

Señor  
DIRECTOR DIVISIÓN DE NUEVAS CREACIONES  
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO  
E. S. D.

REF: Expediente No. **NC2022/0000054**, solicitud de patente de invención a favor de **Luis Eduardo GUTIERREZ-SHERIS**, Título: **SISTEMA Y MÉTODO MEDIANTE EL USO DE UN CONSENSO DE GRADIENTE DE IDONEIDAD DE LA CADENA DE BLOQUES**

## **RESPUESTA A EXAMEN DE FONDO**

**JUAN PABLO CADENA SARMIENTO**, portador de la tarjeta profesional No. 57.492 del C.S.J., obrando como apoderado de la sociedad solicitante me permito dar respuesta al Oficio No. 20602 relacionado con la patentabilidad de la presente invención y notificado el 26 de noviembre de 2024.

En respuesta a las observaciones realizadas como resultado del examen de fondo, es pertinente mencionar que las reivindicaciones cumplen a cabalidad con los requerimientos de patentabilidad establecidos en la Decisión 486 de 2000, razón por la que respetuosamente disiento de los comentarios del Examinador a este respecto. A continuación, sin embargo, se presentan argumentos y se mencionan las modificaciones realizadas para atender a cada uno de los requerimientos del Examinador:

### **1. Modificación a las reivindicaciones**

Me permito dirigir la atención del Señor Examinador en el hecho que el Capítulo Reivindicatorio adjunto al presente memorial ha sido modificado de la siguiente manera:

- Las antiguas reivindicaciones 2 y 22 a 33 han sido eliminadas del Capítulo Reivindicatorio adjunto al presente memorial de respuesta. En consecuencia, la numeración y dependencia de las reivindicaciones ha sido modificada de conformidad con lo anteriormente mencionado.

- Se ha modificado la reivindicación 1 incluyendo las características de la antigua reivindicación 2, con el fin de definir la materia a proteger.
- Se ha reemplazado la expresión “*retroconectados*” por la expresión “*hacia atrás*” en la reivindicación 1, de acuerdo con lo divulgado en la descripción.
- Se ha reemplazado la expresión “*protocolo de chismes*” por la expresión “*protocolo gossip*” en las antiguas reivindicaciones 3 a 5 (nuevas reivindicaciones 2 a 4), de acuerdo con lo divulgado en la descripción. La anterior modificación se realiza ya que se presentó un error de traducción en las reivindicaciones.
- Se ha reemplazado la expresión “*entre homólogos*” por la expresión “*peer-to-peer*” de acuerdo con lo divulgado en la descripción y así corregir un error de traducción.
- Se ha reemplazado la expresión “*credenciales*” por la expresión “*tokens*” de acuerdo con lo divulgado en la descripción.
- Se incluyeron los números de referencia de los elementos de la invención de acuerdo con lo establecido en las Figuras y siguiendo las sugerencias del examinador.

Así las cosas, es importante mencionar que las modificaciones realizadas a las reivindicaciones cumplen en su totalidad con los requisitos establecidos en el Artículo 34 de la Decisión 486, teniendo en cuenta que estas **no involucran una ampliación a la protección correspondiente a la divulgación tal y como se presentó originalmente.** Por el contrario, el Capítulo Reivindicatorio Modificado y adjunto al presente memorial cumple con los lineamientos establecidos por la Decisión Andina 486 y encuentra sustento a lo largo de la memoria descriptiva de la solicitud.

## 2. Claridad del Capítulo Reivindicatorio

Con respecto a la redacción de “*interconectados retroconectados*” y “*computadors*” en la reivindicación 1, me permito señalar que se ha reemplazado la expresión “*retroconectados*” por la expresión “*hacia atrás*” de acuerdo con lo divulgado en la descripción. De igual manera se reemplazó el término “*computadors*” por “*computadoras*” y corregir el error tipográfico. Por lo tanto, se superan dichas objeciones.

En cuanto al término “*protocolo de chismes*”, me permito señalar que ha sido reemplazado por la expresión “*protocolo gossip*” de acuerdo con lo divulgado en la descripción y así superar el error de traducción. Por lo cual, se solicita el retiro de dicha objeción.

Considerando las sugerencias del Examinador y el hecho que la materia reclamada en el Capítulo Reivindicatorio adjunto al presente memorial cumple con las disposiciones del Artículo 30 de la Decisión 486, y corresponden a una limitación razonable del concepto inventivo desarrollado por el inventor, de manera atenta solicito a ese Despacho retirar las objeciones en cuanto a la claridad de la materia reclamada.

### 3. Descripción

Con el fin de corregir un error en la traducción de la solicitud presentada inicialmente, me permito presentar junto a este memorial, una copia de la descripción modificada en donde se reemplazó el término “*par a par*” por el término “*peer-to-peer*”, de acuerdo con lo divulgado en la solicitud PCT.

### 4. Nivel Inventivo

Con respecto al nivel inventivo de la presente solicitud, el Examinador asegura que las reivindicaciones **1 a 21** carecen de nivel inventivo en vista de las enseñanzas de los documentos **D1** y **D2**.

En respuesta a lo anterior, de manera atenta me permito manifestar que contrario a los señalamientos establecidos por el Examinador, la presente solicitud, tal y como se encuentra reclamada en el Capítulo Reivindicatorio adjunto a este memorial goza de nivel inventivo y presenta diferencias con los documentos del arte previo citados en el Oficio Técnico No. 20602.

En primer lugar, me permito aclarar que el Capítulo Reivindicatorio ha sido modificado y limitado a elementos específicos que permiten definir la materia reivindicada en la presente solicitud haciendo que difiera completamente de las enseñanzas de **D1** y **D2**.

Además, me permito señalar que el campo al que se refiere la invención reclamada se encuentra dentro de la técnica de *blockchain* (cadena de bloques); particularmente en lo relacionado con métodos de consenso de *Gradiente de Idoneidad*. En este sentido, y considerando que este tipo de métodos tienen como objetivo que cualquier intercambio de información (por ejemplo una *transacción de información*) realizada mediante *blockchain* debe ser validada para cumplir y/o incrementar parámetros de seguridad/veracidad

establecidos mediante la regulación de los nodos involucrados, la invención reivindicada en la presente solicitud proporciona una metodología de configuración técnica de *un libro mayor electrónico, de bloques interconectados, de un sistema de cadena de bloques, de uno o más nodos (denominados de billetera), y de uno o más nodos de construcción de bloques*, con el objetivo de la implementación de un proceso (*proceso de propagación de datos de confiar, pero verificar*) en donde al menos un nodo de construcción de bloques está configurado para crear al menos un nuevo bloque interconectado en el sistema de cadena de bloques antes de verificar la corrección de uno o más bloques precedentes a los que dicho al menos un nuevo bloque interconectado está vinculado, lo cual no se encuentra divulgado en el arte previo citado.

Ahora bien, como se puede observar en la nueva reivindicación 1, el método implementado por computador se limitó mediante la característica de *“implementar en el sistema de cadena de bloques, un proceso de propagación de datos de confiar, pero verificar, en donde al menos un nodo de construcción de bloques está configurado para crear al menos un nuevo bloque interconectado en el sistema de cadenas de bloques antes de verificar la corrección de uno o más bloques precedentes a los que se vincula dicho al menos un nuevo bloque interconectado”*. En este sentido, el proceso de propagación de datos de confiar, pero verificar, se utiliza para determinar cuál de los bloques o cadenas que compiten se agrega a la cadena de bloques o al libro mayor distribuido, y cuál de los nodos que compiten se selecciona para recibir una recompensa por el bloque o cadena que se agrega a la cadena de bloques o al libro mayor distribuido, el cual **aumenta significativamente el rendimiento transaccional de los sistemas actualmente conocidos al reducir las restricciones de linealidad (y otras restricciones informáticas que retrasan el procesamiento general del sistema) y permite que el rendimiento de procesamiento del sistema se aproxime al rendimiento máximo teórico para cadenas de bloques y sistemas de libro mayor electrónico distribuido en general.**

En segundo lugar, con respecto al arte previo citado, me permito señalar que **D1** se dirige a un sistema de consenso para cadenas de bloques que carecen de una autoridad central de confianza. También establece que la pluralidad de dispositivos conectados a la red puede entonces determinar un candidato principal para enviar un siguiente bloque de datos para agregar o anexar al libro de contabilidad distribuido, utilizando un criterio de selección que puede comprender seleccionar un minero cuyo mensaje de anuncio produzca uno de los siguientes: una diferencia más pequeña, una diferencia más grande, una diferencia más cercana a un valor objetivo, una diferencia más cercana al valor objetivo módulo una constante, una diferencia calculada utilizando una función que toma como entrada el valor de la salida hash y el valor del hash de la clave pública y el identificador de dirección único.

Sin embargo, **D1** no divulga o sugiere un esquema de confiar, pero verificar y tampoco proporciona alguna enseñanza relacionada con dichos esquemas ni con que el libro mayor electrónico distribuido tiene una pluralidad de bloques interconectados hacia atrás, dispuestos como una o más instancias de estructuras de datos de cadena de bloques lineales, arreglos de bloques no lineales, mallas n-dimensionales o estructuras de datos reticulares, o gráficos acíclicos dirigidos, tal como se requiere en la reivindicación 1 actualmente reclamada. Por lo tanto, la presente solicitud es inventiva con respecto a las enseñanzas de **D1**.

Por otro lado, y con el fin de “*curar*” las deficiencias de **D1**, el examinador cita el documento **D2** porque describe un sistema mejorado para procesar transacciones con una moneda criptográfica. También enseña que el sistema utiliza un protocolo de cadena de bloques como registro público de transacciones para garantizar que solo se puedan usar tokens válidos como entrada de transacción y que solo se puedan usar una vez. Los testigos integran las transacciones en una cadena de bloques y una vez que suficientes testigos confirman un bloque, este se convierte en una parte permanente e indeleble de la cadena de bloques, y por lo cual, un experto en la materia podría llegar a la materia reivindicada en la presente solicitud de forma obvia. Sin embargo, **D2** no divulga o sugiere un esquema de confiar, pero verificar y tampoco proporciona alguna enseñanza relacionada con dichos esquemas ni con que el libro mayor electrónico distribuido tiene una pluralidad de bloques interconectados hacia atrás, dispuestos como una o más instancias de estructuras de datos de cadena de bloques lineales, arreglos de bloques no lineales, mallas n-dimensionales o estructuras de datos reticulares, o gráficos acíclicos dirigidos, tal como se requiere en la reivindicación 1 actualmente reclamada. Por lo cual, es evidente que las enseñanzas de **D1** y **D2** solos o en combinación no habrían motivado a un experto en la materia para llegar de forma obvia a la invención tal como se reivindica actualmente. Por tanto, se evidencia una actividad inventiva de la presente solicitud en vista del arte previo citado **D1** y **D2**.

En tercer lugar, considerando lo indicado en el Manual para el Examen de Solicitudes de Patentes de Invención en las Oficinas de Propiedad Industrial de los países de la Comunidad Andina que establece que *“Para juzgar si la invención definida por las reivindicaciones realmente se deriva de manera evidente del estado de la técnica, hay que determinar si carece de nivel inventivo cuando se consideran las diferencias entre ésta y el estado de la técnica más cercano. El examinador tiene la carga de probar que la invención carece de nivel inventivo y no sólo limitarse a establecer las diferencias entre la solicitud y dicho estado de la técnica”* (subrayado y negrilla fuera de texto), me permito llamar la atención de ese Despacho en el hecho que cada invención es de cierta medida una combinación de elementos relacionados entre sí, por lo que no es suficiente

afirmar que dichos elementos y/o características ya habían sido previamente divulgados, partiendo de combinaciones de enseñanzas convenientes, y que por ende, la combinación de fracciones de diferentes documentos permiten la deducción obvia de la invención reclamada, por lo tanto, el examinador debe demostrar con argumentos válidos, **por qué** un experto en la técnica habría llegado a la invención reivindicada de **forma obvia** basándose en las enseñanzas de **D1 y D2**.

Así las cosas, es claro que el Examinador basó su análisis solo en la comparación de las diferencias de la presente solicitud con el documento **D1** y la búsqueda de otro documento que “**curara**” dichas diferencias. Además, el hecho de que un experto en la materia pueda modificar las características específicas necesarias, como lo asume el Examinador, para llegar al objeto reivindicado no significa que el experto en la materia lo haría con la expectativa de alguna mejora, ya que en ninguno de los documentos citados se sugieren los mismos elementos y características del método implementado por computador reivindicado en la presente solicitud, ni mucho menos los efectos que este tendría.

Además, **D1** y **D2** no establecen una mínima indicación de qué modificaciones debería efectuar una persona versada en la materia para poder obtener el método implementado por computador aquí reclamado de acuerdo con el nuevo Capítulo Reivindicatorio adjunto a este memorial.

De esta forma, me permito señalar que la sola similitud entre el método reclamado y aquellos del arte previo (*si es que la hay*) es **insuficiente** para determinar o concluir una supuesta falta de nivel inventivo y consideramos entonces que en un juicioso análisis de nivel inventivo **debió haber alguna razón o motivación** que hubiese llevado a un experto en la técnica a **modificar el estado de la técnica más cercano para llegar a la invención reivindicada de forma obvia**. Hacemos énfasis también en que el examinador no ha proporcionado ninguna evidencia o razón válida por la cual un experto en la técnica se habría sentido motivado a realizar las modificaciones necesarias a los métodos de la técnica anterior para llegar de forma obvia a la presente invención.

Por lo tanto, el análisis de nivel inventivo de solicitudes de patente, no se trata de encontrar en el arte previo **una simple invitación a experimentar con uno o más cambios aleatorios**, sino de identificar en los documentos citados una enseñanza y guía específica para hacer las modificaciones necesarias que lo lleven con certeza a la materia reclamada, lo cual no sucede en este caso. Por lo cual, la materia reivindicada en la nueva

reivindicación **1** difiere de gran manera con aquella divulgada en el arte previo y, por tanto, es inventiva a la luz de las enseñanzas de **D1** y **D2**.

Por cualquiera de las razones anteriores, es más que evidente que el análisis de nivel inventivo no puede ser basado en un estándar “*posible de alcanzar*”, debido a que esto haría que toda invención fuera no patentable. Así, como se resalta en el Artículo 18 de la Decisión 486, las invenciones deberían considerarse como inventivas cuando la materia reclamada no puede ser derivada del arte previo. Adicionalmente, la presente invención es inventiva a la luz de los documentos citados, pues siguiendo las enseñanzas de **D1** y **D2**, la persona medianamente versada en la materia no obtendría la materia modificada y aquí reclamada, ni podría esperar que ésta presentase los efectos presentados y que se han discutido anteriormente.

Así las cosas, y en vista de lo anteriormente expuesto, respetuosamente solicito el retiro de las objeciones de la presente solicitud y la concesión del privilegio solicitado, ya que en opinión del solicitante cumple con todos los requisitos establecidos en la Decisión 486.

## 5. Anexos

- Capítulo Reivindicatorio Modificado.
- Capítulo Descriptivo Modificado.
- Tasa Oficial de modificación a las reivindicaciones.

**JUAN PABLO CADENA SARMIENTO**

T.P. No. 57.492 del C.S.J.  
C.C. 80.410.337 de Usaquén  
Calle 70 A No. 4 – 41



## REIVINDICACIONES MODIFICADAS

1. Un método implementado por computador en una red informática, comprendiendo el método implementado por computador:

configurar un libro mayor electrónico distribuido en la memoria electrónica (114) de una o más computadoras (110) en un sistema de cadenas de bloques (100), el libro mayor electrónico distribuido tiene una pluralidad de bloques interconectados hacia atrás (200), dispuestos como una o más instancias de estructuras de datos de cadena de bloques lineales, arreglos de bloques no lineales, mallas n-dimensionales o estructuras de datos reticulares, o gráficos acíclicos dirigidos;

configurar cada uno de los bloques interconectados (210A, 210B, 210C) para incluir un conjunto ordenado de registros de datos individuales (220A-C), de modo que al menos uno de los registros refleje una transformación de al menos una parte del estado global del libro mayor electrónico distribuido;

configurar el sistema de cadenas de bloques (100) para que comprenda una red peer-to-peer de nodos de computador, con una o más computadoras configuradas como nodos billetera (120), y con una o más computadoras configuradas como nodos de construcción de bloques (110);

configurar el uno o más nodos de billetera (120) para transmitir uno o más de los registros de datos a uno o más nodos de construcción de bloques (110) en la red peer-to-peer;

configurar el uno o más nodos de construcción de bloques (110) para construir uno o más bloques interconectados (210A, 210B, 210C) en el libro mayor electrónico distribuido seleccionando y ordenando uno o más de los registros de datos para incluirlos en dicho uno o más nuevos bloques interconectados; e

implementar en el sistema de cadenas de bloques (100), un proceso de propagación de datos de confiar, pero verificar, en donde al menos un nodo de construcción de bloques (110) está configurado para crear al menos un nuevo bloque interconectado en el sistema de cadenas de bloques (100) antes de verificar la corrección de uno o más bloques precedentes a los que dicho al menos un nuevo bloque interconectado se vincula.

2. El método implementado por computador de la reivindicación 1, en donde el proceso de propagación de datos de confiar, pero verificar incluye además propagar dicho al menos un nuevo bloque interconectado en el sistema de cadenas de bloques (100) a través de un protocolo gossip.

3. El método implementado por computador de la reivindicación 2, en donde el protocolo gossip permite que dicho al menos un nuevo bloque interconectado se propague a través del sistema de cadenas de bloques (100), que incluye, además:

transmitir dicho al menos un bloque interconectado a través del protocolo gossip como un encabezado de bloque, mientras que se excluye la mayor parte de los datos pertenecientes a dicho al menos un bloque interconectado.

4. El método implementado por ordenador de la reivindicación 2, en donde el protocolo gossip permite transmitir dicho al menos un nuevo bloque interconectado sin que uno o más registros de datos pertenezcan a dicho al menos un nuevo bloque interconectado, de modo que dichos uno o más registros de datos se transmiten separadamente de dicho al menos un nuevo bloque interconectado.

5. El método implementado por computador de la reivindicación 1, que además incluye:

al menos otro nodo de construcción de bloques (110) creando al menos un siguiente bloque interconectado encima de dicho al menos un nuevo bloque interconectado después de que dicho al menos un nuevo bloque interconectado se distribuye al sistema de cadenas de bloques; y

el al menos otro nodo de construcción de bloques creando simultáneamente el al menos un siguiente bloque interconectado mientras que uno o más registros de datos pertenecientes a dicho al menos un nuevo bloque interconectado son descargados por el al menos otro nodo de construcción de bloques (110), o mientras dicho al menos un nuevo bloque interconectado se valida.

6. El método implementado por computador de la reivindicación 1, en donde el proceso de propagación de datos de confiar, pero verificar incluye además hacer que el al menos un nodo de construcción de bloques (110) incluya un filtro Bloom en un encabezado del bloque de dicho al menos un nuevo bloque interconectado en el sistema de cadenas de bloques (100), de modo que el proceso de propagación de datos de confiar, pero verificar hace que el al menos un nodo de construcción de bloques (110) distribuya el filtro Bloom con el al menos un nuevo bloque interconectado en el sistema de cadenas de bloques (100).

7. El método implementado por computador de la reivindicación 6, en donde el filtro Bloom está configurado para representar un conjunto de cuentas contables electrónicas distribuidas u otros datos afectados u operados durante uno de (i) la creación, (ii) la validación, o (iii) la ejecución de dicho al menos un nuevo bloque interconectado en el sistema de cadenas de bloques (100).

8. El método implementado por computador de la reivindicación 6, en donde el filtro Bloom de un bloque es una estructura de datos configurada al menos

con la propiedad de que cualquier cuenta u otros datos encontrados en el filtro Bloom es posible o probable que tenga su estado modificado por la validación y ejecución del bloque, o que haya sido operado o afectado de otra manera por el bloque o por registros de datos pertenecientes a el bloque, y

con la propiedad de que cualquier cuenta que no se encuentre en el filtro Bloom seguramente no cambiará en cuanto a su estado o no se verá afectada por el bloque o por los registros de datos que pertenecen al bloque.

9. El método implementado por computador de la reivindicación 8, en donde crear dicho al menos un nuevo bloque interconectado incluye, además:

identificar cuentas afectadas u operadas cuando se ejecutan uno o más contratos inteligentes, o cuando se agregan uno o más registros de datos a dicho al menos un nuevo bloque interconectado;

comprobar para asegurar que cada cuenta identificada no esté contenida dentro de un filtro Bloom de al menos un bloque interconectado anterior que precede a dicho al menos un nuevo bloque interconectado; y

si una cuenta identificada está contenida dentro de un filtro Bloom del al menos un bloque interconectado anterior, detener la ejecución de dicho uno o más contratos inteligentes o detener la inclusión de dicho uno o más registros de datos, y descartar dicho uno o más contratos inteligentes o uno o más registros de datos de dicho al menos un nuevo bloque interconectado.

10. El método implementado por ordenador de la reivindicación 1, en donde el proceso de propagación de datos de confiar, pero verificar utiliza la estructura de datos de gráficos acíclicos dirigidos (DAG) para disponer dicho al menos un nuevo bloque interconectado.

11. El método implementado por computador de la reivindicación 10, en donde el proceso de propagación de datos de confiar, pero verificar permite que dicho al menos un nuevo bloque interconectado tenga referencias posteriores a uno o más bloques interconectados (210A, 210B, 210C) o subcadenas en el sistema de cadenas de bloques (100), provocando que al menos una parte del libro mayor electrónico distribuido constituya un gráfico acíclico dirigido (DAG).

12. El método implementado por computador de la reivindicación 11, en donde el proceso de propagación de datos de confiar, pero verificar asegura que los múltiples bloques interconectados (210A, 210B, 210C) o subcadenas que se construyen simultáneamente por dicho nuevo bloque interconectado incluyen registros asociados con conjuntos distintos y mutuamente excluyentes de cuentas u otros datos, de modo que los múltiples bloques interconectados o subcadenas que se construyen afectan a conjuntos de cuentas mutuamente excluyentes u otros datos.

13. El método implementado por computador de la reivindicación 12, en donde el proceso de propagación de datos de confiar, pero verificar permite que los múltiples bloques interconectados (210A, 210B, 210C) o subcadenas se comparen rápidamente para determinar si alguno de los múltiples bloques interconectados o subcadenas opera en cuentas comunes u otros datos.

14. El método implementado por computador de la reivindicación 12, que además incluye la comparación de los respectivos filtros Bloom de los múltiples bloques interconectados (210A, 210B, 210C) o subcadenas para determinar con certeza que los bloques interconectados o subcadenas en

el sistema de cadenas de bloques (100) afectan u operan sobre conjuntos de cuentas mutuamente excluyentes u otros datos.

15. El método implementado por computador de la reivindicación 1, en donde el proceso de propagación de datos de confiar, pero verificar permite la creación simultánea de una pluralidad de nuevos bloques interconectados en el libro mayor electrónico distribuido, de modo que múltiples nodos en el sistema de cadenas de bloques (100) están creando uno o más de la pluralidad de nuevos bloques interconectados en paralelo; y

en donde el proceso de propagación de datos de confiar, pero verificar garantiza que los registros de datos no se propaguen ampliamente a través del sistema de cadenas de bloques (100) antes de la creación de bloques interconectados, de modo que los registros de datos producidos por un nodo determinado solo se comparten con otros nodos determinados dentro del sistema de cadenas de bloques (100); y

en donde el proceso de propagación de datos de confiar, pero verificar garantiza que un nodo de construcción de bloques (110) respectivo solo cree dicho al menos un nuevo bloque interconectado utilizando registros de datos propagados por ciertos nodos dentro del sistema de cadenas de bloques (100).

16. El método implementado por computador de la reivindicación 1, en donde si un nodo de construcción de bloques (110) descubre que dicho al menos un nuevo bloque interconectado no se puede validar, dicho al menos un nuevo bloque interconectado se descarta junto con cualquier bloque nuevo o subsiguiente que se construya sobre dicho al menos un nuevo bloque interconectado; y

en donde un bloque no se puede validar si ocurre uno de los siguientes: (i) una raíz de árbol Merkle (225A-D) no se puede reproducir utilizando un conjunto de registros de datos (220A-C) que pertenecen a dicho bloque en el sistema de cadenas de bloques (110), (ii) un hash de bloque (240A-C) en dicho bloque se especifica incorrectamente, o (iii) un registro de datos perteneciente a dicho bloque no es válido o viola una regla en el sistema de cadenas de bloques.

