código fuente utilizado por el sistema de cómputo (...)”*.

Se aclara que el documento D3 hace mención de que (...) el módulo de análisis de predicción (120) informa las probabilidades de que cada cambio en el código de software pueda causar un error de construcción de software (Pág. 4, Párr. 036). Los ejemplos adicionales de características de la lista de cambios pueden incluir cualquier actividad realizada por otros desarrolladores en archivos de código fuente o líneas de archivos de código fuente que se hayan modificado (...) (Pág. 3, Párr. 031). (...) las probabilidades de errores de creación de software se pueden calcular mediante un tipo de regresión o utilizando modelos de aprendizaje automático que incluyen, entre otros, máquinas de vectores de soporte (...) (Pág. 4, Párr. 040). En estos párrafos se puede apreciar nuevamente que el módulo de análisis de predicción se determina la probabilidad de un error de construcción de software (problema detectado) a partir de alteraciones en el código fuente, además que estas probabilidades pueden ser calculadas a partir de modelos de aprendizaje automático tal como una máquina de vector de soporte, siendo que un vector de características hace parte del conocimiento del técnico medianamente versado en la materia en una máquina de vectores de soporte (SVM)⁴, puesto que, para el entrenamiento y clasificación de la SVM, es necesario un vector de características que define el conjunto de propiedades por las cuales serán diferenciados los datos, por lo cual esta característica es igualmente anticipada por el documento D3 y se mantienen las objeciones por nivel inventivo.

Posteriormente, el solicitante argumenta que los documentos D1, D2 y D3 no mencionan *“(...) “utilizar la entrada para actualizar la información sobre el problema detectado (...)”*.

Al respecto, es pertinente aclarar que el documento D3 menciona que el módulo de actualización (122) puede entonces actualizar los datos históricos para reflejar si la lista de cambios provocó un error de construcción de software (Pág. 5, Párr. 044). En este párrafo se puede apreciar que el módulo (122) realiza la actualización de datos históricos de acuerdo con el error de construcción de software generado por la alteración del código fuente, es decir, del problema detectado, por lo que esta característica se anticipa de igual manera por el documento D3 tal como se evidencia en el estudio de nivel inventivo realizado.

Por otro lado, el solicitante argumenta que *“(...) La invención divulgada en el documento D3, entre otras diferencias, no hace una detección previa de un problema, ni tampoco determina si el origen del problema es una modificación por alteración de código. Lo que hace la invención del documento D3 es identificar los cambios en el software y, para cada cambio o para una lista de cambios, calcula la probabilidad de que dicho cambio pueda crear un error de construcción de software (ver párrafos [0014], [0032] y [0035] del documento D3). Nótese particularmente que la invención del documento D3 calcula dicha probabilidad para los cambios de software, lo que es diferente a la invención de la presente solicitud que calcula la probabilidad para un problema detectado (...)”* - Negrita y subrayado original del texto-.

Se aclara nuevamente que al divulgar el documento D3 que (...) el módulo de análisis de predicción (120) informa las probabilidades de que cada cambio en el código de software pueda causar un error de construcción de software (Pág. 4, Párr. 036). Los ejemplos adicionales de características de la lista de cambios pueden incluir cualquier actividad

Resolución N° 74051

Ref. Expediente N° NC2019/0011810

realizada por otros desarrolladores en archivos de código fuente o líneas de archivos de código fuente que se hayan modificado (...) (Pág. 3, Párr. 031), este menciona que el módulo de análisis de predicción calcula la probabilidad de un cambio en el código fuente que ocasiona un error en la construcción del software, lo cual implica en un problema que se detecta al encontrar un error por alteraciones del código, por lo cual se reitera que esta característica se anticipa por el documento D3 y se mantienen las objeciones por nivel inventivo de acuerdo con el artículo 18 de la Decisión 486.

De otra parte, el solicitante argumenta que el documento D3 no menciona ni sugiere “(...) generar (220) un vector de características para el problema detectado (...)”.

Al respecto, cabe aclarar que el documento D3 hace referencia a que (...) las probabilidades de errores de creación de software se pueden calcular mediante un tipo de regresión o utilizando modelos de aprendizaje automático que incluyen, entre otros, máquinas de vectores de soporte (...) (Pág. 4, Párr. 040). En este párrafo se evidencia tal como se mencionó anteriormente, que las probabilidades de un cambio en el código fuente que indica un error de construcción de software (problema detectado) pueden ser calculadas a partir de modelos de aprendizaje automático tal como una máquina de vector de soporte, siendo que un vector de características hace parte del conocimiento del técnico medianamente versado en la materia en una máquina de vectores de soporte (SVM)⁴, puesto que, para el entrenamiento y clasificación de la SVM, es necesario un vector de características que define el conjunto de propiedades por las cuales serán diferenciados los datos, por lo que de igual manera el documento D3 anticipa esta característica y junto con la combinación de enseñanzas de los documentos D1 y D2 se anticipan todas las características esenciales de la solicitud.

Adicionalmente, el solicitante argumenta que el documento D3 no menciona ni sugiere “(...) recibir información sobre un problema detectado (...)”.