17. El método implementado por computador de la reivindicación 1, en donde el proceso de propagación de datos de confiar, pero verificar aumenta el rendimiento de transacciones del sistema de cadenas de bloques (100) en relación con el rendimiento de transacciones de un sistema de cadenas de bloques de prueba de trabajo de consenso de Nakamoto.

18. El método implementado por computador de la reivindicación 2, en donde el al menos un nodo de construcción de bloques (110) es un nodo de construcción de bloques identificable configurado con una dirección de recompensa que se incluye con al menos un bloque interconectado sujeto que está siendo construido por dicho al menos un nodo de construcción de bloques identificable; y

en donde dicho al menos un bloque interconectado sujeto es un bloque firmado criptográficamente por dicho al menos un nodo de construcción de bloques identificable que utiliza al menos una clave criptográfica asociada con dicha dirección de recompensa,

en donde el al menos un bloque firmado se propaga en la red de cadena de bloques.

19. El método implementado por computador de la reivindicación 18, en donde al menos otro nodo de construcción de bloques (110) implementa un proceso de evaluación de confianza, mediante el cual, dicho al menos otro nodo de construcción de bloques, al recibir dicho bloque firmado, determina si generar un nuevo bloque subsiguiente que se basa en dicho bloque firmado o encabezado de bloque antes de que dicho bloque firmado o encabezado de bloque sea validado;

en donde el proceso de evaluación de confianza incluye además evaluar si dicha dirección de recompensa ha firmado algún bloque anterior propagado previamente en la red de cadena de bloques, y en donde dichos bloques anteriores firmados por dicha dirección de recompensa son válidos o no válidos.

20. El método implementado por computador de la reivindicación 18, en donde configurar el proceso de propagación de datos de confiar, pero verificar incluye además al menos uno de:

(a) configurar el bloque firmado para especificar una tarifa de construcción como una cantidad de tokens de libro mayor electrónico distribuidos, cuyo bloque firmado está configurado como un bloque incentivado, y al menos otro nodo de construcción de bloques genera un nuevo bloque posterior configurado con una referencia a dicho bloque incentivado;

en donde la tarifa de construcción de bloques incentivada se transfiere desde la dirección de recompensa a una dirección de libro mayor distribuida especificada por el al menos otro nodo de construcción de bloques; o

(b) configurar el bloque firmado para especificar tokens en riesgo como una cantidad de tokens de libro mayor electrónico distribuidos, cuyo bloque firmado funciona como un bloque garantizado, y configurar al menos otro nodo en la red de cadena de bloques construye un registro de penalización que contiene o hace referencia al bloque garantizado,

cuyo registro de penalización, cuando se incorpora a un nuevo bloque posterior, hace que los tokens en riesgo especificados se transfieran desde la dirección de recompensa a una dirección de libro mayor distribuida especificada por el al menos otro nodo en la red de cadena de bloques, si el bloque garantizado es inválido.

## **SISTEMA Y MÉTODO MEDIANTE EL USO DE UN CONSENSO DE GRADIENTE DE IDONEIDAD DE LA CADENA DE BLOQUES**

### **CAMPO DE LA INVENCION**

La presente invención generalmente se refiere a un campo de tecnología de la cadena de bloques, que también se conoce como un libro mayor electrónico distribuido para almacenar datos a los que múltiples partes pueden acceder, modificar, actualizar, mantener y verificar. Más específicamente, la presente invención se refiere a un método de selección de la cadena de consenso (el método de consenso de Gradiente de Idoneidad) que simplifica o elimina una etapa de prueba de trabajo que desperdicia y consume muchos recursos para determinar qué cadena o bloque debe seleccionarse entre las cadenas y los bloques que compiten para continuar usando la construcción de la cadena de bloques. Adicionalmente, la presente invención se refiere a determinados registros de datos distribuidos complejos que interoperan para proporcionar al libro mayor electrónico distribuido con ciertas capacidades especializadas.

### **ANTECEDENTES**

Las partes individuales pueden tener o mantener libros mayores electrónicos distribuidos o cadenas de bloques, o pueden contener entradas para diferentes partes y pueden replicarse y distribuirse entre varios participantes en una red. Los libros mayores electrónicos distribuidos se usan por varias aplicaciones, que incluyen el intercambio y el registro de transacciones en criptomonedas, tal como, por ejemplo, transacciones que involucran Bitcoin, Ethereum, Ripple (XPR) y otras criptomonedas. Además, las cadenas de bloques también pueden usarse para registrar y confirmar la identidad o existencia de una persona, objeto o evento. Los registros de datos que confirman tal identidad o existencia pueden luego usarse para tal aplicación como confirmación de la identidad de un titular de cuenta, o para confirmar la relación entre dos titulares de cuenta, o para confirmar la ocurrencia de un evento en el mundo real, y para usar luego esa información para dirigir el comportamiento contractual.

Las cadenas de bloques también pueden usarse como una solución para registrar, rastrear y mediar la propiedad de un activo o múltiples activos, este último codificado como un token (o múltiples tokens) en una cadena de bloques, que puede transferirse o de cualquier otra manera manipularse a través de elementos del libro mayor electrónico distribuido, tal como contratos inteligentes y registros de datos en cadena.

Además, las cadenas de bloques también pueden usarse para facilitar el comercio de divisas y el envío de dinero al servir como un registro de transacciones representadas en cadena mediante los tokens de activos respaldados por divisas. Por ejemplo, los tokens de activos respaldados por divisas pueden crearse por una institución de confianza (por ejemplo, un banco) que emite tales tokens y operaciones o contratos con otra institución (por ejemplo, un banco diferente) mediante el uso de contratos inteligentes en una cadena de bloques, para efectuar la transferencia de tokens de activos (y activos subyacentes) desde una entidad de terceros a otra. Tal transferencia puede condicionarse tras la verificación de que se han cumplido los requisitos que puedan establecerse en el contrato inteligente asociado para ese token de activo.

### DESCRIPCIÓN DE LA INVENCION

En vista de lo anterior, es un objeto y una característica de la presente invención proporcionar un sistema y método informático para la implementación de un libro mayor electrónico distribuido o cadena de bloques que usa un método de consenso de Gradiente de Idoneidad para determinar cuál de los nodos que compiten se selecciona para recibir una recompensa por un bloque particular o una cadena que se ha agregado a la cadena de bloques. Otro objeto y una característica de la presente invención es implementar un consenso de Gradiente de Idoneidad para determinar qué nodo se permite construir el siguiente bloque en el libro mayor electrónico distribuido o cadena de bloques, o qué bloque o cadena se usa como base sobre la cual continuar construyendo el libro mayor electrónico distribuido o cadena de bloques, y cómo resolver un conflicto entre cualquier bloque o cadena en caso de un conflicto entre bloques o cadenas alternativas que compiten.

Otro objeto y una característica de la presente invención es la implementación de una variante de consenso de Distancia Hash y/o una variante del consenso de Casillero del método de consenso de Gradiente de Idoneidad para determinar cuál de los nodos que compiten se selecciona para recibir una recompensa por un bloque o cadena que se agrega a la cadena de bloques. Estas y otras variantes del método de consenso de Gradiente de Idoneidad de la presente invención reducen o eliminan el método de consenso de Prueba de Trabajo menos eficiente mediante la reconceptualización del rol de resolución de conflictos y el proceso de consenso como un problema genérico de la evaluación de la Idoneidad entre bloques o cadenas que compiten. De acuerdo con al menos una modalidad, la variante de consenso de Distancia Hash puede usar un valor escalar de distancia hash como parte de su métrica de Idoneidad al comparar cadenas para la resolución de conflictos. La variante del

consenso de Casillero puede incorporar el acto de “congelar” tokens como un método complementario para generar una métrica de Idoneidad para usar en la comparación de cadenas.

Otro objeto y una característica de la presente invención es la implementación de un consenso de Gradiente de Idoneidad o una variante que incorpora la metodología de “confiar, pero verificar” para determinar cuál de los bloques o cadenas que compiten se agrega a la cadena de bloques o al libro mayor distribuido, y cuál de los nodos que compiten se selecciona para recibir una recompensa por el bloque o cadena que se agrega a la cadena de bloques o al libro mayor distribuido. Esta estrategia aumenta significativamente el rendimiento transaccional de los sistemas actualmente conocidos al reducir las restricciones de linealidad (y otras restricciones informáticas que retrasan el procesamiento general del sistema) y permite que el rendimiento de procesamiento del sistema se aproxime al rendimiento máximo teórico para cadenas de bloques y sistemas de libro mayor electrónico distribuido en general.

Otro objeto y una característica de la presente invención es el uso e implementación de los filtros de Bloom para representar todas las cuentas afectadas durante la ejecución de un bloque. También permite construir el nuevo bloque sin tener que ejecutar el bloque anterior para determinar qué cuentas pueden haberse actualizado en el transcurso de la ejecución de ese bloque.

Otro objeto más y una característica de la presente invención es el uso e implementación de los registros de Penalidad, y el uso de los filtros de Bloom para optimizar el procesamiento paralelo dentro de la cadena de bloques o la red de libro mayor electrónico distribuido.

Otro objeto más y una característica de la presente invención es la implementación de un método para estabilizar el suministro y el precio de los tokens de cadena de bloques mediante el uso de la información con respecto a los cargos del token nativo especificadas en los registros de datos agregados a los nuevos bloques. Además, la presente invención proporciona un mayor precio del token nativo y estabilidad de suministro al usar datos en cadena relacionados con las operaciones entre los pares de token nativos y definidos por el usuario como indicadores de inflación o deflación.

Otro objeto más de la presente invención es implementar un sistema y un método para la implementación y el mantenimiento de un sistema informático del libro mayor distribuido conectado a la red que incluye (1) una red interconectada de una pluralidad de computadoras, cada una incluye un procesador que ejecuta las instrucciones de computadoras almacenadas en una memoria electrónica de cada computadora para implementar y mantener un sistema de libro mayor electrónico distribuido implementado como una cadena de bloques vinculada hacia atrás de múltiples bloques de la cadena de bloques interconectados; (2) un primer nodo en tal red de computadoras, en donde

el procesador del primer nodo ejecuta el software peer-to-peer para crear un primer bloque competidor o un segmento de la cadena de bloques que se añadirá a la cadena de bloques; (3) un segundo nodo en tal red de computadoras, en donde el procesador del segundo nodo ejecuta un software peer-to-peer para crear un segundo bloque competidor o un segmento de la cadena de bloques que se añadirá a la cadena de bloques; (4) la red de computadoras (nodos) que ejecutan las instrucciones de computadoras que aplican un consenso de Gradiente de Idoneidad para calcular y aplicar un cálculo del valor de idoneidad más alto entre los primeros y segundos bloques o segmentos de la cadena de bloques que compiten; (5) los nodos que ejecutan el software informático de acuerdo con al menos una modalidad para determinar cuál de los nodos primero o segundo tiene permitido agregar un bloque siguiente o un segmento de la cadena de bloques en la cadena de bloques y cuál de los nodos primero o segundo tiene permitido compartir una recompensa por el bloque agregado o el segmento de la cadena de bloques; y (6) agregando el siguiente bloque o segmento de la cadena de bloques a la cadena de bloques después de la determinación.

Otro objeto y una característica de la presente invención es el uso de los diferentes tipos de registros, que incluyen los registros de génesis de token, los registros de transferencia, los registros de órdenes comerciales, los registros de liquidación, los registros de propuesta, los registros de determinación y los registros de vinculación de patrones. Estos diferentes registros, y la metodología de consenso de Gradiente de Idoneidad y sus variantes son útiles en (1) la implementación de sistemas de ejecución de contratos inteligentes orientados a eventos del mundo real que usan cadenas de bloques y sistemas basados en cadena de bloques; (2) el procesamiento de confirmación de identidad mediante el uso de cadenas de bloques y sistemas basados en cadena de bloques; (3) el seguimiento y título de activos de cadenas de bloques y sistemas basados en cadena de bloques; (4) los tokens de activos basados en monedas, procesos de comercio de divisas y remesas que usan cadenas de bloques y sistemas basados en cadena de bloques; y (5) la implementación de un sistema automatizado para la emisión, venta, transferencia y comercialización de tokens que pueden caracterizarse como valores, y que se regulan como valores de acuerdo con múltiples regulaciones de valores, reglas y restricciones que rigen las transacciones de valores.

Otro objeto más y una característica de la presente invención es el uso de los diferentes tipos de registros, que incluyen, sin limitación, los registros de génesis de token, los registros de transferencia, los registros de órdenes comerciales, los registros de liquidación, los registros de propuesta, los registros de determinación y/o los registros de vinculación de patrones para implementar los tokens

derivados y básicos, y controlar el suministro y el valor de los tokens, así como también la transferencia y el comercio de tokens.

Otro objeto y una característica de la presente invención es el uso de diferentes tipos de registros, que incluyen, sin limitación, los registros de génesis de tokens, los registros de transferencia, los registros de órdenes comerciales, los registros de liquidación, los registros de propuesta, los registros de determinación y/o los registros de vinculación de patrones para facilitar el intercambio de los diferentes tipos de tokens (por ejemplo, los tokens nativos y los diferentes tokens del usuario) y también la implementación de otras órdenes comerciales (por ejemplo, órdenes de mercado, órdenes limitadas y órdenes de limitación de pérdidas).

Estos y otros objetos, ventajas, aspectos y características de la presente invención son como se describió más abajo y/o apreciados y bien comprendidos por los expertos en la técnica.

#### DESCRIPCIÓN BREVE DE LOS DIBUJOS

Las características y los aspectos anteriores y otros de la presente invención serán más evidentes al leer la siguiente descripción detallada junto con los dibujos adjuntos, en los cuales:

La **Figura 1** ilustra la estructura general y la organización de varios componentes de una cadena de bloques en una red distribuida de acuerdo con al menos una modalidad de la presente invención.

La **Figura 2** ilustra la estructura general y la organización de una cadena de bloques de acuerdo con al menos una modalidad de la presente invención.

La **Figura 3** ilustra el flujo de proceso para la implementación del consenso de Casillero de un sistema y método de consenso de Gradiente de Idoneidad de acuerdo con al menos una modalidad de la presente invención.

Las **Figuras 4A y 4B** ilustran la estructura, organización e información del consenso de Casillero almacenado en una implementación del casillero de un sistema y método de consenso de Gradiente de Idoneidad de acuerdo con al menos una modalidad de la presente invención.

La **Figura 5** ilustra una determinación de consenso de Gradiente de Idoneidad entre dos cadenas de bloques que compiten A y B por el bloque N, de acuerdo con al menos una modalidad del sistema y método de consenso de Gradiente de Idoneidad de la presente invención.

La **Figura 6** ilustra la organización y los datos almacenados en un bloque de acuerdo con al menos una modalidad de la presente invención.

La **Figura 7** ilustra un diagrama de flujo y un algoritmo de negociación entre pares de tokens nativos y definidos por el usuario de acuerdo con al menos una modalidad que aumenta y/o garantiza el precio del token nativo y la estabilidad de suministro.

La **Figura 8** ilustra el procedimiento de determinación de la propuesta de acuerdo con al menos una modalidad.

La **Figura 9** ilustra el proceso de distribución de recompensas de la propuesta a cuentas de votación exitosa de acuerdo con al menos una modalidad.

La **Figura 10** ilustra el procesamiento de los registros de vinculación en una cadena de bloques de acuerdo con al menos una modalidad.

La **Figura 11** ilustra el procesamiento de los registros comerciales en una cadena de bloques de acuerdo con al menos una modalidad.

La **Figura 12A-D** ilustra un proceso simplificado de procesamiento de propagación de bloque mediante el uso de la variación “confiar, pero verificar” del consenso de Gradiente de Idoneidad mediante el uso de los filtros de Bloom de acuerdo con al menos una modalidad.

La **Figura 13A** ilustra una calificación por “Idoneidad” de los bloques con filtros de Bloom para un proceso simplificado mediante el uso de la variación “confiar, pero verificar” del consenso de Gradiente de Idoneidad de acuerdo con al menos una modalidad.

La **Figura 13B** ilustra la diferencia entre una cadena de bloques lineal y una implementación de gráfico acíclico dirigido de la variación de “confiar, pero verificar” del consenso de Gradiente de Idoneidad de acuerdo con al menos una modalidad.

La presente invención se describe ahora con referencia a los dibujos, en donde los numerales de referencia similares se usan para referirse a los elementos similares. En la siguiente descripción, los siguientes términos son útiles para comprender el uso y operación de un sistema o un componente de un sistema que incluye al menos una modalidad de la presente invención.

### **Estructura y Organización de la Cadena de bloques**

Una estructura de datos de la cadena de bloques es una estructura de datos inmutable de solo adjuntos, que comprende un conjunto ordenado de “**Bloques**” individuales, con cada bloque que representa un conjunto ordenado de registros de datos individuales. Un “**Libro mayor Electrónico Distribuido**” es una base de datos, duplicada en múltiples dispositivos, cuyo estado en cualquier momento se determina mediante la aplicación de transformaciones de datos discretas en un orden determinado y bien definido. Mediante el diseño, una estructura de datos de la cadena de bloques se usa típicamente para alojar un libro mayor electrónico distribuido, de manera que las

transformaciones de datos individuales del libro mayor electrónico distribuido corresponden a los registros de datos referenciados mediante los bloques de la estructura de datos de la cadena de bloques. Dentro del contexto de esta aplicación, el término “**Cadena de bloques**” se usa inclusivamente para referirse a varios tipos de estructura de datos sobre los cuales puede implementarse un libro mayor electrónico distribuido (que incluyen las estructuras de datos de la cadena de bloques lineales, estructuras de datos de malla n-dimensional, y gráficos acíclicos dirigidos) así como también al libro mayor electrónico distribuido en sí que se implementa sobre tal estructura de datos.

Un “**Sistema de Cadena de bloques**” es un sistema de dispositivos informáticos conectados entre sí a través de una red de intercambio de datos distribuida, dentro de la cual, en combinación con cualquier software asociado, puede implementarse una cadena de bloques y un libro mayor distribuido.

La **Figura 1** ilustra la estructura general de un sistema de la cadena de bloques usado en una red de intercambio de datos distribuido. Un sistema de la cadena de bloques 100 incluye típicamente un dispositivo de procesamiento de la cadena de bloques 110, un dispositivo de billetera 120, un dispositivo de navegación de datos de la cadena de bloques 130, y un dispositivo proveedor 140, todos los cuales se conectan a la Internet 190 (o una red distribuida). El sistema de la cadena de bloques puede incluir múltiples dispositivos de procesamiento de la cadena de bloques 110, múltiples dispositivos de billetera 120, múltiples dispositivos de navegación de datos de la cadena de bloques 130, y múltiples dispositivos proveedores 140, todos los cuales se conectan a la Internet 190 (o una red distribuida). Además de los dispositivos de procesamiento de la cadena de bloques separados 110, los dispositivos de billetera separados 120, los dispositivos de navegación de datos de la cadena de bloques separados 130, y los dispositivos de proveedores separados 140, un sistema de la cadena de bloques puede contener dispositivos de combinación 150 (no se muestra) que combinan características y funcionalidad de todos o alguna porción de estos dispositivos, o que de manera simultánea realizan la función de todos o alguna porción de estos dispositivos, y que también se conectan a la Internet 190 (o una red distribuida).

El dispositivo de procesamiento de la cadena de bloques 110 puede ser una computadora tal como un teléfono móvil, teléfono inteligente, tablet, computadora portátil, computadora de escritorio, computadora servidor, dispositivo informático diseñado específicamente, u otro tipo de dispositivo informático, con uno o más procesadores de computadora 112, memoria de computadora 114 para almacenar las instrucciones de computadoras, una base de datos 116 para procesar información de

la cadena de bloques (que incluyen registros y/o transacciones), y un módulo de comunicación 118 para conectarse a la Internet 190 y/o la red distribuida. El dispositivo de procesamiento de la cadena de bloques también puede incluir opcionalmente una pantalla 155 (no se muestra), y puede consistir en múltiples computadoras o una red de computadoras (ya sea directamente conectadas o distribuidas), todas del mismo tipo o de diferentes tipos. El dispositivo de billetera 120, el dispositivo de navegación de datos de la cadena de bloques 130, el dispositivo proveedor 140, y el dispositivo de combinación 150 tienen típicamente los mismos componentes que el dispositivo de procesamiento de la cadena de bloques 110.

El dispositivo de procesamiento de la cadena de bloques 110 funciona como un **“Nodo de Construcción de Bloques”**, que también se denomina “minero” en las cadenas de bloques de prueba de trabajo. Los Nodos de Construcción de Bloque son responsables de ensamblar nuevos bloques que reflejen la inclusión de nuevos registros o transacciones en la cadena de bloques, y de vincular esos bloques a la cadena de bloques. Los Nodos de Construcción de Bloque también son responsables de confirmar algorítmicamente si los bloques que se han vinculado a la cadena de bloques son válidos, y si los registros o las transacciones se incluyen válidamente en la cadena de bloques. Los Nodos de Construcción de Bloques también son responsables de propagar los bloques y los registros de datos dentro de la red. En al menos una modalidad de la presente invención, cada Nodo de Construcción de Bloques se asocia con una cuenta o dirección en la cadena de bloques, a la cual pueden asignarse las recompensas de minería de bloque en cuentas o direcciones. Tal cuenta o dirección también puede usarse por un Nodo de Construcción de Bloques para identificarse de manera segura y sus actividades dentro de la red y en la cadena a través del uso de firmas criptográficas.

El dispositivo de billetera 120 funciona como una **“Billetera”** que actúa para almacenar de forma segura claves criptográficas, cuyas claves se usan para firmar criptográficamente nuevos registros de datos que se proponen la inclusión en la cadena de bloques. Las firmas criptográficas garantizan que los Nodos de Construcción de Bloque solamente incluyan los registros de datos que se autoricen debidamente. Además de almacenar las claves criptográficas y otros datos seguros, las Billeteras pueden generar y firmar criptográficamente nuevos registros de datos y transmitirlos a uno o más Nodos de Construcción de Bloque, típicamente a través de la Internet 190 u otra red.

El dispositivo de navegación de datos de la cadena de bloques 130 funciona para proporcionar a los usuarios un medio para leer, ver o acceder de cualquier otra manera a los datos asociados con la cadena de bloques, de solo lectura.

El dispositivo proveedor 140 puede incluir una o más computadoras u otros dispositivos de procesamiento informático que facilitan la actividad de los Proveedores de Cadena de Bloques. El término “**Proveedor de Cadena de Bloques**”, en el contexto de la presente solicitud, se refiere a cualquier persona o entidad que ofrece, emite, vende o distribuye cualquier token a uno o más usuarios, o que proporciona servicios que de alguna manera se verifican, confirman, proporcionan o transmiten a través de una cadena de bloques, por ejemplo, servicios de verificación de identidad. Un dispositivo proveedor ejecuta un software que permite a los Proveedores de cadena de bloques proporcionar tales servicios.

El término “**Nodo**” en la presente solicitud se refiere a una computadora tal como un teléfono móvil, teléfono inteligente, tablet, computadora portátil, computadora de escritorio, computadora servidor, un dispositivo informático diseñado específicamente de otro dispositivo informático que ejecuta el software peer-to-peer de la cadena de bloques y se comunica con otras computadoras similares que operan en una red de intercambio de datos distribuidos conectada como la Internet.

El término “**Red de Nodo**” se refiere a una colección de computadoras que ejecutan el mismo software peer-to-peer de la cadena de bloques, que trabajan para construir una sola cadena de bloques compartida y se conectan entre sí a través de una red de intercambio de datos distribuida conectada como la Internet.

Los registros de datos aceptados para la inclusión en una cadena de bloques se almacenan o se referencian en los bloques de un libro mayor distribuido o Cadena de bloques. Los bloques individuales pueden contener datos de los registros y/o las transacciones. Alternativamente, los bloques pueden hacer referencia a tal dato a través de un hash criptográfico o resumen que resuma los datos, que el hash o resumen puede generarse mediante una estructura de datos separada que contiene los registros y las transacciones, por ejemplo, un árbol de Merkle. Para las cadenas de bloques de criptomonedas, la propiedad de las criptomonedas se vincula a las direcciones únicas o los números de cuenta incluidos como datos dentro de estos registros y transacciones. En tales cadenas de bloques de criptomonedas, el saldo de las criptomonedas asociado con una dirección o número de cuenta particular puede derivarse de todo el historial de los registros y las transacciones preservados por el libro mayor distribuido o la Cadena de bloques, a partir de su origen.

La estructura general de una estructura de datos de la cadena de bloques lineal 200 se ilustra con referencia a la **Figura 2**. La estructura de datos de la cadena de bloques 200 es una estructura de datos inmutable de solo adjuntos, que comprende uno o más bloques individuales 210A-C, que se muestran en la **Figura 2** como los Bloques 10, 11 y 12. Cada bloque 210A-C representa un conjunto

único de registros de datos 220A-C. Estos registros de datos de un bloque particular pueden no contenerse en ese bloque, sino que pueden representarse en el bloque por uno o más valores hash criptográficos, ya sea la raíz(es) de uno o más árboles Merkle 225A-D (que se muestran en la **Figura 2** para el hash Tx\_Root para el bloque 210B, o el hash de una estructura de datos agregada (tal como, el hash de algún mapa de valor clave). Cada hash criptográfico 230 es único para los datos de los que se deriva, y por diseño cambiará radicalmente incluso si se altera el aspecto más pequeño de sus datos subyacentes.

Cada bloque 210A-C en la estructura de datos de la cadena de bloques 200 puede describirse mediante, o en términos del hash criptográfico de sus contenidos 220A-C. El “**Hash de Bloque**”, en el contexto de la presente solicitud, es este hash criptográfico del contenido de un bloque en particular. Cada bloque se refiere a su bloque precedente en la cadena 200 mediante el hash de bloque de ese bloque precedente 240A-C, e incluye esa referencia entre sus propios datos (es decir, los datos que se usan para derivar su propio hash de bloque). En otras palabras, el bloque 210B almacena como sus propios datos el hash de bloque 240B del bloque precedente 210A. Cada bloque también puede almacenar la marca de tiempo 250 (mostrado para el bloque 210B en la **Figura 2**), que podría indicar la hora de la formación o aceptación del bloque en la cadena de bloques.

Esta organización asegura la inmutabilidad de la estructura de datos de la cadena de bloques 200. Dada la existencia de cualquier bloque en una estructura de datos de la cadena de bloques bien formada (por ejemplo, el último bloque de la cadena), la totalidad de la cadena precedente, que incluyen ambos bloques y todos los registros que representan, son inmutables. Cualquier cambio en cualquier registro o cualquier bloque tendría un efecto en cascada en los hashes criptográficos que constituyen la cadena, e invalidaría todos y cada uno de los bloques aguas abajo en su forma original. Por lo tanto, cualquier cambio en cualquier registro del último bloque 210N en la cadena 200 invalidaría todos los bloques aguas abajo (vinculados).

Un nuevo nodo que ingresa en la Red de Nodo debe validar la cadena, para asegurar que cada bloque represente con precisión los registros atribuidos a él, y que todos los bloques en la cadena de bloques se vinculen correctamente. Esta validación de reajuste se realiza en secuencia, desde el primer bloque 210A y su primer registro hasta el último bloque 210N y su último registro.

Además, cada nuevo bloque generado por cualquier Nodo(s) de Construcción de Bloques en la Red de Nodo debe validarse por otros Nodos de Construcción de Bloques, o no se agregará a la Cadena de Bloques. Además, debido a que el proceso de validación y el proceso de creación del nuevo bloque pueden realizarse en paralelo por muchos Nodos de Construcción de Bloques diferentes en

la misma red, cada Nodo también debe decidir qué instancia de la cadena de bloques debe usarse cuando existe más de una instancia válida del libro mayor o de la cadena de bloques en la red. Este es un proceso de decisión que suele denominarse como una “**Resolución de Conflictos**”. El término “**Algoritmo de Consenso**” se refiere al procedimiento determinado seguido por los Nodos de Construcción de Bloques de la red para decidir inequívocamente qué versión válida de la cadena de bloques es “canónica” y puede continuar usándose. Cada Nodo de Construcción de Bloques decide independientemente qué versión de la cadena de bloques es canónica, y procede a generar nuevos bloques con referencia a esa versión. Dado que todos los Nodos en la red por diseño ejecutan el mismo software, comparten todos los mismos datos y usan el mismo Algoritmo de Consenso, estas decisiones independientes resultan en que todos los Nodos acepten un único libro mayor distribuido o Cadena de bloques.

El “**Token**”, en el contexto de la presente solicitud, se refiere a una unidad de valor que se rastrea dentro de la cadena de bloques. Los tokens pertenecen a cuentas o direcciones particulares, y pueden transferirse de una cuenta o dirección a otra a través de la inclusión de un registro de transferencias en la cadena de bloques. Los tokens pueden crearse a través de la operación del Algoritmo de Consenso implementado dentro de un sistema de la cadena de bloques en particular - por ejemplo, como una recompensa que incentiva a los Nodos de Construcción de Bloque a contribuir con recursos informáticos a la operación de la red - o los tokens pueden generarse a través de ciertos tipos de registros, específicamente los “**Registros de Génesis**”.

Cada Nodo de Construcción de Bloques mantiene una copia duplicada de los valores de token asociados con cada cuenta o dirección dentro de la cadena de bloques. En cualquier momento, el estado actual de los tokens, cuentas o direcciones se codifica dentro de la cadena de bloques como la serie completa de registros que se han agregado a la cadena de bloques antes de ese momento. Por lo tanto, para determinar de manera confiable el saldo del token actual para cualquier cuenta o dirección determinada, un nodo ejecutará y validará todo el historial del libro mayor distribuido o la cadena de bloques, creando y transfiriendo los tokens entre las cuentas y direcciones hasta que se alcance el estado actual. Una vez validada la cadena completa, las actualizaciones se aplican al aceptar y validar, o al generar, nuevos bloques.

La “**Cuenta**”, en el contexto de la presente solicitud, es una dirección con la que se asocian determinados datos en la cadena de bloques. Los tokens, contratos inteligentes u otros datos pueden asociarse con una cuenta. El número de cuenta se relaciona con una o más claves públicas que se usan en combinación con una o más claves privadas para firmar criptográficamente los registros que

se crean en nombre de una cuenta, y que pueden alterar ciertos datos asociados con la cuenta. Solamente un registro que se firma criptográficamente con su clave de cuenta correspondiente se considerará válido e incluido en la cadena de bloques. En algunas modalidades, el número de cuenta puede ser un hash u otra transformación de una única clave pública.

La “**Transferencia**” o “**Transferencia por Token**”, en el contexto de esta solicitud, es un tipo de registro particular que reasigna un determinado valor del token desde la cuenta de firma a alguna cuenta de destino.

Los “**Cargos por Token**”, en el contexto de esta solicitud, se refiere al requisito de que un registro incluya un cargo, que es una característica común de las cadenas de bloques. En al menos una modalidad, la presente invención se ocupa de una mayor complejidad con respecto a los cargos, debido a que determinados registros vuelven a ejecutarse para una cantidad de bloques en sucesión. El “**Contrato Inteligente**”, en el contexto de esta solicitud, es un programa pequeño ejecutable que puede agregarse como un registro a la cadena de bloques. Existe una variedad de tiempos de ejecución de contratos inteligentes, y una variedad de lenguajes de nivel superior se compilan a tales tiempos de ejecución de contratos ejecutables.

El “**Segmento de la Cadena de Bloques**”, en el contexto de la presente solicitud, es una subsección discreta y contigua de una cadena de bloques. Un segmento de la cadena de bloques puede ser una porción de la cadena de bloques canónica actualmente aceptada como tal por los nodos dentro del sistema de la cadena de bloques, o puede representar una porción de una cadena de bloques en conflicto que difiere de la cadena de bloques canónica. Cada bloque dentro de un segmento de la cadena de bloques puede describirse como que tiene una “**Altura**” con relación al primer bloque dentro de la cadena de bloques como un todo, calculado al contar la distancia, en términos de número de bloques, entre el origen de las cadenas de bloques, y el bloque en cuestión. Cuando los segmentos de la cadena de bloques se comparan para la resolución de conflictos, típicamente ambos se originarían a la misma altura del bloque.

### **Visión General de los Algoritmos de Consenso de la Cadena de bloques**

Una característica que es común en la mayoría de las cadenas de bloques y los sistemas de la cadena de bloques es el uso de un algoritmo de consenso para resolver conflictos entre cadenas que compiten. Un sistema de la cadena de bloques implementará un algoritmo de consenso como una forma de que diferentes nodos no confiables lleguen a un acuerdo en cuanto al estado actual de la cadena de bloques en cuestión.

La presente invención, en al menos una modalidad, usa e implementa el Consenso de Gradiente de Idoneidad, y en otras modalidades usa e implementa ciertas variaciones del Consenso de Gradiente de Idoneidad, que incluyen el consenso de Distancia Hash y el consenso de Casillero. Estos métodos y sistemas de consenso reemplazan las metodologías de consenso conocidas, que incluyen la metodología de consenso original de “prueba de trabajo” de Nakamoto, y las metodologías de consenso conocidas de “prueba de participación”, dentro de un sistema de la cadena de bloques.

Una cadena de bloques típica comprende un conjunto ordenado de “bloques”, en donde el orden de estos bloques se incorpora en una cadena de referencias de cada bloque a su bloque precedente (o a múltiples bloques anteriores). Cada referencia posterior es un resumen o hash único generado criptográficamente del contenido del bloque precedente. Tales datos de referencia posterior se acumulan a los otros datos encapsulados en el bloque, que juntos se pasan a través de una función de resumen criptográfico o función hash para determinar la referencia única que contendrá el bloque subsecuente. Los bloques dentro de las cadenas de bloques que usan arreglos no lineales de bloques contendrán múltiples referencias posteriores.

La creación, secuenciación y vinculación de nuevos bloques, y el contenido de cada nuevo bloque, se determinan por computadoras vinculadas a la red (Nodos de Construcción de Bloques) a las que se les ha otorgado el derecho de tomar decisiones en nombre de toda la red, de acuerdo con un algoritmo de consenso. En un sistema de la cadena de bloques, es posible que varios nodos de construcción de bloques se trabajen de manera simultánea para crear nuevos bloques y agregarlos a la cadena de bloques, aumentando de esta manera la cadena de bloques y agregando nuevos registros de datos a ella.

Cada Nodo de Construcción de Bloques intenta agregar bloques que apuntan a la versión de la cadena de bloques que el algoritmo de consenso determina que es canónico, eligiendo entre las diferentes versiones que ha aprendido de otros nodos en la red. Los nodos de construcción de bloques típicamente abandonan los nuevos bloques que intentan construir después de confirmar que la versión de la cadena de bloques sobre la que se construyen se reemplaza por una nueva versión de la cadena de bloques. Luego, los nodos procederán a generar nuevos bloques que hagan referencia (es decir, basarse en) a la nueva versión de la cadena de bloques.

Por ejemplo, en una modalidad, si los nodos de construcción de bloques dentro de un sistema de la cadena de bloques se trabajan para construir un nuevo bloque en altura N, cuando uno de esos nodos genera satisfactoriamente un nuevo bloque válido en altura N y lo vincula a la cadena de bloques canónica, esta nueva cadena de bloques expandida puede sustituir a la versión anterior. En ese punto,

los nodos que se enteran de la nueva versión pueden comenzar a funcionar para construir el bloque subsecuente a la altura  $N + 1$ , que hará referencia al bloque a la altura  $N$ .

Sin embargo, los nodos que compiten pueden proponer de manera simultánea los bloques alternativos y las cadenas alternativas. Cuando dos o más versiones que compiten de la misma cadena de bloques compiten por la supremacía dentro de un solo sistema de la cadena de bloques, este hecho se conoce como “bifurcación”. Los conflictos entre diferentes versiones o segmentos de la cadena de bloques causados por una bifurcación pueden resolverse de manera determinista dentro de un sistema de la cadena de bloques, siempre y cuando los nodos dentro de la red se informen completamente sobre la naturaleza de cada bifurcación. Al recibir el mismo conjunto de opciones, todos los nodos dentro de un sistema de la cadena de bloques usarán el mismo algoritmo de consenso para elegir la misma bifurcación. Por esta razón, un sistema de la cadena de bloques permanecerá cohesivo siempre y cuando todos los nodos participantes implementen el mismo algoritmo de consenso. Cada nodo usará el algoritmo de consenso para determinar qué cadena alternativa termina con un bloque en altura  $N$  para referenciar cuando se construye un bloque en altura  $N + 1$ .

Dentro de cada bloque, los datos se representan como un conjunto de registros o transacciones. Los registros y las transacciones incluidos en los bloques codifican las transformaciones al estado global mantenido por el sistema de la cadena de bloques. Tales registros se ejecutan en una secuencia bien definida y predecible a medida que se generan, evalúan, validan y agregan nuevos bloques a la cadena de bloques por cada nodo de construcción de Bloques. Al acordar qué bloques y registros se incluyen en la cadena de bloques y cuáles no - qué acuerdo se habilita por el algoritmo de consenso de sistema - los nodos de construcción de bloques pueden garantizar que todo el sistema mantenga el mismo estado global.

Alguna porción de los registros o transacciones dentro de los bloques conmemora típicamente la creación o transferencia de los tokens. Los valores del token se reasignan desde determinadas direcciones de origen especificadas a determinadas direcciones de destino especificadas a través de los registros de datos individuales, cada uno de los cuales se firma criptográficamente para demostrar que cada transferencia se autoriza. Alternativamente, los tokens se agregan a cuentas o a direcciones según el algoritmo de consenso cuando se crea un bloque.

En las cadenas de bloques de las criptomonedas, un nodo de construcción de bloques que agrega satisfactoriamente un nuevo bloque a la cadena de bloques, recibirá típicamente una recompensa en forma de token. Para incentivar a los nodos a competir en la producción de nuevos bloques y validar las futuras actualizaciones a la cadena de bloques, cada nuevo bloque contiene un registro que asigna

una recompensa en forma de token a una dirección especificada por el nodo que crea el bloque. Incluir tales actos de incentivo para aumentar la participación en el proceso de construcción de bloques y validación de bloques, aumentando de esta manera la seguridad general del sistema.

Diferentes algoritmos de consenso difieren en cómo se determina y asigna esta recompensa. En un sistema de la cadena de bloques que usa un algoritmo de prueba de trabajo, por ejemplo, los mineros reciben recompensas en forma de token por haber resuelto ciertos problemas informáticos. Estos nodos de construcción de bloques tienen derecho a agregar nuevos bloques a la cadena de bloques después de resolver un problema informático antes de que cualquier otro nodo lo haya hecho.

### **Consenso de Gradiente de Idoneidad en Comparación con Estrategias de Consenso Conocidas**

El algoritmo de consenso de Gradiente de Idoneidad y las variaciones del consenso de Casillero y Distancia Hash de la presente invención, usadas con una o más modalidades, son distintas en un número de formas importantes a partir de las estrategias de consenso de prueba de concepto y prueba de participación conocidas que se usan para varias las cadenas de bloques existentes.

Una diferencia importante entre el método de consenso de Gradiente de Idoneidad de la presente invención, en comparación con las metodologías de consenso existentes y conocidas (que incluyen las metodologías de consenso de prueba de trabajo y de prueba de participación) es que en el sistema y método de Gradiente de Idoneidad, ningún bloque o nodo es privilegiado. En su lugar, todos los nodos activos pueden construir bloques de manera simultánea y transmitiendo los bloques que se construyen. Cada nodo es incentivado (pero no es necesario) a compartir cada bloque que genera, tan pronto como completa la construcción, para mejorar las posibilidades del nodo de recibir una recompensa por el nuevo bloque. Si ese bloque tiene la Idoneidad más alta entre los bloques que compiten, luego el uso de ese bloque se extenderá por toda la red; se desechará un bloque de Idoneidad más bajo.

Por el contrario, en los conocidos enfoques de consenso de la cadena de bloques que se usan en la actualidad, un nodo transmitirá un bloque solamente después de que el nodo haya alcanzado algún estado privilegiado - que es diferente de la metodología usada en el consenso de Gradiente de Idoneidad de al menos una modalidad de la presente invención. Entre los enfoques de consenso que se usan actualmente, diferentes enfoques de consenso asignarán a un Nodo de Construcción de Bloques su estado privilegiado a través de diferentes medios.

Otra diferencia entre el consenso de Gradiente de Idoneidad y las metodologías de consenso conocidas existentes es que los enfoques existentes se basan totalmente en la potencia informática o el poder criptográfico para asegurar los sistemas de la cadena de bloques. Por el contrario, el

consenso de Distancia Hash y las implementaciones del consenso de Casillero del consenso de Gradiente de Idoneidad, implementados de acuerdo con al menos una modalidad, calculan la Idoneidad a través de un proceso que combina tanto la potencia informática como el poder criptográfico (es decir, el control sobre los tokens en la cadena). Al usar una relación de los tokens enviados dentro de la cadena de bloques por distancia hash como una métrica de Idoneidad, los métodos de consenso de Gradiente de Idoneidad disminuyen o eliminan la posibilidad de que los nodos computacionalmente potentes ejerzan influencia concentrada sobre la producción de bloques. Cualquier nodo que intente ejercer influencia concentrada tendría que controlar un monto desproporcionado tanto de la potencia informática participante como de los tokens mantenidos en cadena. En una red ampliamente distribuida, con muchos nodos participantes, tal posibilidad sería difícil, si no imposible, para un solo actor lograrlo, tanto tecnológicamente como económicamente. Otra diferencia entre el consenso de Gradiente de Idoneidad de la presente invención y el consenso de prueba de trabajo conocido y los tipos de consenso de prueba de participación es la forma en que los nodos eligen qué bloque consultar cuando ensamblan nuevos bloques (es decir, qué bloque o segmento de la cadena de bloques construir), y por implicación en qué nodo se le otorga el derecho de incluir registros de datos en el bloque y asignar la recompensa en forma de token de ese bloque. El Gradiente de Idoneidad y los métodos de Distancia Hash/consenso de Casillero de la presente invención no usan algoritmos o métricas separados al decidir sobre qué bloque construir sobre una mano, y qué cadena elegir al resolver conflictos por otra. En su lugar, cada nuevo bloque potencial que se transmite a la red se evalúa como una cadena que compite con el mismo algoritmo que se usa para evaluar y comparar las bifurcaciones potenciales. Esta simplificación hace que la aplicación del consenso de Gradiente de Idoneidad sea mucho más eficiente para las aplicaciones de la cadena de bloques donde el tiempo de procesamiento es un problema clave. Además, reduce el costo de las transacciones que involucran las cadenas de bloques, el tiempo necesario para crear y verificar un nuevo bloque, y también promueve la participación de más nodos en el proceso de consenso/resolución de conflictos.

En al menos una modalidad de la presente invención, los nuevos bloques únicos se evalúan como una bifurcación o segmento de la cadena de bloques de altura uno, pero se evalúan de manera simultánea contra las bifurcaciones/segmentos de la cadena de bloques de altura arbitraria si todas las bifurcaciones se ramifican desde el mismo estado inicial.

En comparación, los algoritmos conocidos de prueba de trabajo y prueba de participación se ocupan de determinar qué nodo puede permitirse difundir el siguiente bloque, en función del bloque. En caso

de que dos nodos satisfagan los criterios para permitir que ambos generen el siguiente bloque en la cadena - o si se generan dos cadenas separadas como un resultado de una partición o bifurcación de la red - se usa un algoritmo separado para la resolución de conflictos para decidir qué cadena debe descartarse y qué cadena debe usarse para continuar construyendo.

El consenso de Gradiente de Idoneidad en general, así como también la Distancia Hash (que es un subtipo de consenso de Gradiente de Idoneidad en al menos una modalidad de la presente invención) y el consenso de Casillero (que es un subtipo de consenso de Distancia hash en al menos una modalidad de la presente invención) en particular, todos usan fórmulas para calcular un valor escalar bien definido (es decir, Idoneidad) en cada segmento de la cadena de bloques, para, comparar los bloques, o los segmentos de la cadena de bloques. Estas fórmulas determinan si un bloque o segmento de la cadena de bloques se selecciona sobre otro de manera no binaria (no es estrictamente una elección de validez sí/no). Puede ser válida una gran cantidad de bloques y cadenas, pero solamente la cadena con la mayor Idoneidad se elige por los nodos participantes.

Los nodos hacen crecer continuamente la cadena, agregando bloques y transmitiendo el valor de Idoneidad de sus cadenas a través de la red. Cuando un nodo se entera de la existencia de una cadena de mayor Idoneidad, solicita y recibe una actualización de la cadena del nodo en posesión de esa cadena, y luego realiza actividades de construcción de bloques a partir de esa cadena. En al menos una modalidad, un nodo puede participar en una nueva actividad de generación de nuevo bloque para una o un número de cadenas diferentes de manera simultánea, pero desechará las cadenas (o segmentos) de menor Idoneidad a favor de construir sobre la cadena de mayor Idoneidad una vez que pueda confirmar que la cadena de mayor Idoneidad es en realidad válida.

En los sistemas conocidos de consenso de prueba de trabajo, la determinación de si un bloque en particular es o no válido y calificado es una determinación binaria. Cada uno de los múltiples nodos “mineros” competirá para resolver el problema matemático que hará que su bloque sea válido. El primer nodo que genere un bloque válido transmite ese bloque, y de esta manera reivindica el derecho a especificar el siguiente bloque aceptado por la red (y para reivindicar la recompensa por el problema criptográfico resuelto y la creación/validación del nuevo bloque). Las bifurcaciones se minimizan porque la dificultad del problema matemático se calibra para garantizar que se resuelva aproximadamente una vez por bloque por un único nodo. Cuando el problema matemático se resuelve más de una vez y por más de un nodo, se trata como una excepción, cuya excepción se maneja por un algoritmo o conjunto de reglas por separado.

De manera similar, en los conocidos sistemas de prueba de participación, se usa algún algoritmo para asignar a un nodo particular el derecho a generar un bloque, y ese bloque, cuando se transmite a la red, se acepta o rechaza según esa determinación. La elección del nodo asignado dependerá de la “participación” que tenga ese nodo único (por ejemplo, cierta cantidad de propiedad de la criptomoneda, o la edad, antigüedad, o alguna otra participación o propiedad cuantificable que se vincula al nodo), posiblemente en combinación con algún factor aleatorio.

Por el contrario, en el consenso de gradiente de Idoneidad, implementado de acuerdo con al menos una modalidad de la presente invención, la mayoría de los nodos construyen, publican y transmiten bloques, y los bloques compiten cabeza a cabeza en base a la Idoneidad. Todos los bloques que eventualmente se incorporan en la cadena de bloques canónica final se habrán seleccionado al ganar un torneo de manera que los bloques generados por nodos en la red se compararán entre sí para determinar el bloque con la más alta Idoneidad. Por lo tanto, todos o la mayoría de los nodos en una red de consenso de Gradiente de Idoneidad transmiten sus bloques, como opuesto a los nodos que usan metodologías de consenso que solamente transmiten los bloques individuales que satisfacen criterios estrechos.

Las variaciones del consenso de Gradiente de Idoneidad compararán la idoneidad de toda la cadena, o la idoneidad de los segmentos de la cadena de bloques que contienen múltiples bloques, en lugar de solamente comparar los bloques individuales. En tales variaciones, la evaluación de los bloques se realiza mediante la evaluación de las cadenas, las bifurcaciones o los segmentos de la cadena de bloques.

En un consenso de Distancia Hash, que es una variante o especie particular del consenso de Gradiente de Idoneidad, como se usa en al menos una modalidad de la presente invención, la Idoneidad de un bloque o segmento de la cadena de bloques es una función de la distancia numérica entre el hash de cada bloque, y el “hash objetivo” de cada bloque. En una variación, la Idoneidad puede determinarse en base a minimizar la distancia hash, o en otra variación, al maximizar cierta relación de “tokens agregados usados” por la distancia hash. Esta distancia numérica se determina algorítmicamente según la implementación específica del consenso. Las variaciones del consenso de Distancia Hash, que incluyen el consenso de Casillero, pueden tener diferentes formas y metodologías para decidir sobre el hash objetivo y la distancia hash.

En al menos una modalidad del consenso de Distancia Hash, la salida de la función de la Idoneidad puede variar en dependencia de alguna agregación de los valores del token asignados dentro de los registros que se han incluido en los bloques o segmentos de la cadena de bloques que se comparan.