Se aclara que el documento D3 hace mención de que el módulo de interrupción de compilación (122) puede enviar los cambios que causan errores de compilación de software al módulo de actualización (124) (Pág. 3, Párr. 025). En este párrafo se evidencia que el módulo de actualización recibe información sobre los cambios o alteraciones del código fuente que pueden causar errores de compilación o de construcción de software y como fue indicado anteriormente, estos cambios que generan errores corresponden a errores detectados sobre el funcionamiento del software, por lo que esta característica se anticipa igualmente por el documento D3 tal como se evidencia en el análisis de nivel inventivo realizado.

Por otra parte, el solicitante afirma que “(...) Aunque el oficio N° 8374 hace referencia al párrafo [0065] del documento D1 indicando que menciona que “los parches de software relacionados con errores de alta prioridad clasificados y priorizados según las preferencias del cliente pueden agruparse en una o más actualizaciones periódicas de software”, dicha referencia está equivocada debido a que el documento D1 no tiene párrafo [0065] y no menciona lo anterior, donde dicha referencia en realidad se refiere al documento D2 (...)” -Subrayado original del texto-.

Se aclara que tal como lo menciona el solicitante, el párrafo 065 trasladado corresponde al documento D2 y no al documento D1, por lo que se ha realizado la corrección correspondiente y se considera superada esta objeción.



Resolución N° 74051

Ref. Expediente N° NC2019/0011810

A continuación, el solicitante argumenta que “(...) El párrafo [0065] del documento D2 se refiere a actualizaciones y parches de software que actualizan el software que no corresponden a actualizaciones de información sobre un problema detectado y mucho menos corresponden a actualizaciones de información sobre un problema detectado para entrenar el sistema (...)” -Subrayado original del texto-.

Al respecto, resulta del caso aclarar que tal como lo menciona el solicitante, el documento D2 no hace referencia explícita a actualizaciones de información sobre un problema detectado para entrenar el sistema, por lo que dicha objeción se considera superada y el análisis de nivel inventivo sobre las reivindicaciones independiente 1 y 8 se ha actualizado sin incluir dicha afectación.

Finalmente, el solicitante menciona que “(...) *De los argumentos aquí presentados, para conceder el privilegio de patente solicitado para la materia objeto de las 20 reivindicaciones ahora presentadas en el capítulo reivindicatorio modificado, y que de subsistir alguna duda o inquietud respecto a la patentabilidad de la presente invención, a pesar de los argumentos técnicos ahora presentados, sin apartarse del espíritu inventivo originalmente revelado en la solicitud internacional, se le brinde la oportunidad al solicitante de aclararla o de limitar la invención, mediante la emisión de un segundo examen de fondo, teniendo en cuenta el nuevo capítulo reivindicatorio que se somete a revisión antes de que se emita una decisión definitiva por parte de este Despacho, de conformidad con lo dispuesto en 1.2.2.5.1 del artículo primero de la Resolución 54093 de 2012* (...)”.

Esta Oficina aclara que ya se han realizado tres exámenes de patentabilidad de la presente solicitud, y que esta Oficina no considera necesario realizar un cuarto examen de patentabilidad teniendo en cuenta que no se presentó una modificación sustancial del capítulo reivindicatorio, ni el pago de la tasa correspondiente a un nuevo examen de patentabilidad; además de que es posible seguir empleando los documentos citados en el Oficio N° 8374, ante los cuales el solicitante ya tuvo la oportunidad de exponer sus argumentos.

Que de acuerdo con la modificación presentada por el solicitante en el escrito radicado bajo el N° NC2019/0011810 el 27 de diciembre de 2021 y teniendo en cuenta el objeto concedido, se modifica el título de la invención el cual quedará de la siguiente manera: “MÉTODO, SISTEMA Y MEDIO DE ALMACENAMIENTO LEGIBLE POR COMPUTADORA PARA TOMA DE DECISIONES MEDIANTE APRENDIZAJE AUTOMÁTICO PARA ALERTAS ORIGINARIAS DE SISTEMAS DE MONITORIZACIÓN”.

Con fundamento en las anteriores consideraciones, el Superintendente de Industria y Comercio (E),

RESUELVE

ARTÍCULO 1: Denegar la patente de invención a la solicitud que entró en fase nacional en virtud del Tratado de Cooperación en Materia de Patentes (PCT), para la creación titulada:

**“MÉTODO, SISTEMA Y MEDIO DE ALMACENAMIENTO LEGIBLE POR
COMPUTADORA PARA TOMA DE DECISIONES MEDIANTE APRENDIZAJE**

Resolución N° 74051

Ref. Expediente N° NC2019/0011810

***AUTOMÁTICO PARA ALERTAS ORIGINARIAS DE SISTEMAS DE
MONITORIZACIÓN***

ARTÍCULO 2: Notificar el contenido de la presente resolución a MICROSOFT TECHNOLOGY LICENSING LLC advirtiéndole que contra ella procede el recurso de reposición, ante el Superintendente de Industria y Comercio, el cual podrá ser interpuesto en el momento de la notificación o dentro de los diez (10) días hábiles siguientes a ella.

NOTIFÍQUESE Y CÚMPLASE

Dada en Bogotá D.C., el 24 de octubre de 2022

EL SUPERINTENDENTE DE INDUSTRIA Y COMERCIO (E),



Firmado digitalmente
por: JUAN CAMILO
DURAN TELLEZ
Fecha: 2022.10.24
09:02:12 COT
Razón: Validez jurídica.
Ubicación: Bogotá,
Colombia.