En otras modalidades, los sistemas de consenso relacionados pueden incorporar otras entradas al calcular la Idoneidad. Por ejemplo, una variación del consenso de Casillero de la presente invención (descrito en la presente descripción) combina la distancia hash con el valor del token agregado que se asigna a los “casilleros” que se usan para determinar el hash objetivo. En cualquier caso, estos valores del token pueden denominarse “tokens agregados usados” para un bloque o segmento de la cadena de bloques determinado.

En el consenso de Casillero, que es una variante o especie particular del consenso de Distancia Hash, implementado de acuerdo con al menos una modalidad, el hash objetivo se determina por los usuarios que envían tokens a los casilleros. El consenso de casillero divide el intervalo potencial de la salida hash en un conjunto continuo de casilleros, de manera que la salida del algoritmo hash relevante caiga solamente en un casillero. Cada casillero es un intervalo numérico, y cada registro que asigne los tokens a un casillero especificará el casillero al que contribuyen los tokens. Los tokens que se envían a los casilleros en un bloque pueden usarse para determinar el hash objetivo para un bloque futuro - es decir, para un bloque que aún no se ha generado un número fijo de bloques mayor que el bloque.

### **Comparación con el Consenso de Prueba de Trabajo de Nakamoto**

Dentro de la metodología y el sistema de consenso de la Prueba de Trabajo de Nakamoto, todos los nodos trabajan para crear los bloques, pero un nodo solamente transmitirá un bloque si gana una carrera para resolver un problema informático de la prueba de trabajo de fuerza bruta. Debido a la baja probabilidad de resolver tal problema, un número muy pequeño de nodos alcanzan el estado privilegiado requerido para crear, validar y publicar un bloque y agregar ese bloque a la cadena de bloques.

El consenso de Gradiente de Idoneidad de la presente invención en al menos una modalidad omitirá el problema informático de la prueba de trabajo. Un bloque es válido siempre y cuando esté correctamente construido en términos de formato, y siempre y cuando sus registros internos tengan integridad (es decir, el mismo valor del token no se transfiere dos veces; los valores de cuenta no se vuelven negativos; y todos los demás datos encontrados dentro del bloque son válidos).

Cuando se publica más de un bloque al mismo tiempo, o cuando emerge más de una subcadena dentro de la red, con el consenso de Prueba de Trabajo, la cadena más larga se acepta por todos y cada uno de los nodos. Mientras la longitud de la cadena teóricamente podría usarse como un criterio de Idoneidad en el consenso de Gradiente de Idoneidad, sería insuficiente por sí solo diferenciar entre dos bloques o segmentos de la cadena de bloques que compiten que compiten por la inclusión

en la cadena de bloques.

Debido a la ausencia de un problema de prueba de trabajo dentro del consenso de Gradiente de Idoneidad, un nodo malicioso que trata de maximizar la longitud de la cadena podría construir rápidamente bloque sobre bloque sin restricciones, sin pasar tiempo y dedicando recursos informáticos al trabajo necesario de ejecutar, procesamiento y validación de contratos inteligentes, registros y transacciones para su inclusión en los bloques. Como un resultado, los nodos cooperativos estarían en desventaja porque serían penalizados por intentar realmente evaluar la Idoneidad, mientras que el nodo malicioso sería recompensado por limitar sus recursos informáticos para su propio uso.

En las implementaciones del consenso de Prueba de Trabajo de Nakamoto, la posibilidad de las cadenas que compiten es relativamente poco frecuente, y los nodos comienzan a construir sobre el primer nuevo bloque válido que encuentran sin evaluar la idoneidad en absoluto, solamente la validez. La mayoría del tiempo, no hay cadenas que compiten para comparar.

Por el contrario, con el consenso de Gradiente de Idoneidad, antes de que comience la operación de construcción de los bloques, un nodo indagará de los nodos circundantes en cuanto a la métrica de Idoneidad de otras cadenas y descargará la cadena de Idoneidad más alta de entre las que están presentes. A continuación, luego comenzará a construir sobre esa cadena. Este mecanismo anima a cada nodo a buscar activamente un gran número de cadenas de bloques que compiten, clasificarlas por su idoneidad, elegir la de mayor Idoneidad y, luego, descargar y construir sobre esa cadena de bloques.

Uno de los beneficios clave de la presente invención es que debido a que la longitud de la cadena y la dificultad de la cadena se reemplazan por métricas de Idoneidad más eficientes en términos de energía, se elimina el gasto de energía y los recursos informáticos requeridos para verificar y establecer la longitud de la cadena y la dificultad, y se aumenta significativamente el rendimiento de la cadena de bloques.

Por el contrario, la metodología de consenso de la Prueba de Trabajo incluye el algoritmo de prueba de trabajo como una etapa adicional necesaria para permitir el uso de la longitud de la cadena como una métrica que puede usarse cuando se intenta resolver conflictos entre cadenas que compiten. Esta etapa adicional es informáticamente ineficiente y la energéticamente ineficaz, y aumenta innecesariamente el tiempo de procesamiento requerido para hacer crecer la cadena de bloques.

Las variaciones del consenso de Gradiente de Idoneidad de la presente invención son algoritmos simplificados que eliminan esta etapa innecesaria y desperdiciador. El consenso de distancia Hash

en al menos una modalidad determina un valor escalar que puede colocar un bloque dentro de un gradiente de valores de Idoneidad. Más significativamente, el consenso de Distancia Hash usa el valor escalar de distancia hash como parte de la métrica de Idoneidad al comparar las cadenas, mientras que la prueba de trabajo ignora enteramente las comparaciones del hash de bloque. al comparar las cadenas a través de la longitud del bloque, o usa solamente el valor de hash objetivo al comparar la dificultad agregada de dos cadenas, ignorando el hash de bloque.

### **Comparación con el Consenso de Prueba de Participación**

Hay una variedad de implementaciones de Prueba de participación (como, por ejemplo, Casper), pero la mayoría comparte algunos aspectos básicos. La Prueba de Participación depende del poder criptográfico, a diferencia de la Prueba de Trabajo que depende de la potencia informática.

Esencialmente, en el consenso de Prueba de Participación, se otorga el derecho a generar o proponer bloques a los nodos que ponen los tokens en participación, o que pueden reivindicar un conjunto de tokens en la cadena que prestan esa autoridad al nodo. En algunas implementaciones, hay cierta calidad inherente de los tokens referenciados que se usan para establecer qué nodo tiene autoridad de generación de bloques.

En otros, como Casper, los nodos generadores de bloques votan en tiempo real (no en cadena, sino como parte del protocolo de red) qué bloque aceptar. En otras palabras, el nodo generador de bloques tiene cierta autoridad específica del mundo real para aceptar o emitir una decisión sobre el nuevo bloque.

La Prueba de Participación es un enfoque para determinar qué nodo o nodos tienen permitido trabajar en la creación de nuevos bloques. Por el contrario, en el consenso de Gradiente de Idoneidad, cualquier nodo puede crear nuevos bloques, y se alienta a todos los nodos a trabajar en la creación de nuevos bloques. Además, los nodos de construcción de bloques en el consenso de Gradiente de Idoneidad no tienen ningún requisito o incentivo para contener, congelar o no asignar los tokens en absoluto, a diferencia de los nodos de construcción de bloques que operan dentro de muchos sistemas de Prueba de Participación. Cuando se toman en cuenta los tokens en el consenso de Gradiente de Idoneidad, es evaluar la actividad del token en la cadena generalmente a los fines de determinar la Idoneidad, en lugar de establecer si un nodo en particular tiene permitido o no publicar un bloque o votar en un bloque.

El consenso de Casillero de al menos una modalidad requiere que algunos tokens se “bloqueen” en la cadena durante un período, y las cuentas que bloquean estos tokens pueden recibir una recompensa por hacerlo. Sin embargo, no hay vinculación entre la participación de estas cuentas y la

determinación de qué nodo genera el bloque que se acepta (que es efectivamente aleatorio). En al menos algunas modalidades, tras la aceptación de un bloque, pueden otorgarse recompensas separadas tanto al Nodo de Construcción de Bloques como a las cuentas que congelan los tokens.

El consenso de Gradiente de Idoneidad de al menos una modalidad no tiene un protocolo de nominación/tipo de voto en el que participan los nodos autorizados, como se usa en determinados sistemas de consenso de Prueba de Participación. La convergencia se logra en el Gradiente de Idoneidad porque todos los nodos coinciden en los criterios que se usan para evaluar la Idoneidad. El consenso de Gradiente de Idoneidad generalmente y el consenso de Distancia Hash en particular no necesariamente requieren en todas las modalidades que los tokens se involucren como parte del algoritmo de consenso; por ejemplo, el valor de Distancia Hash solo puede usarse para determinar la Idoneidad. Cuando la Distancia Hash usa los tokens, como indicó anteriormente, puede usar solo el historial de los tokens en cadena, no los tokens identificados o asociados de alguna manera con el nodo de construcción de bloques.

El consenso de casillero, así como también otras variaciones del consenso de Distancia Hash de al menos una modalidad, se basan en una combinación de la potencia informática (componente de distancia-hash) y poder criptográfico (componente de valor del token), a diferencia de solamente el poder criptográfico para el consenso.

### **Definiciones Relacionadas con el Consenso y las Variaciones del Gradiente de Idoneidad**

El término “**Idoneidad**”, en el contexto de esta solicitud, se refiere a una puntuación numérica que se deriva del estado de una instancia de la cadena de bloques particular, o del estado de un segmento de la cadena de bloques particular. La presente invención usa e implementa una o más versiones del algoritmo “**Consenso de Gradiente de Idoneidad**” para determinar una puntuación de Idoneidad. En virtud del Consenso de Gradiente de Idoneidad, de acuerdo con al menos una modalidad de la presente invención, el segmento de la cadena de bloques que tiene el valor de Idoneidad más alto se seleccionará como la base sobre la cual continuar con la construcción de la cadena de bloques. En otras palabras, será seleccionado por los nodos (o Nodos de Construcción de Bloques) en la Red de Nodo como el segmento de la cadena de bloques correcto en caso de un conflicto entre diferentes segmentos presentes en la red. Los diferentes algoritmos de Consenso de Gradiente de Idoneidad y sus enfoques distintivos para resolver conflictos entre las diferentes cadenas de bloques y los segmentos de cadenas de bloques que compiten, se describen en más detalle más abajo.

El “**Hash Objetivo**”, en el contexto de esta solicitud, es un valor numérico perteneciente a un bloque a una altura particular, cuyo valor numérico se conoce antes de la creación de esos bloques y se

deriva de algún estado preexistente. El hash objetivo puede compararse con el hash de bloque. del bloque en cuestión para calcular la distancia hash de esos bloques. La “**Distancia Hash**”, en el contexto de esta solicitud, es el valor absoluto de la diferencia entre el hash de bloque de esos bloques y su hash objetivo. La distancia hash es un componente clave del cálculo de la idoneidad dentro del Consenso de Distancia Hash.

En al menos una modalidad de la presente invención, una variación del consenso de Gradiente de Idoneidad, denominada “**Consenso de Distancia Hash**”, calcularía la Idoneidad como alguna función de la Distancia Hash. Por ejemplo, una versión calcularía el Idoneidad de la siguiente manera:

$$\sum_{i=G}^N c \frac{V_{tokens \text{ únicos}}}{k + j \cdot D_{distancia hash}} y$$

donde V es el valor agregado de ciertos tokens calificados dentro de un bloque, D es la distancia hash para ese bloque y c, k, j e y son constantes, para todos los bloques con alturas G a N dentro del segmento.

Los “**Tokens Únicos**”, en el contexto de esta solicitud, se refieren a los valores del token que no se ha transferido, reusado o reasignado dentro de un segmento de la cadena de bloques particular; los tokens que no se han reciclado entre las diferentes cuentas y, por lo tanto, no se miden dos veces cuando se usan para los valores del token agregado para calcular la Idoneidad de una cadena particular. La medición de los tokens únicos requiere rastrear el origen de los tokens de cuenta a cuenta hasta el primer bloque de un segmento de la cadena de bloques que se evalúa. En al menos una modalidad del Consenso de Distancia Hash denominada “**Consenso de Casillero**”, la Idoneidad se calcularía en la misma manera que el Consenso de Distancia Hash, pero donde V incluye los tokens asignados a los casilleros a través de los registros de asignación de casilleros, y la distancia hash D se determina mediante el uso de un valor especificado con un casillero determinado como el hash objetivo. Por ejemplo, en una variación del consenso de Casillero, el hash objetivo es el punto medio del casillero al que se ha asignado el más grande a través de los registros de asignación de casilleros para el bloque en cuestión.

El “**Casillero**”, en el contexto de esta solicitud, se refiere a un intervalo numérico; uno de un conjunto de intervalos numéricos contiguos que se superpone con el intervalo de salida de una función hash

criptográfica. Por ejemplo, para cualquier entrada, la salida hash SHA- 256 será un número entre cero y  $2^{256}$ ; el casillero más bajo tendrá cero como su límite inferior, y el casillero más alto tendrá  $2^{256}$  como su límite superior.

El “**Registro de Asignación de Casilleros**”, en el contexto de esta solicitud, se refiere a un tipo particular de registro en la cadena de bloques que vincula un determinado valor del token con un casillero conceptual particular y una altura del bloque particular. La cuenta originaria que firma el registro de asignación de casilleros con su clave criptográfica puede poner este valor del token en riesgo de pérdida total o parcial a cambio de una oportunidad de ganar una mayor recompensa en forma de token en el futuro, algún número del bloques que sigue al bloque a la altura especificada. Todos los registros de asignación de casilleros pueden competir por una cantidad limitada de ranuras, y ganarían la inclusión en la cadena de bloques (y por lo tanto ganarían una oportunidad de poner en riesgo los tokens) en base al tamaño del cargo que incluyen. En al menos una modalidad, las ofertas para las ranuras se realizan en un tipo de una subasta Holandesa. Dado que se acepta el precio de oferta de compensación más bajo para todas las ofertas, se acumula un excedente de cargos no usados para los postores más altos. En una modalidad, el excedente de cargos puede devolverse, pero en otra modalidad, vuelve a aplicarse en los bloques futuros para persistir el registro más allá de su tenencia inicial. Además, los registros de larga duración pueden volver a ejecutarse para algunos bloques después del bloque donde se incluyen por primera vez. El cargo, de acuerdo con al menos una modalidad, se establece para el recubrimiento de este conjunto preciso y consistente de bloques, después de lo cual el registro necesitaría agregarse nuevamente.

### **Proceso y Método del Consenso de Gradiente de Idoneidad**

El método de consenso de la cadena de bloques de Gradiente de Idoneidad de la presente invención puede implementar las siguientes etapas, elementos y características (aunque no todas las etapas, elementos y características son necesarios en cada modalidad).

(1) El consenso de Gradiente de Idoneidad se implementa dentro de un sistema de la cadena de bloques y una red que consiste de los Nodos de Construcción de Bloques. Cuando un nuevo Nodo de Construcción de Bloques se conecta en línea, se conecta a la red al emparejarse con otros Nodos de Construcción de Bloques ya conectados a la red.

(2) Un Nodo de Construcción de Bloques recientemente conectado descarga los bloques de la cadena de bloques, encabezados de bloques, contratos inteligentes, registros de datos y transacciones de nodos pares dentro de la red de nodo. Descarga todos los bloques, encabezados de bloque, contratos inteligentes, registros y transacciones que se han agregado a la cadena de bloques hasta la altura

corriente de la cadena de bloques  $N - 1$ , y luego ejecuta contratos inteligentes y valida y procesa los registros y transacciones en secuencia, construyendo su propia representación interna del estado global.

(3) El Nodo de Construcción de Bloques comienza a recibir nuevos contratos inteligentes, registros y transacciones de nodos pares, así como también de billeteras y otros participantes de la red. El Nodo de Construcción de Bloques los agrega a un casillero de contratos inteligentes, registros o transacciones no procesados.

(4) El Nodo de Construcción de Bloques comienza a recibir información con respecto a los bloques recién contruidos que tienen la altura  $N$  generada por otros Nodos de Construcción de Bloques, y que se comparten dentro de la red a través de un protocolo gossip.

(5) El Nodo de Construcción de Bloques evalúa la idoneidad de los bloques recién contruidos en altura  $N$  de los que tiene conocimiento, para identificar qué bloques tienen el mayor valor de Idoneidad de acuerdo con los criterios de cualquier fórmula o algoritmo de idoneidad que esté en uso por el sistema de la cadena de bloques en su conjunto.

(6) El Nodo de Construcción de Bloques elige el bloque(s) de idoneidad más alto sobre el cual construir y comienza a construir un nuevo bloque en altura  $N + 1$ , ensamblando contratos inteligentes, registros y transacciones del casillero. El estado global se actualiza mediante la ejecución de contratos inteligentes y la validación y el procesamiento de los registros y las transacciones contenidos en los nuevos bloques.

(7) Otros nodos en la red continúan transmitiendo los bloques y los encabezados de bloque a medida que los crean. Los bloques y los encabezados de bloque se comparten tan rápido como se construyen, con poca retraso. Algunos de estos bloques estarán en la altura  $N$ , y otros estarán en la altura  $N + 1$ , y algunos de estos podrían estar en alturas aún más altas. En al menos una modalidad, el Nodo de Construcción de Bloques recibirá estos bloques y comenzará a analizarlos en un hilo, proceso de sistema o corutina separado, aparte del hilo, proceso de sistema o corutina que construye el nuevo bloque.

(8) Si el Nodo de Construcción de Bloques tiene conocimiento de un bloque en altura  $N$  que es de mayor idoneidad que el bloque sobre el que se construye actualmente, puede desechar el trabajo que ha hecho hasta el momento y comenzar a construir sobre ese bloque, en dependencia del tamaño de la discrepancia.

(9) Después de construir un nuevo bloque a la altura  $N + 1$ , el Nodo de Construcción de Bloques calculará la idoneidad de ese nuevo bloque y lo comparará con la idoneidad de otros bloques

candidatos a la misma altura que se transmiten a través de la red. Si el Nodo de Construcción de Bloques determina que el nuevo bloque que produjo tiene una idoneidad mayor que los otros bloques de los que ha tomado conocimiento, o si el nuevo bloque tiene al menos una idoneidad que supera un determinado umbral, luego el nodo transmitirá ese bloque a la red.

(10) El Nodo de Construcción de Bloques regresa a la etapa (3) anterior para comenzar el proceso de nuevo. En efecto, el nodo recibe nuevos registros y nuevos bloques, eligiendo el bloque de mayor idoneidad del que tiene conocimiento a la altura  $N$  más 1 ( $N + 1$ ), y comienza a construir un nuevo bloque a la altura  $N$  más 2 ( $N + 2$ ). Alternativamente, en al menos una modalidad, si se tiene conocimiento de un bloque ya construido a una altura más alta, por ejemplo a la altura del bloque  $N$  más 2, o  $N$  más 3, - y si el segmento de la cadena de bloques al que pertenece tiene una mayor Idoneidad agregada que los otros nodos, luego el Nodo de Construcción de Bloques puede saltarse adelante y comenzar a construir un nuevo bloque a una altura más alta, - por ejemplo, a la altura del bloque  $N$  más 3, o  $N$  más 4.

En al menos una modalidad de la presente invención, cuando se evalúa la idoneidad de un bloque determinado, puede evaluarse de manera individual, o puede evaluarse para el segmento de la cadena de bloques o toda la cadena de bloques a la que pertenece el bloque. Cuando se compara la idoneidad de dos bloques, ya sea la idoneidad de toda la cadena de bloques a la que pertenecen los bloques, o la idoneidad de dos segmentos de la cadena de bloques que incorporan el bloque de antepasado más reciente que comparten ambos bloques.

En al menos una modalidad, la idoneidad calculada para un bloque particular puede agregarse al encabezado del bloque que se comparte entre los nodos de red, junto con la idoneidad agregada de ciertos segmentos de la cadena de bloques de longitud predeterminada a la que pertenece el bloque, de manera que una indicación de la idoneidad del bloque puede ser conocida por los nodos antes de que se descargue un bloque completo, para acelerar la evaluación de la idoneidad por los Nodos de Construcción de Bloques.

### **Variante de Consenso de Distancia Hash del Consenso de Gradiente de Idoneidad**

Una variación de al menos una modalidad del algoritmo y método del consenso de Gradiente de Idoneidad de la presente invención es el algoritmo y método del Consenso de Distancia Hash.

La variante del consenso de Distancia Hash del consenso de Gradiente de Idoneidad decide el valor de Idoneidad de un segmento de la cadena de bloques de acuerdo con una función de la Distancia Hash de cada bloque dentro del segmento de la cadena de bloques que se considera. La Idoneidad de un bloque individual dentro de un sistema de Consenso de Distancia Hash puede determinarse al

calcular la idoneidad de un segmento de la cadena de bloques de tamaño uno que contiene ese bloque único.

La fórmula de Idoneidad de la Distancia Hash, de acuerdo con al menos una modalidad, puede describirse de la siguiente manera:

$$F(H, V) = \sum_{i=g}^n c \frac{V_i}{k + j \cdot H_i}^y$$

H: valores de distancia hash para todos los bloques en el segmento que se evalúa

$H_i$ : valor de distancia hash para el bloque  $i$

V: “Uso del Token Agregado” para todos los bloques en el segmento que se evalúa

$V_i$ : “Uso del Token Agregado” para el bloque  $i$ , para los tokens únicos dentro del segmento de la cadena de bloques que se evalúa.

$g$ : la altura del primer bloque dentro del segmento de la cadena de bloques que se evalúa

$n$ : la altura del último bloque dentro del segmento de la cadena de bloques que se evalúa

$c, k, y$ : constantes sintonizables

$j$ : constante sintonizable, probablemente la relación 1 sobre el máximo posible del valor de hash de bloque

Las modalidades alternativas del consenso de Distancia Hash pueden variar la manera en que se generan las entradas para esta función, o pueden modificar esta función con otra función que acepta valores de distancia hash como criterios de entrada.

### **Variante del consenso de Casillero del Consenso de Distancia Hash**

La implementación del consenso de Casillero de un sistema y método del consenso de Distancia Hash de acuerdo con al menos una modalidad de la presente invención se muestra con referencia a las **Figuras 3, 4A-B y 5**. Tales acciones descritas por estas figuras se realizan por un Nodo de Construcción de Bloques o minero que ejecuta una modalidad del algoritmo de consenso de Casillero.

Con referencia a la **Figura 3**, en 310, dentro del bloque numerado  $N$  (el bloque corriente), la salida de recompensa del bloque numerado  $N$  menos 1 (es decir,  $N - 1$ ) se asigna a direcciones ganadoras entre los registros del casillero asociados con el bloque numerado  $N$  menos  $G$  (donde  $G$  es un valor entero constante predefinido), así como también al Nodo de Construcción de Bloques o minero que generó el bloque numerado  $N$  menos 1. En 320, dentro del bloque numerado, se ensamblan las

transacciones N y los registros asignados a ese bloque y se ejecutan los contratos inteligentes dentro de ese bloque.

Luego, en 330, varios nodos, aparte del Nodo de Construcción de Bloques, transmiten los registros de asignación de casilleros a la red. Estos registros envían los tokens a las direcciones del casillero para el bloque numerado N. En 335, alguna porción de estos registros de asignación de casilleros se incorpora en el bloque numerado N, donde el número de registros de asignación de casilleros incorporados en el bloque es menor o igual a un conteo predefinido de límite superior X. En 340, la recompensa en forma de token para el bloque numerado N se calcula mediante el nodo de construcción de bloques. La naturaleza de esta recompensa depende de la naturaleza o la variación o modalidad; por ejemplo, puede incorporar alguna porción de los tokens enviados previamente a los casilleros, o cargos de transacción o de registro, o los nuevos tokens generados como parte de un algoritmo de suministro de token inflacionario. El valor de la recompensa en forma de token se almacena en una dirección de retención temporal para el bloque, para que se asigne al nodo de construcción de bloques y al(los) contribuyente(s) del casillero más adelante en el proceso. Luego, en 345, se finaliza el bloque N, y la referencia al bloque anterior (hash criptográfico del bloque anterior) se almacena dentro del bloque N. El hash también se calcula para el bloque N, y el bloque se propaga a la red de cadena de bloques para la evaluación de la Idoneidad.

En 350, otros nodos de construcción de bloques dentro de la red de nodo evalúan el nuevo bloque y lo comparan con los bloques que compiten en base a los criterios de la Idoneidad. Dado que los nodos implementan el mismo algoritmo de consenso, si todos los nodos comparten acceso a la información y al contenido de los diferentes bloques que compiten, todos elegirán independientemente y acordarán el mismo bloque, lo que sea que tenga una Idoneidad óptima. En 355, los nodos de evaluación determinan si el bloque propuesto o el segmento de la cadena de bloques maximiza la Idoneidad y si el nuevo bloque numerado N (o el segmento de la cadena de bloques que contiene este bloque) tiene el valor de Idoneidad más alto entre los bloques que compiten (o los segmentos de la cadena de bloques).

Si es así, los nodos de evaluación aceptan un nuevo bloque para la cadena de bloques y comienzan a construir los nuevos bloques para el bloque recién aceptado en 360. De cualquier otra manera, en la etapa 370, se determina que el nuevo bloque no es el bloque ganador (en base a la evaluación de la Idoneidad) y se desecha en lugar de un bloque que compite con un valor de Idoneidad más alto. Por lo tanto, en lugar de construir sobre el nuevo bloque, en 380, los billetes de evaluación comienzan a construirse a partir de un bloque alternativo que compite y que tiene un valor de Idoneidad más

alto. Luego, se repite el proceso de construcción y evaluación del bloque para el siguiente bloque con el número  $N + 1$  ( $N + 1$ ), a partir de 310.

De acuerdo con al menos una modalidad, los nodos de construcción de bloques incorporan por convención las asignaciones del token del casillero de mayor valor que encuentran. Por convención, solamente se incorporan al bloque el número  $X$  de registros de asignación de casilleros que asignan la mayor cantidad de tokens a los casilleros. Sin embargo, diferentes nodos de construcción de bloques dentro de la misma red pueden priorizar los registros de asignación de manera diferente, siempre y cuando no se incluyan más de  $X$  registros de asignación de casilleros en cualquier bloque. Los valores de las apuestas de mayor valor mejoran la posibilidad de que el bloque propuesto se adopte ampliamente y finalmente se acepte.

La evaluación de la Idoneidad de los bloques o segmentos de la cadena de bloques que compiten - y la selección del bloque o segmento de la cadena de bloques “ganador” - puede implementarse en al menos una modalidad de acuerdo con los siguientes criterios y método. En la etapa (1), el valor agregado del casillero para todo el bloque o segmento de la cadena de bloques se calcula incorporando los valores de todos los casilleros dentro de todos los bloques del segmento (esto es los “tokens agregados usados” para ese segmento de la cadena de bloques). En la etapa (2), la “distancia hash agregada” para todo el bloque o segmento de la cadena de bloques se calcula adicionando los valores individuales de la distancia hash para cada bloque dentro del segmento (es decir, la distancia entre el hash de bloque. y el casillero ganador para cada bloque, todos juntos). En la etapa (3), los “tokens agregados usados” se dividen por la “distancia hash agregada” para generar el valor de Idoneidad del segmento. Se prefieren mayores valores de Idoneidad.

Alternativamente, la evaluación de la Idoneidad de los bloques o segmentos de bloques también puede implementarse en al menos una modalidad como la siguiente. En la etapa (1), se calcula un valor agregado del casillero para cada bloque individual (en otras palabras, el valor de los “tokens agregados usados” primero se calcularía por separado para cada bloque en el segmento). En la etapa (2), se calcula un valor de Idoneidad individual para cada bloque dividiendo el valor agregado del casillero de los bloques por la distancia hash de ese bloque individual. En la etapa (3), los valores de Idoneidad de los bloques individuales se suman en un solo valor de Idoneidad para el segmento de la cadena de bloques. En otras palabras, para cada casillero ganador, divida el valor del casillero por la distancia hash, luego agregue todos los cocientes resultantes. Como en el enfoque anterior, se favorecerán los valores de idoneidad más grandes.

Cualquiera de las dos variaciones precedentes puede complementar el cálculo del valor de los

“tokens agregados usados” incorporando el valor del uso de los tokens elegibles dentro del segmento de la cadena de bloques. Por ejemplo, las transferencias y los cargos de token pagadas dentro del segmento se incorporarían además del valor de los tokens asignados a los casilleros, de manera que los segmentos de la cadena de bloques que contienen registros que mueven más tokens tienen una Idoneidad más alta. Las transferencias o gastos de token cíclicos se desconocían por completo: dentro del segmento de la cadena de bloques, cualquier utilización de tokens que se considere no puede incluirse en el total si se transmiten de nuevo después de una primera utilización; si el gasto se envía a una dirección, cualquier gasto de esa dirección se reduce posteriormente por el monto recibido, y cualquier monto con descuento recibido no se cuenta para la dirección del destinatario.

Las **Figuras 4A-B** ilustran la estructura del consenso de Casillero, la organización y la información almacenada en una implementación del casillero de un sistema y método del consenso de Gradiente de Idoneidad de acuerdo con al menos una modalidad de la presente invención.

Cada casillero representa un intervalo de valores potencialmente conformes al valor de hash de bloque. que aún no se ha determinado para un bloque futuro numerado N. En otras palabras, cada casillero contiene un hash objetivo potencial para el bloque numerado N. Tal hash objetivo para un casillero puede determinarse de varias maneras en dependencia de la modalidad: puede ser el punto medio del rango de valores representado por el casillero, puede ser el valor mínimo o máximo representado por el casillero, o por alguna otra función que se implementa de manera consistente en todos los nodos. Una asignación de tokens a un casillero en particular es un voto a favor de ese hash objetivo.

Con referencia a las **Figuras 4A-B**, los usuarios que compiten (es decir, los titulares de tokens) apostan al hash objetivo de futuros bloques al enviar los tokens a los casilleros individuales. Los tokens enviados al casillero en el bloque numerado N menos G (bloque  $N - G$ ) que esté más cerca del valor de hash de bloque. final del bloque número N ganan una recompensa en forma de token calculada para el bloque número N.

Los usuarios envían los tokens a los casilleros mediante la generación de registros de asignación de casilleros (especificando el índice del casillero, número del bloque, valor del token, etc.), cuyos registros se firman con la clave privada de la cuenta o dirección que contribuye al valor del token. Esos registros son nodos de construcción de bloques de difusión, de manera que estos registros pueden incluirse en un bloque que se acepta en la cadena de bloques. Los nodos de construcción de bloques difundirán ciertos registros de asignación de casilleros propuestos y los incluirán en los nuevos bloques, al menos en una modalidad, clasificando los registros propuestos y seleccionando

el número X de registros que asignan el valor del token más alto a los casilleros para un bloque en particular. Los nodos desecharán los registros de asignación de bloque propuestos que no se incorporaron satisfactoriamente en bloques anteriores, y solamente intentarán incorporar transacciones de registros de asignación de bloque para el bloque actual.

Con referencia a las **Figuras 4A-B**, la Asignación A 410, que se registra como parte del bloque N – G (bloque numerado N menos G), “gana” (o se le asigna) una recompensa en forma de token para el bloque numerado N sobre la Asignación B 420 porque se ha comprometido con el casillero con el valor más cercano al hash final del bloque numerado N. Tal recompensa en forma de token puede incluir los tokens asignados a otros casilleros que no son “ganadores” (es decir, que no son los casilleros del bloque N - G que son los más cercanos al hash de bloque N).

En al menos una modalidad, los casilleros pueden representarse como direcciones, de manera que las asignaciones de los tokens pueden realizarse mediante la transferencia de tokens (o al señalar) a tales direcciones.

Los casilleros pueden representarse como direcciones 440, de manera que las asignaciones de tokens pueden realizarse transfiriendo fichas a tales direcciones. En el caso mostrado, una cadena de bloques usa hashes de bloque de ocho bytes, pero puede tener hashes que son más largos o cortocircuitados de acuerdo con otras modalidades. En el caso de esta modalidad, los primeros cuatro bytes pueden usarse como un valor para describir cada casillero, limitando de esta manera la cantidad de casilleros disponibles, de manera que haya menos casilleros que valores de hash disponibles. En modalidades alternativas, puede usarse ya sea más o menos de la mitad de la longitud del hash para definir casilleros, hasta el tamaño del hash de bloque completo.

En el ejemplo que se muestra en las **Figuras 4A-4B**, pueden definirse más de dos mil millones de casilleros, pero en modalidades alternativas, pueden definirse más o menos casilleros. Con tantos casilleros, el casillero más cercano con cualquier valor puede no ser el casillero inmediatamente más cercano a un hash; el casillero elegido será el valor de almacenamiento del casillero más cercano.

En al menos una modalidad, la cadena de bloques puede usar palabras de ocho bytes para representar direcciones a o desde las cuales pueden enviarse tokens. Las direcciones dentro de esta cadena de bloques pueden distinguirse entre sí de acuerdo con su uso. Los casilleros se representan mediante direcciones que se distinguen de las direcciones de contrato o de cuenta de usuario.

Con referencia a la **Figura 4A**, una manera de distinguir tales direcciones entre sí es usar un prefijo. Por ejemplo, el primer byte 462 de una dirección puede usarse para codificar el tipo de dirección de “asignación de casilleros”. Si los últimos cuatro bytes 468 de la dirección se usan para codificar el

casillero (por ejemplo, índice de casillero), luego los tres bytes restantes 465 pueden usarse para codificar el número del bloque al que pertenece la apuesta. En tal configuración, se podrían usar seis bytes para representar el número del bloque al que se dirige el registro de asignación de casilleros, lo que permite que se haga referencia a cualquiera de las ~16.8 millones de generaciones de bloques. En otras modalidades, la longitud de la dirección puede ser mayor, lo que permite que se contabilicen y usen más números de bloque.

Alternativamente, como se ilustra en la **Figura 4B**, puede designarse una dirección de asignación de casilleros fija 470 como destino de toda transacción de asignación de casilleros, con el número del bloque 480 y el índice del casillero 490 como campos separados dentro del registro de asignación de casilleros, junto con el monto del token y la cuenta de origen del token o la dirección de origen del token. En otras modalidades, otros datos también pueden incluirse como campos separados del registro. Por ejemplo, el tipo de dirección y el número del bloque pueden incluirse como campos separados dentro de un registro de asignación de casilleros, de manera que la dirección del casillero en sí misma que comprende solamente el índice del casillero.

Un ejemplo de la determinación del consenso de Gradiente de Idoneidad entre dos cadenas de bloques A y B que compiten para el bloque de construcción N, de acuerdo con al menos una modalidad de la invención preestablecida se muestra en la **Figura 5**.

La decisión entre las cadenas que compiten puede manejarse mediante una variedad de algoritmos que tienen en cuenta los tokens asignados a los casilleros, así como también las distancias hash ganadoras, de acuerdo con la presente invención. En un ejemplo, un sistema que consiste de Nodos de Construcción de Bloques determina qué segmento de la cadena de bloques tiene el valor de Idoneidad más alto, calculado para ser  $\sum W / \sum \Delta$ , donde “W” representa el valor de los tokens agregados usados para un bloque determinado (es decir, la suma de todos los valores de asignación de casilleros y otro uso del token elegible dentro del bloque), y “ $\Delta$ ” representa la distancia hash para un bloque determinado.

Como se ilustra en la **Figura 5**, un nodo de construcción del Bloque numerado N debe decidir entre construir en la Versión A 510 o la Versión B 520 del bloque numerado N menos uno (Bloque N-1), cuyos bloques están en los extremos terminales de los segmentos de la cadena de bloques Versión A y Versión B. En otras palabras, estas dos versiones del bloque con el número N menos uno que compiten entre sí en la red, pero solamente uno se incluirá en la cadena de bloques que contiene el bloque N.

Para determinar el bloque correcto sobre el cual construir, el nodo debe calcular un valor de

Idoneidad para cada segmento de la cadena de bloques distinto al que se evalúa. En el ejemplo que se muestra en la **Figura 5**, el segmento de la cadena de bloques 510 de la Versión A tiene un peso =  $(5,000 + 2,000 + 5,000 + 1,200) / (100+100+200+100) = 13,200 / 400 = 33$ . La cadena 520 Versión B tiene un peso =  $(5,000 + 2,000 + 8,000 + 1,000) / (100+100+100+50) = 16,000 / 350 = 45.71$ . Por lo tanto, la cadena Versión B tiene un valor de Gradiente de Idoneidad más alto. Una vez que se informa a un nodo de construcción de bloques con respecto a ambas cadenas, construirá el bloque N en la cadena 520 de la versión B debido a su mayor valor de Idoneidad. La versión B del bloque numerado N - 1 se referencia por el nuevo bloque numerado N que se construye, de esta manera hace crecer la cadena.

Para cada bloque en la **Figura 5**,  $\Delta$  se determina al comparar el hash de bloque. de cada bloque con los registros de asignación de casilleros del bloque situado en el número G de bloques antes de evaluar el bloque. Dado que el valor de  $\Delta$  depende de tener conocimiento del hash de bloque. que se considera, no se calcula por el nodo de construcción de bloques que construye el bloque que describe, pero en cambio se calcula por el nodo que construye el siguiente bloque. El valor  $\Delta$  del bloque numerado N se incorporará en el bloque numerado N más 1, por lo que se desconoce el valor  $\Delta$  del Bloque N en la **Figura 5**.

La **Figura 6** ilustra la información que puede almacenarse en un bloque de acuerdo con al menos una modalidad de la presente invención. El bloque puede contener tal información como un Puntero al Bloque Padre o a un Hash de Bloque Padre 610, que conecta el bloque con su bloque precedente (es decir, padre). También puede incluir una Asignación de la Recompensa de la Apuesta en Bloque de los Padres 620, es decir, una asignación de recompensas en forma de token a las cuentas que hicieron asignaciones del casillero que se usaron para determinar la idoneidad del bloque anterior (es decir, padre). Puede contener un Cálculo  $\Delta$  Preservado para el Bloque Padre 630, que es el valor de la distancia hash calculado para el bloque precedente (es decir, padre). Las transacciones y contratos estándar 640 pueden colocarse en un bloque o incluirse indirectamente como un puntero o un hash a los datos actuales. El bloque también puede incluir Transacciones de Asignación de Casilleros 650, es decir, registros de asignación de casilleros que se usan para calcular la idoneidad de un bloque futuro. También puede incluir una Transacción de Recompensa para un Nodo de Construcción 660, que asigna la porción de la recompensa al nodo de construcción del bloque que generó el bloque, y un Marcador de Posición de Recompensa para los Ganadores 670, que almacenaría temporalmente la recompensa que se asignará en el siguiente bloque a los contribuyentes del casillero que ayudaron a determinar la Idoneidad de este bloque. Finalmente,

puede contener un Valor de Hash de Bloque. 680, que es un hash criptográfico de todos los demás elementos del bloque.

Los bloques se construyen para conformar una estructura interna que es verificable. Los datos dentro de un bloque deben pedirse de manera determinista, de manera que dos bloques que contengan los mismos datos siempre deben ser idénticos. Dentro de un bloque, las transacciones estándar y los contratos 640 pueden registrarse independientemente de los datos relevantes para la información del Algoritmo de Consenso. Esto significa que este algoritmo de consenso debería ser capaz de admitir una funcionalidad de aplicación variada que puede implementarse sobre él.

En dependencia de la implementación específica de la cadena de bloques, puede permitirse que cada nodo de construcción de bloques asigne alguna porción de la recompensa de bloque agregada a su propia dirección 660. La recompensa que se asignará a los ganadores de este bloque debe asignarse a una dirección de marcador de posición 670, debido al hecho de que el receptor es inconsciente hasta que se haya calculado el valor hash para todo el bloque. El bloque subsecuente hijo distribuirá esta recompensa.

### **Método de Consenso de Casillero Alternativo**

Otra versión del algoritmo y método de Consenso de Casillero de acuerdo con al menos una modalidad de la presente invención es una variante de manera que no hay ningún ganador único al que se le asignen recompensas de casilleros. Más bien, todas las cuentas que asignan tokens a los casilleros pueden potencialmente recibir recompensas.

En esta variación, el hash objetivo se determina por el casillero que ha recibido la asignación más alta de tokens entre todos los casilleros. Al igual que en la versión precedente del Consenso de Casillero descrito anteriormente, cada bloque permite hasta  $X$  número de registros de asignación de casilleros, cada registro que tiene un valor de asignación de token  $w$ . En este escenario, sin embargo, el casillero que tiene la asignación más alta de tokens agregada  $\sum w$  se designa como el casillero objetivo, lo que representa un valor de hash objetivo  $T$ .

En al menos una modalidad, los registros de asignación de casillero para el bloque a la altura  $N$  menos  $G$  (bloque  $N - G$ ) se usan cuando se forma el bloque a la altura  $N$  para determinar cuál es el valor de hash objetivo para ese bloque. Por ejemplo, el punto medio del casillero con el más alto valor de asignación de token en el bloque en la altura  $N$  menos  $G$  se convertiría en el hash objetivo para el bloque en altura  $N$ . Entre todos los potenciales nuevos bloques generados por los nodos de construcción de bloques, el bloque que tiene el valor hash más cercano a ese hash objetivo  $T$  es más probable que los nodos en la red lo usen para construir el siguiente bloque en altura  $N$  más uno.

En al menos una modalidad, la recompensa se distribuye proporcionalmente a todas las cuentas que contribuyeron con tokens a los casilleros de ese bloque. El nuevo bloque creado en la altura  $N$  más uno contendrá un registro o registros que distribuyan proporcionalmente las recompensas en forma de token para el bloque anterior en la altura  $N$  a aquellas cuentas que originaron los registros de asignación de casillero incluidos con el bloque en la altura  $N$  menos  $G$ . Cada cuenta recibirá una asignación proporcional de la recompensa en dependencia del tamaño de la asignación que hizo a los casilleros en ese bloque. Alguna porción de la recompensa en forma de token también puede asignarse a una cuenta designada por el nodo de red que ensambló/construyó el bloque aceptado  $N$ . En al menos una modalidad, la recompensa consistiría en las mismas recompensas que se incluirían en cualquier bloque dentro de una implementación del Consenso de Idoneidad, que incluye los cargos de token incluidos en varios registros y transacciones, cargos usados para pagar la ejecución del contrato inteligente, y cualquier nuevo token generado como parte de un algoritmo de suministro de token inflacionario. Sin embargo, los tokens que contribuyeron a los casilleros no se incluirían en la recompensa, pero se devolverán a las cuentas que originalmente contribuyeron.

A diferencia de la versión precedente del Consenso de Casillero descrito anteriormente, los usuarios en este escenario no se incentivan a asignar tokens a los casilleros en busca de obtener una gran recompensa, a riesgo de perder todos los tokens asignados. En su lugar, los usuarios reciben incentivos para bloquear los tokens en casilleros para el número  $G$  de bloques debido a la garantía de que recibirán una porción de la recompensa de bloque después de que haya finalizado el período de bloqueo. Al bloquear estos tokens, la cadena de bloques se protege, porque cualquier cadena que compite paralela o creada en un ataque de doble gasto tendrá que tener acceso a una mayor cantidad de tokens únicos que aquellos que contribuyeron a los bloques generados dentro del período de bloqueo.

### **Consenso de Distancia Hash Sin Casilleros**

En otra variación del consenso de distancia Hash de acuerdo con la presente invención, los usuarios no asignan los tokens a los casilleros. En cambio, los usuarios bloquean sus tokens por un período de tiempo a cambio de una porción de la recompensa.

A diferencia de la variación del consenso de Casillero del consenso de distancia hash descrita anteriormente, los tokens bloqueados no influyen en el hash objetivo en absoluto. Pero, de manera similar al método de consenso de Casillero, el valor agregado de los tokens bloqueados en esta modalidad contribuye al cálculo del valor de Idoneidad de un bloque o una subcadena. La Idoneidad se establece comparando el hash de los bloques individuales en una cadena propuesta con un hash

objetivo apropiado para cada bloque, y combinando ese valor de distancia hash a través de alguna función con los tokens totales asignados dentro de la cadena. El hash objetivo en cada caso provendría de una fuente en cadena; por ejemplo, el hash objetivo podría ser una transformación del hash de bloque ya calculado para el bloque que ocurre en cierta cantidad  $G$  de bloques antes del bloque actual.

Para ilustrar: dejemos que la asignación total de tokens,  $S$ , dentro de un segmento de la cadena de bloques sea igual a  $\sum s$ , la suma de todos los tokens nativos asignados dentro de cada bloque de ese segmento. Dejemos que la distancia hash total,  $D$ , dentro del segmento de la cadena de bloques sea igual a  $\sum d$ , la suma de las distancias individuales entre el hash de cada bloques y el hash objetivo para todos los bloques en ese segmento. Luego, el segmento de la cadena de bloques elegido para construir puede determinarse eligiendo la cadena con el valor  $S/D$  más grande (es decir, el valor más grande  $\sum s / \sum d$ ), que se consideraría el valor de Idoneidad para ese segmento.

En otra modalidad, el bloque elegido puede determinarse eligiendo la cadena con el valor más grande  $\sum (s/d)$ . En otras modalidades alternativas, el bloque elegido puede determinarse eligiendo la cadena con el valor más grande  $(\sum a / \sum (d+c)^b)^j$ , donde los valores  $a$ ,  $b$ ,  $c$  y  $j$  son constantes o se conocen de cualquier otra manera determinista en cada iteración en base al estado de la cadena, de una manera acordada por la red.

En otra variación, el Idoneidad de un segmento de la cadena de bloques se determina por una función  $f(s_1, s_2, s_3, s_4, \dots s_Y, s_Z, d_1, d_2, d_3, d_4, \dots d_Z, d_Z)$ , para los bloques 1 al  $Z$  dentro de ese segmento. La distancia hash  $d_i$  en esta variación, como en otras variaciones, es el valor absoluto de la diferencia entre el hash de bloque  $i$  que se evalúa y el hash objetivo. El hash objetivo se determina por algún algoritmo fijo capaz de determinar el objetivo antes de que se realice la evaluación de consenso. En al menos una modalidad, el hash objetivo para el bloque numerado  $N$  sería el hash criptográfico del hash de bloque ya calculado para el bloque numerado  $N$  menos  $G$ . En al menos otra modalidad, el hash objetivo del bloque numerado  $N$ ,  $t_N$ , sería el hash real del bloque numerado  $N$  menos  $G$  anterior, más algún desplazamiento predeterminable  $c$ , de acuerdo con la fórmula:

$$d_N = |h_N - h_{N-G}| + c$$

### **Consenso de Gradiente de Idoneidad Mediante el uso del Espacio y el Tiempo**

En otra variación del consenso de gradiente de Idoneidad de acuerdo con al menos una modalidad de la presente invención, los usuarios no asignan los tokens a los casilleros, y el uso de un valor de distancia hash es opcional. En su lugar, la idoneidad calculada para cada bloque depende principalmente del número de etapas informáticas realizadas por los registros de procesamiento

dentro del bloque (tiempo), la cantidad de datos o la cantidad de registros en el bloque (espacio) y/o el valor de los tokens únicos dentro del bloque.

La modalidad mediante el uso del método del Consenso de Gradiente de Idoneidad mediante el uso del tamaño y el tiempo, que se basa en el proceso general de consenso de Gradiente de Idoneidad de la presente invención, puede implementar las siguientes etapas, elementos y características (aunque no todas las etapas, elementos y características se requieren en cada modalidad). Primero, se recopilan determinadas métricas medibles con respecto a cada bloque:

(1) Se calcula cierto valor del “token único”  $V_i$  para cada bloque en la subcadena, cuyo valor es igual a cierta cantidad de tokens únicos asignados dentro de ese bloque, por ejemplo (a) los tokens asignados a los casilleros dentro de ese bloque; (b) los tokens incluidos como cargos para los registros agregados a ese bloque; (c) los tokens asignados de cualquier otra manera dentro del bloque. Aquí, como en otra parte en la presente descripción, “tokens únicos” se refiere a la “tokens agregados usados” de los tokens que no se han transferido más de una vez, reusados o reasignados dentro de un segmento de la cadena de bloques particular, es decir, los tokens que no se han reciclado entre las diferentes cuentas, y por lo tanto no se miden dos veces cuando los valores del token agregado se usan para calcular la idoneidad de una cadena particular.

(2) Se calcula algún valor del tamaño  $S_i$  para cada bloque de la subcadena, cuyo valor puede ser igual al número de registros que se han agregado al bloque (es decir, que se encapsulan dentro de su árbol Merkle u otra estructura de datos), o puede ser igual a la cantidad de memoria ocupada por el bloque, incluidos todos los registros que representa.

(3) Se determina algún valor de tiempo de ejecución  $T_i$  para cada bloque de la subcadena, cuyo valor representa la cantidad de etapas informáticas requeridas para completar la validación y evaluación del bloque, que incluye la ejecución de todos los contratos inteligentes invocados por los registros dentro del bloque. Este valor se determinaría mediante la validación real de la subcadena, en el proceso de ejecución de todos los contratos inteligentes invocados y el seguimiento de todas las etapas computacionales.

(4) Parte del valor de la distancia hash  $D_i$  se calcula para cada bloque en la subcadena, como está descrito en otra parte en la presente descripción.

Segundo, se define una función particular que acepta estos valores como parámetros — $F(V, S, T, H)$ , donde  $V$ ,  $S$ ,  $T$  y  $D$  son listas ordenadas del valor del token único agregado, el tamaño, el tiempo de ejecución y los valores de la distancia hash para cada uno de los bloques en la subcadena— cuya función representa la métrica de Idoneidad usada para determinar la Idoneidad para el uso en el

proceso del consenso de Gradiente de Idoneidad descrito en la presente descripción:

(1) En una variación de esta función, se deciden algunos parámetros escalares constantes — $s$ ,  $t$ ,  $d$ — cada uno de los cuales representa el peso relativo dado a cada uno de los valores dinámicos anteriores del bloque cuando se agregan para determinar el valor de la métrica de Idoneidad para la subcadena, de acuerdo con la siguiente fórmula:

$$F(V, S, T, H) = \sum_{i=G}^N V + s \cdot S_i + t \cdot T_i + d \cdot D_i$$

Si algún valor se debe descontar completamente, luego su constante acompañante se establecería en cero. Por ejemplo, si solamente se usan la cantidad del token único y el tamaño que se usa para calcular la idoneidad, luego los valores asignados a  $d$  y  $t$  serían cero.

(2) También podrían usarse un número de variaciones de esta función; algunos ejemplos de las fórmulas específicas que combinan todo lo anterior serían como sigue (donde, además de las variables y constantes descritas anteriormente,  $G$  es el índice del primer bloque en la subcadena,  $N$  es el índice del último bloque en la subcadena, y  $c$  y  $k$  constantes arbitrarias seleccionadas para optimizar el rendimiento del algoritmo de Gradiente de Idoneidad):

(i) 
$$F(V, S, T, H) = \sum_{i=G}^N \frac{k \cdot V_i}{c + d \cdot D_i + s \cdot S_i + t \cdot T_i}$$

(ii) 
$$F(V, S, T, H) = \sum_{i=G}^N \frac{d \cdot V_i}{c + D_i} + \frac{s \cdot V_i}{S_i} + \frac{t \cdot V_i}{T_i}$$

(iii) 
$$F(V, S, T, H) = \sum_{i=G}^N \frac{k \cdot V_i}{S_i \cdot (c + d \cdot D_i)}$$

(iv) 
$$F(V, S, T, H) = \sum_{i=G}^N \frac{k \cdot V_i}{T_i \cdot (c + d \cdot D_i)}$$

(v) 
$$F(V, S, T, H) = \sum_{i=G}^N \frac{k \cdot V_i}{c + s \cdot t \cdot S_i \cdot T_i}$$

### **Descontando el Uso de Tokens Duplicados; Identificación de Tokens Únicos**

En al menos una modalidad del consenso de Gradiente de Idoneidad o del consenso de Distancia Hash, el uso de tokens contados para determinar la “Utilización de Token Agregado” de los bloques y los segmentos de la cadena de bloques, o la actividad de asignación de tokens usada de cualquier otra manera como una entrada a un cálculo de idoneidad, no debe incluir los tokens reasignados o reusados dentro del período que se evalúa, pero debe incluir solamente **“Tokens Únicos”**.

Dos segmentos de la cadena de bloques o bifurcaciones que se comparan en el contexto del consenso de Gradiente de Idoneidad finalmente rastrean su origen a un punto de ramificación compartido en algún lugar dentro de su historial. El período de evaluación para la comparación solamente debe incluir los bloques incluidos después de ese punto de ramificación (o un número fijo de bloques que preceden a ese punto de ramificación) hasta el bloque más reciente que se evalúa.

El término **“período de evaluación”** se usa en la presente descripción para describir el conjunto de bloques dentro de cada uno de los dos segmentos que se comparan, cuyos bloques se vinculan de vuelta a través de referencias posteriores recursivas al punto de ramificación compartido, en combinación con el segmento de la cadena de bloques de tamaño P que precede e incluye el bloque en el punto de ramificación compartido (donde  $P \geq 0$ ).

Una vez que se asigna un valor del token a una nueva dirección o cuenta dentro del período de evaluación, luego en ese momento ese valor se incluye en el valor de “Uso del Token Agregado” que se calcula.

Sin embargo, ningún valor del token puede contarse dos veces dentro del mismo período de evaluación para una cadena en particular. Cuando se evalúa si una dirección o cuenta de origen contiene suficiente valor de token para permitir la asignación de sus tokens a otra dirección o cuenta, cualquier valor del token previamente asignado a esa dirección o cuenta dentro del período de evaluación se descartará totalmente de la contribución de esa asignación de token al valor del “Uso del Token Agregado”. Además, cualquier cesión que se origine de tal dirección o cuenta se limitará de manera similar en las asignaciones subsecuentes para fines de determinar el “Uso del Token Agregado” dentro del período de evaluación.

Sin esta limitación, un nodo de construcción de bloques puede construir un bloque con un valor de Idoneidad que es inflado artificialmente por los mismos tokens que se envían de ida y vuelta, o gastados y regastados en cargos, por cuentas controladas por el nodo de construcción de bloques.

### **Recompensas Otorgadas a los Participantes en el Consenso de Gradiente de Idoneidad**

En al menos una modalidad del consenso de Gradiente de Idoneidad de la presente invención, para cada bloque que se agrega a la cadena de bloques, se determina una recompensa en forma de token

de acuerdo con alguna función de la actividad de generación de registros dentro del bloque (asignaciones de token, ejecución de contrato, y cualquier otra actividad registrada dentro del bloque). Esta recompensa se asigna a una o más cuentas elegidas por el nodo de construcción de bloques que ha creado el bloque, así como también a una o más cuentas que pueden ser receptores de recompensas obligatorias de acuerdo con el Algoritmo de Consenso implementado dentro de la modalidad específica.

La recompensa de bloque puede tener una variedad de fuentes diferentes, que pueden incluir uno o todos los siguientes, en dependencia de la implementación o modalidad específica:

- Los tokens completamente nuevos que aumentan el valor total del token agregado disponible en la cadena a una velocidad determinada por bloque, que los tokens se generan como parte de un algoritmo inflacionario de suministro de tokens.
- El total de cargos individuales y premios adjuntos a los registros de datos incluidos en el bloque; los cargos por token y los premios adjuntos a los registros de datos incentivan la inclusión de esos registros de datos particulares en el bloque (es posible que los registros sin recompensas suficientes no se incluyan).
- La rescisión o revocación de los tokens que originalmente se otorgaron  $N$  menos la cantidad  $R$  de bloques antes del bloque actual (donde  $R \leq N$ ), incentivando el uso activo de los tokens.
- En al menos una modalidad del Consenso de Casillero, los valores del token asignados a las direcciones de asignación de casillero del bloque actual  $N$ ; o, alternativamente, los valores del token asignados a las direcciones de asignación de casillero del bloque  $N-G$  que se usan para calcular la distancia hash para el bloque  $N$ .

En dependencia de la implementación o modalidad específica del consenso de Gradiente de Idoneidad, esta recompensa puede asignarse en su totalidad o en parte a una cuenta asociada con el nodo de construcción de bloques que construyó el bloque, y/o puede asignarse total o parcialmente a otras cuentas que participan en el proceso de consenso.

En al menos una modalidad del Consenso de Casillero, una porción de la recompensa se asigna a aquellas cuentas que contribuyen con tokens a los casilleros que ayudan a determinar la idoneidad del nuevo bloque, como está descrito en otra parte en la presente descripción.

### **Variación “Confiar, pero Verificar” del Consenso de Gradiente de Idoneidad**

Otra variación del sistema de Consenso de Gradiente de Idoneidad de acuerdo con al menos una modalidad de la invención aborda las limitaciones de rendimiento transaccional de los sistemas

actuales conocidos y mejora el rendimiento transaccional del sistema de la cadena de bloques.

Uno de los obstáculos reconocidos que obstaculizan la adopción a gran escala de la tecnología de la cadena de bloques en una variedad de sectores es el “rendimiento transaccional” de tales sistemas. La red de tarjetas de crédito Visa informa que maneja aproximadamente 1,700 transacciones por segundo en promedio, con una capacidad de escalar a 24,000 transacciones por segundo. El sistema de tarjeta de crédito MasterCard ha registrado números similares.

La cadena de bloques Bitcoin, por el contrario, no puede superar las 10 transacciones por segundo, y Ethereum típicamente puede procesar hasta solamente 20 transacciones por segundo. Estas conocidas limitaciones de los sistemas de prueba de trabajo dificultan la adopción más amplia de estas cadenas de bloques en los sectores financieros y otros sectores y aplicaciones.

Una limitación importante de los sistemas de la cadena de bloques de prueba de trabajo conocidos (y de muchos sistemas de prueba de participación) es su linealidad. Debido a que la salida de cada transacción puede actuar como la posible entrada de cada transacción subsecuente, todas las operaciones deben ejecutarse secuencialmente y todos los registros deben ordenarse de manera determinista. Además, la cadena de bloques se organiza como una cadena vinculada - una secuencia lineal de nodos. Por lo tanto, solamente puede crearse un nuevo bloque una vez que se conocen los bloques precedentes.

Bajo la restricción de linealidad, el rendimiento máximo de cualquier cadena de bloques puede proyectarse teóricamente como el número de operaciones y registros que pueden procesarse en secuencia mediante un programa que se ejecuta en un solo hilo en una computadora. En dependencia del hardware y el software usados, esto puede ser cientos de miles o millones de operaciones por segundo - lo que puede traducirse en muchos miles de transacciones por segundo, en dependencia de la complejidad informático de las transacciones individuales.

Además, en realidad, el rendimiento real observado varía significativamente de este máximo teórico. Un número de razones diferentes explican esta diferencia y explican por qué las implementaciones conocidas de la cadena de bloques no pueden lograr este rendimiento máximo teórico.

En los sistemas conocidos de consenso de prueba de trabajo, la mayor parte de los recursos informáticos se dedican a resolver el problema de la prueba de trabajo. Calcular la prueba de trabajo crea un retraso, pero este retraso es tratado por estos sistemas como una característica en lugar de un defecto. La dificultad de la prueba de trabajo se dirige a un retraso de tiempo específico entre los bloques, y la dificultad se ajusta para sincronizarse con ese retraso de tiempo.

Los sistemas conocidos de consenso de prueba de participación, así como también los sistemas de consenso basados en la votación y la aprobación, se toman el tiempo para coordinar entre los nodos, a fin de decidir qué nodo se autoriza a crear un nuevo bloque. En la práctica, estos enfoques permiten un tiempo de creación de bloques mucho más rápido y un mayor rendimiento de las transacciones que los sistemas de prueba de trabajo, pero típicamente a expensas de requerir algún tipo de centralización.

Aun así, incluso en estos sistemas, la cantidad de tiempo necesario para establecer el consenso no es trivial, ya que requiere los nodos para sincronizar su estado. Además, la mayoría de estos sistemas generan cierto retraso entre los bloques (durante el consenso y la creación), para proporcionar suficiente tiempo para permitir la propagación de datos.

El retraso entre los bloques permite que los registros de datos y los bloques descubiertos se propaguen a través de la red, típicamente a través de un protocolo gossip. Entre estas dos categorías de datos, las transacciones constituirán la mayor proporción.

Cuando se crea y distribuye un nuevo bloque, los nodos dentro de la red validarán el bloque al ejecutar las transacciones que contiene. Una variación de “confiar, pero verificar” del consenso de Gradiente de Idoneidad de acuerdo con al menos una modalidad pretende eliminar o reducir varias de las restricciones conocidas de las implementaciones de la cadena de bloques conocidas, y permitir que el procesamiento del sistema de la cadena de bloques práctico se aproxime al rendimiento máximo teórico (discutido anteriormente).

Un ejemplo de la propagación del bloque de propagación mediante el uso de la variación “confiar, pero verificar” del consenso de Gradiente de Idoneidad de acuerdo con al menos una modalidad de la invención se describe más abajo con referencia a las **Figuras 12A-D** y **13A-B**.

En al menos una modalidad, la presente invención seguirá ejecutando las operaciones de las cadenas de bloques linealmente, pero usará y aprovechará los sistemas informáticos multiprocesador/multinúcleo/multicpu (o clústers) al realizar el procesamiento de validación en una implementación de software paralela y multihilo (o multiproceso). En al menos una modalidad de la presente invención, cada una de las siguientes actividades puede ocurrir de manera simultánea y en gran medida independiente de las otras actividades:

- (a) el hilo principal de construcción de bloques, proceso o corutina construye sobre el bloque de mayor Idoneidad conocido por el nodo de entre los bloques ya difundidos a la red;

- (b) un hilo, proceso o corutina continúa en base al mejor bloque anterior propio del nodo que se produjo por sí mismo (con validación y confirmación inherentes) en caso de que el hilo, proceso o corutina “ganador” no pueda validarse (es decir, es fraudulento);
- (c) uno o más hilos/procesos en paralelo se basan en el bloque secundario, o un bloque de descubrimiento tardío, como otra copia de seguridad del bloque de mayor Idoneidad;
- (d) uno o más hilos/procesos en paralelo adquieren de nodos pares todas las transacciones de bloques adquiridos externamente, y luego proceden a validar esos bloques.

Mediante el uso de esta metodología de consenso de Gradiente de Idoneidad de la presente invención, un nodo enfrenta un gran riesgo de que los esfuerzos de construcción de bloques se desperdicien en caso de que se adopte un bloque fraudulento y no verificable como la base sobre la cual comenzar a construir el siguiente bloque. Tal bloque fraudulento deberá desecharse, junto con cualquier trabajo realizado para construir sobre él.

Sin embargo, siempre que la incidencia de fraude sea baja, luego el rendimiento de la implementación de acuerdo con al menos una modalidad de la presente invención correspondería a la velocidad de los hilos/procesos que trabajan para construir el siguiente bloque. Incluso estar dentro de uno o dos órdenes de magnitud del rendimiento máximo teórico haría que la variación de consenso de Gradiente de Idoneidad de la presente invención, mediante el uso del método de propagación de datos de “confiar, pero verificar”, fuera más eficiente y de mejor rendimiento (con respecto al rendimiento) que las implementaciones existentes de la cadena de bloques.

### **Método de Propagación de Bloques Confiar, pero Verificar mediante el uso de Filtros de Bloom**

En al menos una modalidad, la propagación de datos “confiar, pero verificar” opera como sigue.

(1) Los registros de datos (tal como los registros de transferencia o los registros de órdenes comerciales) no se propagan ampliamente a través de la red antes de la creación de bloques. Antes de la creación del bloque, los registros de datos solamente se comparten con los pares directos, o con los pares de los pares a un pequeño número específico de saltos de distancia del nodo generador de registros de datos.

Cada nodo de construcción de bloques solamente crea los nuevos bloques mediante el uso de los registros de datos que se originan de su vecindad cercana. Los vecinos cercanos se empujan las transacciones entre sí inmediatamente tras la adquisición, pero a un cierto nivel de eliminación, la propagación se detiene.

Un nodo sabría cuántos saltos ha tomado un registro de datos en particular, porque el registro propagado incluiría esa información. Un nodo decidiría si reenviar o no el registro de datos en base a ese conocimiento, y si los datos se reenvían, el nodo receptor se proporcionaría con un número actualizado que reflejaría el salto adicional.

(2) Los bloques recientemente creados se propagan a través de la red a través del protocolo gossip, pero solamente se incluyen ciertos campos de núcleo esenciales con tales datos del bloque. Es una característica común entre las implementaciones de la cadena de bloques que la representación de un bloque en la memoria y como un mensaje transmisible (en otra parte denominado “encabezado del bloque”) excluye la mayor parte de los datos que pertenecen al bloque en concepto. En cualquier caso, se incluye suficiente información con cada bloque para evaluar su Idoneidad y realizar de cualquier otra manera el algoritmo, que incluye el “Filtro Bloom de la Cuenta” del bloque y el valor de Idoneidad calculado.

(3) A medida que los nodos tienen conocimiento de los bloques recién creados, se desechan los bloques de baja Idoneidad, mientras que los bloques de alta Idoneidad se retienen y se reenvían hacia adelante. La mayoría de los nodos convergerán en el mismo conjunto de bloques de alta Idoneidad. Debido al hecho de que los bloques en esta etapa no se validan (excepto por el creador original), existe la posibilidad de que algunos de estos bloques sean fraudulentos. En el contexto de la presente invención, un “**Bloque Fraudulento**” es un bloque que ha sido transmitido, compartido o difundido por un nodo de construcción de bloques a pesar de ser inválido de acuerdo con los estándares y el software del sistema de la cadena de bloques. Para mitigar el riesgo de construir sobre los bloques más fraudulentos, los nodos de construcción de bloques pueden retener y construir sobre más de un bloque para la redundancia.

(4) Inmediatamente después de que la distribución de bloques se haya estabilizado razonablemente - lo que significa que la velocidad de envío de los bloques de alta idoneidad por el cable ha disminuido por debajo de un determinado umbral, medido desde la perspectiva de un nodo individual - el nodo comenzará a construir el siguiente bloque sobre los bloques de alta idoneidad que ha retenido. Creará el bloque mediante el uso de los registros de datos que se han generado localmente, o que se han generado en su vecindario cercano (como se describió anteriormente).

Cuando un nodo crea este nuevo bloque, el nodo creará un filtro de Bloom, que representa todas las cuentas afectadas durante la ejecución del bloque (el “Filtro de Bloom de la Cuenta”). Alternativamente, pueden crearse dos o más filtros de Bloom, que representen las cuentas de origen y las cuentas de destino por separado. En al menos una modalidad, también pueden crearse uno o

más filtros de Bloom adicionales que representan otros datos afectados por el bloque que no se asocian con ninguna cuenta. El(los) filtro(s) de Bloom se incluirá(n) con el bloque y se distribuirá(n) como parte del mensaje enviado de “encabezado del bloque”.

Las cuentas incluidas en los Filtros de Bloom de la Cuenta son aquellas para las cuales se han cambiado algunos datos dentro del bloque con el que se asocian los filtros de Bloom. Si se encuentra una cuenta dentro del(los) Filtro(s) de Bloom de la Cuenta, luego es posible o probable que su estado se modifique si el nodo fuera a validar y ejecutar el bloque. Sin embargo, si no se encuentra una cuenta en el filtro de Bloom, luego se asegura de que el estado de esa cuenta no cambiaría por la validación del bloque y, por lo tanto, su estado actual ya sería conocido por el nodo, incluso si el nodo aún no ha validado o ejecutado el bloque. El filtro de Bloom, de acuerdo con al menos una modalidad de la invención, permite construir un nuevo bloque sin tener que validar y ejecutar el bloque anterior para actualizar los valores de la cuenta. Puede verificarse la membresía de las cuentas en los Filtros de Bloom como un contrato inteligente se ejecuta o cuando se agrega un registro de datos al bloque. Las cuentas afectadas se verifican para garantizar que cada cuenta no se contenga dentro del filtro de Bloom del bloque anterior. Si es así, la ejecución del contrato o la inclusión del registro se detiene y se desecha del bloque.

(5) Los filtros de Bloom usados pueden permitirse una tasa de falsos positivos relativamente alta, lo que les permite ser más pequeños y rápidos que en la mayoría de los casos de uso. Las transacciones que se desechan de un bloque en la mayoría de los casos pueden incluirse en un bloque subsecuente. Debido a que las direcciones se generan a través de hash criptográfico, su distribución ya se distribuye aleatoriamente en el intervalo de posibles valores de hash. Se ha demostrado que una operación de módulo simple es suficiente en tales casos para lograr una baja relación de falsos positivos, sin hashing adicional, como lo muestra Jianyuan Lu, y otros, en su artículo de 2015, “One-Hashing Bloom Filter”, publicado junto con el 23rd Simposio Internacional sobre Calidad de Servicio de la IEEE. Alternativamente, o en combinación, puede usarse un filtro de Bloom comprimido para reducir el tamaño del filtro para la transmisión. La compresión puede hacerse de numerosas maneras conocidas, que incluyen la descrita por Michael Mitzenmacher, de Harvard, en su artículo de 2002, “Compressed Bloom Filters”.

(6) De manera simultánea con la creación de nuevos bloques, en hilos paralelos separados, corutinas o procesos del sistema, los nodos comenzarán a recopilar y transmitir registros de datos que pertenecen a los bloques de alta Idoneidad sobre los que se construyen. Los registros de datos pueden aprenderse de varias maneras:

(a) Si un bloque se originó a partir de un nodo par directo de un nodo, luego el nodo ya tendrá esos registros de datos. Estos registros necesitarían ser pedidos y ensamblados, y se ejecutarían los contratos inteligentes, pero los datos no necesitarían ser retransmitidos.

Una vez que un nodo tenga todos los registros correspondientes a un bloque de Idoneidad alta, luego comenzará a distribuir esos registros a través del protocolo gossip. Otros nodos que reciben estos registros los retendrán y difundirán (o, alternativamente, los desecharán) en dependencia de si esos nodos también han retenido el mismo bloque.

(b) Si a un nodo todavía le falta una transacción, luego puede consultar por esa transacción comunicándose directamente con un nodo (ya sea un par directo o no) que afirma haber validado el bloque asociado.

(7) Tan pronto como se descargue el conjunto de registros de datos correspondientes a un bloque, puede comenzar el proceso de validación del bloque. En al menos una modalidad, esto ocurriría de manera simultánea con la creación de nuevos bloques. Si un bloque no puede validarse por cualquier motivo (por ejemplo, si la raíz de Merkle no puede reproducirse mediante el uso del conjunto de los registros de datos, o si el hash de bloque se especifica incorrectamente, o si los registros de datos violan alguna regla o son de cualquier otra manera inválidos), luego se desecha junto con el nuevo bloque que se construye sobre él.

(8) Para aumentar la posibilidad de crear el bloque de mayor Idoneidad, un nodo puede trabajar para construir los bloques alternativos en paralelo en uno o más hilos separados, corutina, procesos del sistema o clúster de computadoras. Un nodo transmitirá sus bloques de mayor Idoneidad a medida que se crean, desechando los bloques de menor Idoneidad.

El enfoque descrito anteriormente podría tener los siguientes impactos esperados en la práctica:

(a) Localización emergente - ciertos nodos del concentrador recopilan actividad de ciertas cuentas y otros nodos del concentrador recopilan actividad de otras cuentas; la responsabilidad para procesamiento de los registros relativos a diferentes conjuntos de cuentas se compartimenta de manera holgada e informal a través de la topología de red emergente.

(b) Diversificación emergente - los nodos excluirán cuentas de un bloque si aparecieron en el bloque anterior; la implicación es que, combinado con la localización, los bloques en la cadena de bloques alternarán en cuanto a la parte de la red de la que se originan.

(c) Difundir los registros según sea necesario - los registros solamente se propagan en toda la red si existe una alta posibilidad de que se conserven con la cadena de bloques de forma permanente.

El proceso de propagación de los bloques mediante el uso de la metodología de propagación de bloques “confiar, pero verificar” de acuerdo con al menos una modalidad de la presente invención se describe con referencia a las **Figuras 12A-D**.

En la Etapa 1, como se ilustra en la **Figura 12A**, se generan o envían diferentes registros de transferencias mediante las billeteras a varios nodos 1201-1206 (1-6) dentro de una red. Cada registro de transferencias transfiere los tokens de una cuenta a otra 1221-1226 (cuentas representadas por letras). Por ejemplo, el nodo 1201 (1) tiene un registro de transferencias 1221 que transfiere de la cuenta A a la C, el nodo 1204 (4) tiene un registro de transferencias 1224 que transfiere de las cuentas B a la A y de la B a la C, el nodo 1205 (5) tiene un registro de transferencias 1225 que transfiere de la cuenta L a la M y el nodo 1206 (6) tiene un registro de transferencias 1226 que transfiere de la cuenta X a la Y y de la Y a la Z. Los nodos 1201 (1) y 1203 (3) no tienen sus propios registros de transferencias. Los nodos de la relación de pares se muestran mediante líneas interconectadas.

En la etapa 2, que se ilustra en la **Figura 12B**, los nodos 1201-1206 (1-6) de la red comparten registros solamente con sus pares inmediatos. Los registros 1231-1236 que cada nodo tiene después de compartir con los pares inmediatos se muestran como 1231-1236 en la **Figura 12B**. Por ejemplo, el nodo 1201 (1) comparte registros con los nodos 1202 (2) y 1204 (4). Por lo tanto, contiene dentro de un conjunto combinado de registros de transferencias 1231 su propia información de registro de transferencias 1221 (que se muestra en la Figura 12A), para la transferencia desde la cuenta A a la C y la información del registro de transferencias del nodo vecino 1204 (4), para la transferencia desde la cuenta B a la A y de la B a la C. De manera similar, el conjunto combinado de registros de transferencias 1232 para el nodo 1202 (2) contiene información del registro de transferencias de los nodos vecinos 1201 (1) y 1206 (6). Por lo tanto, el nodo 1232 contiene la información del registro de transferencias para las transferencias desde la cuenta A a la C, de la X a la Y y de la Y a la Z. El nodo 1205 (5) contiene la información del registro de transferencias de los nodos vecinos 1204 (4) y 1206 (6), de esta manera, contiene dentro de su computadora la información de los registros de transferencias de memoria para las transferencias desde las cuentas L a la M (los datos originales del registro de transferencias 1225 para el nodo 1205 (5) y los datos del registro de transferencias para las transferencias desde las cuentas B a la A y de la B a la C (desde los datos originales del registro de transferencias 1224 del nodo vecino 1204 (4)) y los datos del registro de transferencias para las transferencias desde las cuentas X a la Y y de la Y a la Z (desde los datos originales del registro de transferencias 1226 del nodo vecino 1206 (6)). La misma metodología se aplica para determinar los datos del registro de transferencias 1231, 1232, 1234, 1235 y 1236 mantenidos dentro de la memoria

de la computadora por los nodos 1201-1206 (1-6). El nodo 1203 (3) no tiene sus propios datos originales del registro de transferencias y su único nodo vecino 1202 (3) tampoco tiene ningún dato original del registro de transferencia. Por lo tanto, no hay datos del registro de transferencias combinados que se mantengan en el nodo 1203, como se ilustra en la **Figura 12B**.

En la etapa 3, que se ilustra en la **Figura 12C**, los nodos 1201-1206 (1-6) construyen bloques que contienen los registros que conocen. Cada bloque tiene un filtro de Bloom que identifica las cuentas contenidas en el bloque 1241-1246, aquí mostrado como una lista de las cuentas (letras) afectadas por las cuentas que son parte de los registros de transferencia de bloque que se muestra en la **Figura 12B**. Por ejemplo, los datos del registro de transferencias del nodo 1201 (1) afectan las transferencias entre las cuentas A, B y C, lo que forma un bloque con el filtro de Bloom ABC (1241), mientras que el nodo 1202 (2) tiene datos de registro de transferencias que afectan las transferencias entre las cuentas A, B, C, X, Y y Z, lo que forma un bloque con el filtro de Bloom ACXYZ. De manera similar, el nodo 1206 (6) forma un bloque con filtro de bloom LMXYZ, el nodo 1205 (5) forma un bloque con filtro de bloom ABCLMXYZ y el bloque 1204 (4) forma un bloque con filtro de bloom ABCLM, como se ilustra en la **Figura 12C**. Simultáneamente, el nodo 1203 (3) genera, o se le envía un nuevo registro de transferencias, que se incluirá en el siguiente bloque; específicamente, una transferencia desde la cuenta Q a la cuenta R.

El uso de letras para representar las cuentas y los bloques es solamente para fines ilustrativos. En modalidades actuales y en otras, el filtro de Bloom no es una lista de letras, o una lista de cuentas, sino una estructura de datos estándar.

En la etapa 4, que se ilustra en la **Figura 12D**, los bloques (formados en la **Figura 12C**) son compartidos por todos los pares de nodo 1201-1206 (1-6) en toda la red. En la ilustración simplificada de la red en la **Figura 12D**, Todos los bloques se comparten con todos los nodos. Por lo tanto, todos los nodos de esta red incluyen la colección compartida de los bloques 1251 - ABCLMXYZ, ABCLM, ACXYZ, LMXYZ y ABC. En otras modalidades de la presente invención, los nodos solamente pueden compartir los bloques (con datos de transferencia de cuenta) si exceden un determinado umbral de valor de Idoneidad.

También en la etapa 4, el nodo 1203 (3) comparte su nuevo registro de transferencias con su único par, el nodo 1202 (2).

En la etapa 5, que se ilustra en la **Figura 13A**, los nodos en la red clasifican los bloques por Idoneidad, para decidir desde qué bloque comenzar a construir. En esta ilustración, se usa una métrica de Idoneidad simplificada que representa un “número de registros” con fines ilustrativos. En

otras modalidades, pueden usarse otras métricas para la métrica de Idoneidad. En el ejemplo ilustrativo de la **Figura 13A**, cada bloque con un filtro de Bloom se evalúa en base a una métrica de Idoneidad que involucra el monto de registros o cuentas incluidos.

Por ejemplo, el bloque con un filtro de Bloom ABCLMXYZ se evalúa en base a los registros de transferencia incluidos para las transferencias desde las cuentas L a la M, de la X a la Y, de la Y a la Z, de la B a la A. Según los criterios de evaluación de Idoneidad “cantidad de registros”, recibe una puntuación de Idoneidad 5. De manera similar, el bloque con un filtro de Bloom ABCLM se evalúa en base a los registros de transferencia incluidos para las transferencias desde las cuentas B a la A, de la B a la C, de la A a la C y de la L a la M, que recibe una puntuación de Idoneidad 4. Los bloques con filtros de Bloom ACXYZ, LMXYZ y ABC, respectivamente, se evalúan de manera similar y cada uno recibe una puntuación de Idoneidad 3.

Si el nodo 1202 (2) o el nodo 1203 (3) fuera a construir un nuevo bloque, crearían un bloque que contiene el registro de transferencias mostrado, transfiriendo los tokens de la cuenta Q a la cuenta R. Si tal bloque se construye en un sistema mediante el uso de una estructura de datos de la cadena de bloques lineal 1381 mostrada en la **Figura 13B**, luego el bloque con el filtro de Bloom ABCLMXYZ se usaría, porque tiene el valor de Idoneidad más alto de 5. Sin embargo, los filtros de Bloom ABC y LMXYZ no se solapan en absoluto, y cuando se combinan, los bloques con estos filtros de Bloom tienen un valor/puntuación de idoneidad mayor de 6. Por lo tanto, si el nuevo bloque se construye en un sistema mediante el uso de una estructura de datos de gráfico acíclico dirigido 1382 que se muestra en la Figura 13B, luego se usarían estos dos bloques.

### **Reducción de los Ataques de Denegación de Servicio que Propagan los Bloques Fraudulentos**

Como se describió anteriormente, un riesgo que enfrentan los nodos de construcción de bloques que implementan el método de propagación de bloques “confiar, pero verificar” es que pueden comenzar a construir un nuevo bloque encima de un bloque fraudulento, que necesita desechar el trabajo después de que se descubre el bloque fraudulento. Este riesgo para los nodos individuales puede escalar más allá de ser un riesgo localizado y puede convertirse en un riesgo en todo el sistema si los nodos maliciosos comienzan a transmitir bloques fraudulentos *en masa*.

Por lo tanto, un atacante que desee ralentizar o deshabilitar la red (es decir, usar el ataque de denegación de servicio conocido) podría intentar enviar una inundación de bloques fraudulentos, con la esperanza de abrumar a otros nodos con desperdicios de trabajo y ralentizar o deshabilitar la operación del sistema de la cadena de bloques. Si bien es poco probable que un ataque de este tipo subvirtiera datos de la cadena de bloques o provocara el robo de tokens (porque todos los bloques se

validan eventualmente), si se introducen demasiados bloques fraudulentos, el rendimiento de la red podría degradarse significativamente debido a los esfuerzos desperdiciados de los nodos construidos sobre bloques fraudulentos.

De acuerdo con al menos una modalidad, la presente invención puede usar varios enfoques para prevenir o como mínimo desincentivar este tipo de ataque y reducir el riesgo y la efectividad de tal ataque.

En primer lugar, los nodos pueden asignar una calificación de riesgo a otros nodos y pueden elegir construir solamente sobre un bloque sin validación si el bloque se ha originado a partir de un nodo de confianza. Los bloques de alta Idoneidad creados por nodos no confiables seguirían siendo construidos, pero solamente después de que se descarguen los registros de datos y se realice la verificación.

(1) En al menos una modalidad, la confianza se establece cuando los bloques recientes producidos por el nodo en cuestión ya se han validado satisfactoriamente. La confianza se perdería inmediatamente si un bloque creado por el nodo se considera fraudulento, pero se recuperaría si el nodo en cuestión produce una cantidad suficiente (potencialmente alta) de bloques válidos después de la detección de un bloque fraudulento.

Una variación de esta estrategia sería una forma de estrategia “tit-for-tat” similar a la descrita por Axelrod y Hamilton en su artículo de 1981 “The Evolution of Cooperation”. En al menos una modalidad de esta estrategia, un nodo que ha perdido la confianza en otro nodo permitiría que ese segundo nodo restableciera su confiabilidad mediante el simple acto de crear un bloque válido. Por el contrario, después de que se haya ganado la confianza, se perdería si el nodo en cuestión publica un único bloque fraudulento. Determinar el estado de confianza de un nodo dependería solamente del historial reciente, no del historial completo del nodo.

Incluso dentro de la misma red de cadena de bloques, diferentes nodos pueden emplear estrategias alternativas para determinar la confianza. No sería necesario que todos los nodos empleen la misma estrategia, ya sea “tit-for-tat” o alguna otra estrategia, ya que el comportamiento de la decisión interna del nodo con respecto a la confianza no necesitaría ser parte del protocolo global compartido por todos los nodos para lograr el consenso.

(2) Los nodos nuevos estarían automáticamente en un estado no confiable, por lo que tendrían que construir reputación para ser de confianza. Como mínimo, tendrían que hacer suficiente trabajo para generar uno o más bloques con una Idoneidad lo suficientemente alta como para que otros nodos corran el riesgo de construirse sobre ellos, según el umbral de riesgo configurado dentro de esos

otros nodos. Dado que un nuevo bloque se construirá antes sobre un bloque generado por un nodo de confianza (y por lo tanto será más probable que se incorpore permanentemente en la cadena de bloques), los nodos tienen un incentivo para ser confiables (en otras palabras, para satisfacer el umbral de riesgo de otros nodos de construcción de bloques).

(3) En al menos una modalidad, cada nodo de construcción de bloques se identificaría por una dirección de cuenta de la cadena de bloques incluida con cualquier bloque o segmento de la cadena de bloques que se transmite. Esta dirección de cuenta sería la dirección de cuenta de recompensa del nodo. Cada nodo firmaría criptográficamente cada bloque que produzca mediante el uso de la clave privada asociada con esa cuenta de recompensa, previniendo que los nodos no autorizados puedan dañar la reputación de otros nodos.

(4) En al menos una modalidad, los nodos muestrean aleatoriamente los bloques no confiables, eligiendo gastar una fracción relativamente pequeña de su esfuerzo informático para construir sobre tales bloques para construir confianza en los nuevos nodos que se unen a la red. Este proceso de muestreo ignoraría la Idoneidad de los bloques que se muestrean, para poder establecer más rápidamente si puede confiarse en un nuevo nodo.

(5) En al menos una modalidad, los nodos compartirían información de los nodos de confianza con los nodos pares. En otras palabras, cuando un nodo de construcción de bloques toma la determinación de que otro nodo es de confianza, compartiría esa determinación con los nodos pares. En una versión, el nodo de construcción de bloques generaría y firmaría criptográficamente un mensaje dedicado declarando que un nodo en particular es de confianza, y compartiría ese mensaje con sus nodos pares. En otra versión, el nodo generaría y firmaría criptográficamente sobres que encapsulan los bloques no validados de terceros que comparte con sus pares, pero solamente en el caso de que esos bloques no validados se originen con (es decir, son firmados criptográficamente por) los nodos de confianza. Los nodos pares extraerían información sobre el nodo de confianza de estos sobres, antes de manejar los bloques no validados.

### **Cargo de Construcción**

En al menos una modalidad, los bloques propuestos pueden contener alguna asignación de token nativo realizada por el nodo de construcción de bloques que propone el bloque, un incentivo para que otros nodos pasen por la molestia de descargar los registros del bloque y realizar la validación. Tal bloque que contiene tal incentivo se denomina en lo sucesivo “**Bloque Incentivado**”, y el incentivo en sí mismo se denomina “**Cargo de Construcción**”.

En esta modalidad, cada bloque incentivado contendría un cargo de construcción que se transferiría automáticamente a la cuenta de recompensa del nodo de construcción de bloques tras la validación del bloque subsecuente. La cuenta de origen para ese cargo sería la dirección de la cuenta proporcionada por el nodo que generó el bloque incentivado. Todos estos encabezados de bloque podrían incluirse luego como registros dentro de la cadena de bloques; si no se elige un bloque sobre el cual construir, los cargos de construcción para ese bloque incentivado se transferirían, no obstante, a la recompensa del bloque que incorpora los registros de encabezado del bloque para estos bloques descartados de baja Idoneidad.

De acuerdo con la presente invención, los encabezados de bloque de los bloques incentivados sobre los que no se elige construir podrían ubicarse o colocarse en dos lugares mutuamente exclusivos en la cadena de bloques: (1) incorporado en la cadena de bloques en sí mismo como bloques de “tío”, referenciados de manera similar a otros bloques dentro de la cadena de bloques, pero sin aportar registros de datos, transacciones, o cualquier modificación de estado a la cadena de bloques, o (2) incluidos entre los registros de datos asociados con bloques individuales, de tal manera que el bloque se conozca por la cadena de bloques y se haga referencia a él, pero sin que ninguno de los datos de ese bloque se incorpore de cualquier otra manera en el estado de la cadena de bloques. Referenciando estos bloques incentivados en un nuevo nodo de esta manera efectuaría la transferencia del cargo de construcción al nodo de construcción de bloques.

El tamaño o el monto del cargo de construcción no se establecería por adelantado, pero lo elegiría el nodo que transmita el bloque propuesto. Puede surgir que el cargo de construcción promedio es similar en tamaño al cargo de registro de datos promedio, porque de cualquier otra manera no pagaría lo suficiente para incluir el encabezado del bloque como un registro de recompensa dentro de un nuevo bloque que se construye.

Dos contraincentivos actuarían en oposición entre sí en la práctica, y desincentivarían el correo no deseado y reducirían o eliminarían los ataques maliciosos de denegación de servicio en una cadena de bloques. Por un lado, el costo de la difusión del bloque incentivaría la espera para ver qué nodos de alta Idoneidad que compiten podrían ser generados por otros nodos de construcción de bloques. Si un nodo que compite genera un bloque de mayor Idoneidad, luego no tiene sentido desechar el cargo por token para transmitir un bloque de menor Idoneidad. Por otro lado, un nodo que ha construido un bloque con un Idoneidad reconociblemente alto (según lo determinado por algún umbral preestablecido) puede incentivarse para incluir una gran cuota de construcción, para garantizar mejor la propagación de ese bloque.

En al menos una modalidad, un cargo de construcción se acumularía en la cuenta de recompensa de un nodo de construcción de bloques solamente en el caso de que el bloque incentivado se evalúe como válido, para garantizar que el nodo de construcción de bloques valide el bloque antes de hacer referencia a él desde la cadena de bloques. Si el bloque que contribuye a el cargo de construcción no es válido, luego el segmento de la cadena de bloques que hace referencia a ese bloque sería inválido; esto significaría que el trabajo realizado por el nodo de construcción de bloques necesitaría ser desechado, y que el nodo de construcción de bloques podría incurrir en una penalización a través de un registro de penalidad (descrito a continuación), y como mínimo perdería cualquier recompensa que se habría ganado de cualquier otra manera.

Dado que el cargo de construcción es parte del bloque que se usa para determinar el hash de bloque y, por lo tanto, la Idoneidad, se debería tomar una decisión con respecto al tamaño del cargo antes de que se conozca la Idoneidad (al menos en el caso donde se use el consenso de Idoneidad de distancia hash). La existencia de este cargo, si es lo suficientemente alta, puede desincentivar el correo no deseado, pero un atacante malicioso solamente puede disuadirse por un cargo que es mucho más alto que lo que los nodos cooperativos pueden disponerse o ser capaces de incluir como un cargo. Aun así, este enfoque incentivaría a los nodos solamente a difundir sus bloques de mayor Idoneidad en aquellos casos en los que puedan producir múltiples bloques de manera simultánea. Por lo tanto, incluso si los cargos de construcción son demasiado bajas para mitigar el riesgo de correo no deseado fraudulento, pueden actuar para moderar el tráfico de red y reducir la efectividad de un ataque malicioso (y, por lo tanto, su posibilidad).

### **Registros de Penalidad**

En otra modalidad, los bloques propuestos pueden ir acompañados de una asignación de token que, a diferencia de lo anterior, se reembolsaría siempre y cuando el bloque sea verificable. Los nodos de construcción de bloques incorporarían este monto en riesgo como una manera de garantía para compensar a otros Nodos de Construcción de Bloques que usan tiempo del procesador para validar un “**Bloque Garantizado**”, en caso de que tal bloque sea inválido. Este monto en riesgo no sería un monto fijo; montos mayores incentivarían la adopción de más riesgos por parte de los nodos de construcción de bloques que decidan construir o no en el bloque garantizado.

En al menos una modalidad, este monto en riesgo se incorporaría a un registro de penalidad, qué el registro de penalidad incluiría todo o alguna porción de lo siguiente: el monto en riesgo, la dirección de origen que contribuye a los tokens (la dirección de cuenta propia del token del nodo), el hash de bloque. garantizado y otra información de identificación del bloque que vincule el registro de

penalidad con el bloque propuesto, la ID de red (por ejemplo, una dirección IP) del nodo de publicación (más nodos pares de nivel -1 que también pueden tener las transacciones), y se firmaría con la clave privada de la dirección de origen.

El monto en riesgo se “reembolsaría” (es decir, se considera no asignado) en caso de que se valide y adopte el Bloque Garantizado. Si el Bloque Garantizado no puede validarse, luego el monto en riesgo en el registro de penalidad se asignará al nodo de construcción de bloques agrega satisfactoriamente el registro de penalidad a la cadena de bloques (es decir, el nodo de construcción de bloques que establece satisfactoriamente que el Bloque Garantizado no es, de hecho, válido). La existencia de un registro de penalidad enviará el monto en riesgo a la recompensa de los nuevos bloques.

En al menos una modalidad, en el caso de que se determine que el Bloque Garantizado es inválido, el mecanismo mediante el cual el monto en riesgo se asigna a un nodo de construcción de bloques es el siguiente: Primero, el nodo de construcción de bloques realiza una validación del Bloque Garantizado, al descubrir que el bloque no es válido y al establecer que puede usarse el registro de penalidad. Segundo, el nodo de construcción de bloques agrega el registro de penalidad a un nuevo bloque, lo que hace que la recompensa disponible para ese bloque aumente por el monto en riesgo del registro de penalidad. Tercero, en caso de que el nuevo bloque sea aceptado por otros nodos de construcción de bloques y se incorpore a la cadena de bloques canónica, la recompensa se asignaría, al menos en parte, a la cuenta del nodo de construcción de bloques que generó el nuevo bloque, y el saldo de la cuenta que contribuyó los tokens al registro de penalidad se reduciría por un monto igual. Sin embargo, para confirmar que el bloque que contiene el registro de penalidad es válido en sí mismo, otros nodos de construcción de bloques intentarán refutar la penalización al ensamblar, validar y ejecutar satisfactoriamente los registros de datos que contribuyen a la formación del Bloque Garantizado al que se hace referencia en el registro de penalidad. Un registro de penalidad que hace referencia a un Bloque Garantizado que no es inválido no sería válido en sí mismo y su inclusión invalidaría el bloque que lo contiene. Si el bloque que incluye el registro de penalidad en sí se garantiza por un registro de penalidad separado, una conclusión de que el primer registro de penalidad es inválido haría que el segundo registro de penalidad sea válido, lo que pondría su monto en riesgo a disposición para su inclusión en una recompensa de bloque.

Si ningún nodo de construcción de bloques encuentra que el Bloque Garantizado asociado con un registro de penalidad particular no es válido, luego ese registro de penalidad simplemente se desecharía.

En una modalidad alternativa, un nodo que solicita detalles del bloque de un segundo nodo podría incluir su propia dirección de cuenta en esa solicitud, causando que el segundo nodo genere y firme un encabezado del bloque que transfiere explícitamente una “carga dirigido” a esa dirección. Si un nodo no confía en el bloque creado por otro nodo, en el caso estándar descargará los registros de datos constituyentes del bloque no confiable y validará el bloque antes de intentar construir sobre él (como se describió anteriormente). Si el nodo ve que el bloque del otro nodo tiene un nivel de Idoneidad muy alto, y sin embargo desea comenzar a construir sobre el bloque sin realizar la validación primero, puede solicitar que el nodo emita una “carga dirigido” a través de un registro especial que apunta al bloque. Tal registro de carga dirigido funcionaría como un tipo de registro de penalidad, enviando tokens a la cuenta del nodo receptor solamente en caso de que el bloque recientemente propuesto (un Bloque Garantizado en este caso) sea inválido. De cualquier otra manera, se desechará la carga dirigido.

### **Arreglos de Bloques no Lineales habilitados por la Propagación de Bloques Confiar, pero Verificar**

Cuando se opera dentro de los límites de una cadena de bloques lineal, el consenso de Gradiente de Idoneidad y la variación o el enfoque de “confiar, pero verificar” para la propagación de datos puede aumentar significativamente el rendimiento de un sistema o red de la cadena de bloques. Además de estas mejoras, pueden lograrse más aumentos de rendimiento al optimizar la estructura de datos de la cadena de bloques.

En al menos una modalidad, las limitaciones de linealidad pueden superarse al permitir que múltiples subcadenas o segmentos de la cadena de bloques se incluyan con la estructura de datos de la cadena de bloques principal en paralelo, transformándola en un gráfico acíclico dirigido (DAG).

Cabe destacar también que no todos los sistemas de libro mayor distribuidos actualmente conocidos usan cadenas de bloques lineales como tales. Algunas implementaciones de la cadena de bloques conocidas usan estructuras de datos de red o malla, y otras usan gráficos acíclicos dirigidos (DAG). La metodología de propagación de bloques “confiar, pero verificar” y la variación del consenso de Gradiente de Idoneidad de acuerdo con al menos una modalidad mejoran aún más la viabilidad y utilidad de una estructura de datos DAG para organizar los bloques, en comparación con lo que se ha conocido y usado en el pasado.

(a) Debido a que todos los nodos se construyen y potencialmente transmiten bloques y subcadenas alternativas continuamente, el número de subcadenas o segmentos de la cadena de bloques ya existentes elegibles para la inclusión en la cadena de bloques/DAG aumenta. La metodología de

consenso de Gradiente de Idoneidad en al menos una modalidad proporciona un contexto mediante el cual estos candidatos pueden clasificarse, de manera que el ajuste más adecuado entre ellos puede incluirse con la cadena principal como parte del DAG (suponiendo un cierto límite superior en cuanto a cuántas subcadenas pueden incluirse).

(b) Al menos una modalidad de la metodología de propagación de bloques de “confiar, pero verificar” tiene un efecto de los registros que se procesan masivamente en paralelo en toda la red. Debido a que los diferentes nodos en la red están trabajando en registros asociados con conjuntos de cuentas distintos y mutuamente exclusiva (según lo habilitado por “confiar, pero verificar” a través del uso de los filtros de Bloom), la ocurrencia de cálculos únicos no duplicados aumenta sustancialmente. Mientras más bloques se compartan que se originen en lugares distintos de la red, mayor será la posibilidad de que se incluyan combinaciones únicas de registros como subcadenas paralelas en la cadena de bloques.

(c) Debido a que la propagación de bloques “confiar, pero verificar” exige que cada bloque contenga un filtro de Bloom que represente el conjunto de cuentas con el que opera (es decir, que se ven afectados por su inclusión en la cadena de bloques), dos bloques o subcadenas pueden compararse rápidamente para determinar si ambos se operan en cualquiera de las mismas cuentas. La comparación de los filtros de Bloom puede proporcionar la certeza de que las dos cadenas no funcionan en ninguna de las mismas transacciones.

(d) Dado que las transacciones no se propagan inmediatamente, sino que se descargan después de que se acepta un bloque, es probable que diferentes regiones de la red de nodo conozcan las diferentes transacciones, lo que hace más fácil combinar dos bloques cuando se generan en diferentes regiones de la red, ya que es menos probable que compartan las mismas transacciones (abordando un problema específico discutido por Lewenberg y otros).

El método de consenso de Gradiente de Idoneidad cuando se combina con la variación de propagación de datos de “confiar, pero verificar” descrita anteriormente, puede implementar las siguientes etapas, elementos y características (aunque no todas las etapas, elementos y características son necesarios en cada modalidad).

(1) Según la propagación de datos de “confiar, pero verificar” descrita anteriormente, los Nodos de Construcción de Bloques agregan los registros y las transacciones que generan a sus propios bloques y comparten esos registros y transacciones con pares inmediatos. Las transacciones también pueden compartirse con pares de pares, y pares de pares de pares, etc., hasta un punto, en dependencia de la

implementación específica o modalidad de la presente invención. Un nodo construirá los bloques que contienen principalmente registros “locales” que fueron construidos por ese nodo o sus pares.

(2) Los nodos de construcción de bloques con estos registros y transacciones “locales”, actualizando los filtros de Bloom para reflejar las cuentas de envío y recepción a las que hacen referencia estos registros y transacciones. Los nodos con suficientes recursos informáticos pueden construir múltiples bloques de manera simultánea, en al menos una modalidad con conjuntos de cuentas mutuamente excluyentes que se ven afectados, mediante el uso de los filtros de Bloom para garantizar esta separación.

En al menos una modalidad alternativa de propagación de bloques de “Confiar, pero Verificar”, los Nodos de Construcción de Bloques, Billeteras y otros participantes de la red que generan nuevos registros y transacciones pueden compartir esos registros y transacciones con cualquier Nodo de Construcción de Bloques. Sin embargo, las cuentas y direcciones se identifican como asociadas con ciertos Nodos de Construcción de Bloques, que se consideran sus nodos “locales”. Los registros y las transacciones se comparten preferentemente con los nodos “locales” de las cuentas a las que se hace referencia dentro de esos registros y transacciones, concentrando de esta manera la ocurrencia de los registros y transacciones que hacen referencia a esas cuentas en los bloques generados por los nodos “locales” y sus pares. Esto conduce a una menor superposición entre los bloques con respecto a las cuentas a las que afectan, y a una menor posibilidad de que dos bloques cualesquiera entren en conflicto entre sí.

(3) Los nodos transmiten los detalles del encabezado del bloque (pero no los registros individuales o las transacciones contenidas en el bloque) en toda la red. En al menos una modalidad, los nodos pueden incluir opcionalmente algún monto de cargo del bloque o monto del token en riesgo para incentivar a otros nodos a comenzar a construir en el bloque.

(4) Los nodos evalúan todos los bloques que se difundieron sin (en la mayoría de los casos) descargar y validar los registros detallados.

Las comparaciones de los filtros de Bloom permiten a los nodos elegir un número de nuevos bloques que operan en conjuntos de cuentas mutuamente exclusivas. Pueden encontrarse detalles más técnicos específicos de la implementación para comparar los filtros de Bloom en el Informe técnico CISE 2004 de Agarwal y Trachtenberg, “Estimación del número de diferencias entre los conjuntos remotos”. Los nodos eligen la combinación de los bloques mutuamente exclusivos con la Idoneidad combinada más alta, hasta un cierto límite de cadenas paralelas (es decir, ancho máximo de DAG).

Los nodos comienzan a construir los nuevos bloques sobre estas subcadenas mediante la ejecución de los registros que operan en las cuentas no encontradas entre las subcadenas seleccionadas, según lo determinado por la comparación con el filtro de Bloom. Con la implementación de la presente invención, existe una probabilidad relativamente alta de que si un registro se rechaza después de compararlo con el filtro de Bloom, luego ese registro ya está en la cadena pero no se ha eliminado del conjunto de registros porque la validación aún no ha continuado.

(5) Los nodos pueden construir bloques alternativos en paralelo sobre diferentes bloques o combinaciones de bloques, consistentes con la metodología de propagación de bloques “confiar, pero verificar” descrito en otra parte en la presente descripción. En otras palabras, los nodos pueden crear nuevos bloques en paralelo, y los nuevos bloques pueden tener referencias posteriores a una o más subcadenas. Construir una variedad de combinaciones de bloques ayuda a mitigar el riesgo de que se introduzca un bloque fraudulento, porque si es necesario desechar el hilo principal del trabajo, otro hilo de trabajo podría ser capaz de recolectar la tarea y continuar el proceso de los bloques de construcción.

Esta construcción de nodo incluye la construcción contra el propio bloque del nodo de la generación anterior. En al menos una modalidad, cada nodo siempre tendría al menos un hilo, proceso o construcción de corutina en la cadena de mayor idoneidad que el nodo ha validado completamente.

(6) En paralelo con la construcción del nuevo bloque, se descargan todas las transacciones para el(los) bloque(s) de mayor Idoneidad y se validan los bloques varios. Los registros incluidos se eliminan del conjunto de registros.

Si en algún momento se descubre que un bloque es inválido, o que los registros no pueden obtenerse, luego se pone fin al esfuerzo de construir sobre ese bloque - que incluyen los esfuerzos para construir bloques sobre ese bloque fraudulento en combinación con otros bloques. El esfuerzo principal de construcción se convierte en el esfuerzo que se construye sobre la combinación de bloques y segmentos/subcadenas de la cadena de bloques con la siguiente idoneidad más alta.

Una vez que se hayan descubierto y desechado todos los bloques fraudulentos (si los hay) y se hayan validado todos los bloques restantes, luego puede usarse la combinación óptima de segmentos/subcadenas de la cadena de bloques para construir un bloque optimizado al máximo (si esto aún no ha ocurrido debido a los esfuerzos paralelos realizados para construir el nuevo bloque).

(7) El nuevo bloque que finalmente se construye y difunde a la red de nodo apuntará a todos los segmentos/subcadenas de la cadena de bloques contribuyentes a través de múltiples referencias anteriores incorporadas en el encabezado del bloque. Entre las subcadenas contribuyentes, la

subcadena en particular que tiene la mayor Idoneidad se considerará la cadena principal. En al menos una modalidad que combina esta metodología de propagación de bloques de “Confiar, pero Verificar” mejorada por DAG con consenso de distancia hash, cuando se realizan los cálculos de distancia hash para determinar la Idoneidad, el hash objetivo se encontrará al contar los bloques hacia atrás a lo largo de la cadena principal hasta que se identifique el origen del hash objetivo para ese bloque.

### **Reducción de los Ataques de Denegación de Servicio que apuntan a Arreglos de Bloques no lineales**

Confiar, pero verificar las variaciones consensuadas de la presente invención que emplean arreglos de bloques no lineales pueden ser susceptibles a un ataque de tipo “denegación de servicio” particular que reduciría significativamente el rendimiento de una cadena de bloques implementando esas estrategias de consenso. Específicamente, un atacante puede crear una serie de registros que se originan desde (son firmados por y asignan tokens desde) o terminan con (asignar tokens a) una sola cuenta, pero que se distribuyen a un gran número de nodos que operan en la red. Si los cargos incluidas con esos registros son lo suficientemente altas, luego pueden incluirse en un gran número de nuevos bloques o subcadenas de manera simultánea.

Confiar, pero verificar las variaciones de consenso de la presente invención que emplean arreglos de bloques no lineales aprovechan el hecho de que si dos bloques o segmentos de la cadena de bloques producidos de manera simultánea no contienen registros que operan en las mismas cuentas - en otras palabras, si las cuentas a las que se hace referencia tanto en los bloques como en los segmentos de la cadena de bloques son mutuamente exclusivos - luego tanto los bloques o los segmentos de la cadena de bloques pueden usarse por un nuevo bloque para construir el bloque simultánea en la expansión de la cadena de bloques. Sin embargo, si casi todos los nodos intentan producir bloques que hacen referencia a la misma cuenta, luego el número del bloques contemporáneos o segmentos de la cadena de bloques que pueden agregarse a la cadena de bloques será severamente limitada. Entre los bloques y segmentos de la cadena de bloques que hacen referencia a esa cuenta, solamente se incluirá uno - el más idóneo - lo que dará una ventaja a los bloques que de cualquier otra manera serían menos idóneos y no hacen referencia a esa cuenta (por ejemplo, los bloques creados por un atacante que intente explotar esta característica).

Existe una variedad de estrategias que podrían emplearse para mitigar este riesgo, o al menos para crear fuertes incentivos contra este tipo de comportamiento, aumentando el costo de tal ataque. Con ese fin, el consenso de Gradiente de Idoneidad, de acuerdo con al menos una modalidad de la

presente invención, puede implementar las siguientes etapas, elementos y características (aunque no todas las etapas, elementos y características son necesarios en cada modalidad).

(1) Las cuentas pueden especificar opcionalmente en sus datos de perfil en cadena o configurar una lista corta y priorizada de los nodos de construcción de bloques preferidos. Además, pueden mantenerse las estadísticas con respecto a la frecuencia de registros que hacen referencia a cada cuenta (por ejemplo, un conteo de referencia total o un conteo de referencia dentro de los últimos 100 bloques, etc.). En al menos una modalidad, para cuentas de alto volumen (por ejemplo, cuentas comerciales y cuentas de negocios), cualquier registro que haga referencia a esas cuentas puede requerir enviarse directamente a los nodos preferidos o reenviarse a estos, si esos nodos están en línea. Solamente los nodos preferidos de estas cuentas incorporarían tales registros en sus bloques, porque otros nodos querrían evitar el riesgo de que sus bloques no se agreguen a la cadena.

(2) En al menos una modalidad, si un registro particular hace referencia a dos cuentas de alto volumen, y si ambas cuentas designan diferentes nodos preferidos, luego el registro puede reenviarse al nodo preferido de la cuenta que registra entre sus estadísticas de uso un mayor volumen de actividad en cadena (medida por el número de registros o cargos totales agregados). Alternativamente, el registro puede enviarse a todos los nodos preferidos enumerados por estas cuentas, cuyos nodos pueden competir entre sí por el bloque que producen para ser aceptados en la cadena de bloques. Siempre y cuando los nodos que intentan incluir tales registros se limiten a una pequeña fracción del número de nodos de construcción de bloques activos en la red, el rendimiento total no debe disminuir como un resultado de los demasiados conflictos que surgen de los nuevos bloques que contienen el mismo registro.

(3) Un nodo de construcción de bloques puede requerir un registro explícito para las cuentas, de manera que si una cuenta no se registra con ese nodo, el nodo no aceptará los registros que hagan referencia a esa cuenta. Tal registro puede, en al menos una modalidad, requerir cierta confirmación de identidad en el mundo real a través de canales fuera de la línea. También puede coincidir con ese nodo que se enumera entre los nodos preferidos de la cuenta en cuestión. En al menos una de tales modalidades, una cuenta se registraría con un número limitado de nodos, y si la cuenta firma transacciones que se han incluido en bloques diferentes emitidos por demasiados nodos, luego la cuenta puede perder su registro.

(4) En al menos una modalidad, un nodo de construcción de bloques puede mantener una lista negra de cuentas que no permitirá que los registros hagan referencia; cualquier registro que haga referencia a tales cuentas se desecharía y no se incluiría en ningún bloque que se construido por ese nodo. Las

cuentas que terminan en esa lista negra pueden, en algunas implementaciones, ser cuentas a las que hace referencia un bloque que compite y, por lo tanto, generar un bloque o segmento de la cadena de bloques construido por el nodo en cuestión que se pierde a otro bloque. Alternativamente, las cuentas que terminan en la lista negra pueden ser aquellas cuentas que repetidamente (con una frecuencia que excede un determinado umbral, o con tal frecuencia que no puede ser contabilizada al azar) envían los tokens a una cuenta que también envía o recibe tokens en los bloques contemporáneos contruidos por otros nodos en la red.

(5) Algunas implementaciones de los nodos de construcción de bloques pueden rechazar un registro que ha sido firmado por una cuenta si los registros recientes firmados por esa cuenta han sido difundidos e incorporados en bloques por otros nodos. Un nodo de construcción de bloques se beneficiará si una cuenta es leal, porque si los registros que hacen referencia a esa cuenta aparecen ocasionalmente en los bloques que compiten, los conflictos entre los bloques serán más frecuentes, y será más difícil para ese nodo que sus bloques o segmentos de la cadena de bloques sean aceptados satisfactoriamente por el resto de la red. En un ejemplo extremo, si una cuenta en particular solamente envía sus registros a un nodo de construcción de bloques, luego esa cuenta en sí nunca será la causa de ningún conflicto que pueda dar lugar a que se rechacen los bloques contruidos por ese nodo. Sin embargo, este arreglo puede no ser beneficioso para las cuentas, ya que puede reducir la probabilidad de que sus registros se agreguen a la cadena de bloques, al no aparecer en los suficientes bloques propuestos.

(6) En al menos una modalidad de la presente invención, un registro puede incluir una referencia a un bloque reciente en la cadena a la que reivindica pertenecer, y una altura del bloque de expiración más allá de la cual el registro ya no sería viable. También puede especificarse una ventana de expiración máxima en todo el sistema de manera que no se pueda agregar un registro mucho después de que se haya creado su bloque de referencia (independientemente de la altura de expiración especificada dentro del registro mismo).

Este enfoque forzaría a un registro a pertenecer a un segmento de la cadena de bloques que contiene el bloque de referencia, por el cual el riesgo del mismo registro existente en más de un segmento de la cadena de bloques se limitaría de manera que solamente pertenecería a los segmentos de la cadena de bloques que se bifurcan del bloque de referencia antes de que haya pasado la ventana de expiración. En al menos una modalidad, puede ser necesario que los registros firmados por una cuenta determinada tengan que apuntar a los bloques dentro del mismo segmento de la cadena de bloques que contiene otros registros firmados por la misma cuenta - lo que significa que todos los

registros firmados por una cuenta existirían dentro de los bloques de antepasado directo de cualquier bloque que contiene otros registros firmados por la misma cuenta, y no dentro de un segmento de la cadena de bloques paralelo dentro de la DAG.

Para que algunas de las estrategias anteriores funcionen, tendrían que ser empleadas por cada nodo de construcción de bloques en la red e implementadas en el software de la cadena de bloques usado por cada nodo de construcción de bloques. Por ejemplo, las estrategias que requieren modificaciones a la representación de datos, o que requieren que la información o configuración se adjunte a objetos de datos en cadena como cuentas (almacenamiento de nodos preferidos, conteos de registros) y registros (almacenamiento de referencias de bloque, altura de expiración), tendrían que facilitarse por cada nodo dentro de una modalidad.

Sin embargo, para que otras estrategias funcionen, no todas tienen que ser empleadas por cada nodo de construcción de bloques en la misma manera. Determinados nodos de construcción de bloques pueden emplear algunas estrategias, mientras que otros no lo hacen, y los nodos diferentes que emplean las mismas estrategias pueden tener niveles de tolerancia o configuraciones diferentes relacionadas con esas estrategias. Por ejemplo, algunos nodos pueden usar listas negras agresivas, mientras que otros no lo hacen; algunos nodos pueden negarse a manejar los registros no asociados con cuentas registradas, mientras que otros no lo hacen; y algunos nodos solamente pueden manejar las cuentas que los enumeran como nodos preferidos, mientras que otros pueden procesar cualquier registro que reciban.

### **Optimización del Consenso de Distancia Hash mediante Nodos Participantes**

Una propiedad y característica importante del consenso de Gradiente de Idoneidad y la variante de propagación de bloques “confiar, pero verificar” descrita en la presente descripción es que el hardware especializado más capaz de acelerar el proceso más allá de los límites de rendimiento actuales será el hardware que sea más capaz de acelerar el código de rotación completo que reside dentro de los contratos inteligentes, y el código que usó para validar los nuevos bloques. Los dispositivos similares a los chips Bitcoin ASIC - “Circuitos Integrados de Aplicación Específica” que se especializan en realizar cálculos de prueba de trabajo rápidamente - mejoran el rendimiento de solamente un aspecto del proceso de construcción de bloques en los sistemas actuales - específicamente, el proceso de hash criptográfico usado para descubrir las soluciones de prueba de trabajo - y pueden no proporcionar una ventaja competitiva suficiente para justificar su costo y uso para otros fines. El consenso de Gradiente de Idoneidad de la presente invención en al menos una modalidad proporciona el incentivo para desarrollar y mejorar el hardware especializado para otros

cálculos, que incluyen los cálculos genéricos. Los avances y mejoras en tal hardware serían ampliamente beneficiosos en un amplio intervalo de contextos, aparte de los algoritmos de consenso de la cadena de bloques.

Al implementar el consenso de Distancia Hash, una mejora en la tasa de hashing de un nodo aumentará la posibilidad de que construya bloques de alta Idoneidad, al aumentar la tasa y el número del bloques válidos que el nodo puede construir. Sin embargo, a medida que la distancia hash se aproxima a cero, hay retornos decrecientes a escala para tal esfuerzo. Las iteraciones repetidas del cálculo de la distancia hash pueden reducir la distancia hash de bloque mediante la búsqueda aleatoria. En cada iteración, el contenido del bloque solamente debe modificarse de maneras pequeñas - por ejemplo, modificando los metadatos de algunos registros de datos triviales - para que se calculen los valores de distancia hash muy diferentes. Sin embargo, la probabilidad de que cada intento subsecuente descubra una variación particular del contenido del bloque que produzca una distancia hash mejor que cualquier disminución previamente descubierta a medida que se hacen más intentos.

Por otro lado, durante el tiempo que se tarda en realizar tales cálculos, los participantes de la red continúan usando más tokens al gastar y transferir más tokens, distribuir más cargos y asignar los tokens a los casilleros (en el caso de una implementación del consenso de Casillero). Cualquier intento iterativo de mejorar la distancia hash como se describió anteriormente - independientemente de si se usa o no la optimización de hardware - quedaría superado por la generación de estos registros, y tendría que volver a configurarse para incorporar estos nuevos registros. Por lo tanto, intentar acercarse a la distancia cero optimizando la operación de hash se vuelve menos valioso a medida que aumenta la cantidad de nuevos registros y las transacciones de difusión a través de la red, y a medida que se usan más tokens. El punto de inflexión podría estar más cerca de cero para las búsquedas iterativas aceleradas por hardware, pero todavía está allí. El punto de inflexión en el que una búsqueda iterativa de distancias hash más bajas ya no vale la pena, el esfuerzo puede estar más cerca de cero cuando mediante el uso de la aceleración de hardware de las operaciones hash, pero en cualquier caso hay un límite más allá de lo que tiene más sentido pasar al siguiente bloque en lugar de intentar optimizar el bloque actual.

La aleatorización básica en toda la red, con todos los nodos de construcción de bloques operando en paralelo, puede dar lugar a distancias hash muy cercanas a cero para los bloques de mayor Idoneidad, y si bien las optimizaciones de hardware pueden hacer que este número baje, tal esfuerzo probablemente no tendría el mismo beneficio que optimizar el tiempo de ejecución de los contratos

inteligentes y otras operaciones esenciales de construcción de bloques. Para optimizar la Idoneidad dentro del consenso de distancia hash,

una búsqueda aleatoria acelerada en el espacio de los posibles registros de datos y combinaciones de bloque/subcadena - realizada mediante la creación de nuevas combinaciones de bloques y registros de datos a medida que estos elementos se vuelven conocidos por los nodos de construcción de bloques - no solamente optimizará la distancia hash (creando al azar nuevos bloques con valores de distancia hash más cercanos a cero), sino que también optimizará los valores de token nativos únicos. Es la combinación de estos dos valores lo que contribuye a la fórmula de Idoneidad de Distancia Hash y a la eficiencia y velocidad del método.

Por lo tanto, la aceleración de hardware de las operaciones hash se ve perjudicada por recursos informáticos ordinarios masivamente paralelos capaces de computar los contratos completos de Turing. Esta tendencia permite que las cadenas de bloques impulsadas por el Gradiente de Idoneidad implementadas de acuerdo con al menos una modalidad de la presente invención para proporcionar los incentivos monetarios adicionales para mejorar la velocidad del cómputo en general. La inversión en la aceleración de los contratos inteligentes completos de Turing no es una optimización con es la única aplicación que es el rendimiento de la cadena de bloques, ya que los aceleradores hash especializados (ASIC) de Bitcoin y Ethereum son, sino que serían ampliamente valiosos para mejorar la velocidad del cómputo en todas las aplicaciones.

### **Consenso de Gradiente de Idoneidad con Retrasos de Tiempo**

En otra variación del consenso de Gradiente de Idoneidad de la presente invención, incorpora los retrasos de tiempo específicos en el proceso de generación de bloques. Cada bloque propuesto (que incluye los bloques que se agreguen subsecuentemente) cumpliría con un segmento de tiempo específico.

Tal segmento de tiempo puede definirse de diferentes maneras:

- El segmento de tiempo puede definirse en términos de tiempo de reloj - descrito como un desplazamiento contra un punto en el tiempo particular, o denominado de alguna manera estándar en relación con los estándares de cronometraje (UTC, etc.)
- El segmento de tiempo puede definirse en términos de un número de unidades informáticas esenciales o etapas necesarias para ejecutarse en secuencia
- El segmento de tiempo puede definirse en términos de una asignación de la prueba de trabajo con un cierto nivel de dificultad, lo que tomará una cierta cantidad de tiempo prevista para ejecutarse en promedio.

El número de registros aceptados en un bloque (es decir, el tamaño del bloque) estaría limitado en relación con la duración del segmento de tiempo. Cada bloque tendría permitido incluir los registros correspondientes a un tiempo de ejecución particular. En el bloque solamente se aceptará una cantidad de registros que represente un tiempo de ejecución que coincida exactamente o aproximadamente alguna fracción  $Q$  de la duración total del segmento de tiempo.

Alternativamente, se permitiría que cada bloque incluyera los registros correspondientes a un tiempo de transmisión total esperado en particular en toda la red (que sea tanto una función de latencia de la red como un tamaño de datos). En el bloque solamente se aceptará una cantidad de registros que se puede esperar razonablemente que se transmitan a través de la red de nodo dentro de una fracción  $Q$  de la duración total del segmento de tiempo.

En otra modalidad, tanto el tiempo de ejecución esperado como el tiempo de transmisión se tomarían en cuenta al determinar el tamaño máximo del bloque. En el bloque solamente se aceptará una cantidad de registros que pueda esperarse que se transmita a través de la red de nodos y luego se ejecute subsecuentemente dentro de alguna fracción  $Q$  de la duración total del tiempo del segmento. La fracción  $Q$  de la duración total del tiempo del segmento, qué fracción se usa para determinar el tamaño del bloque, se establecería para permitir que los nodos validen los bloques y las cadenas que compiten mediante la interpretación de los registros que contienen. Debe haber disponible suficiente tiempo de inactividad dentro del proceso de construcción de bloques para permitir que un nodo atraviese una larga historia de bloques más antiguos y se ponga al día con el bloque actual dentro de un marco de tiempo razonable.

En al menos una modalidad de la presente invención, cada nodo construirá los nuevos bloques encima de cualquier cadena que se haya encontrado que tiene el valor de idoneidad más alto. El proceso de encontrar la cadena con el valor de Idoneidad más alto será el siguiente.

(1) Los nodos pasarán entre sí el valor de Idoneidad (y otra información de identificación) de la cadena que cada uno construye actualmente (es decir, la cadena de Idoneidad más alta que cada nodo ha validado completamente). Adicionalmente, los nodos también pueden compartir información sobre la cadena de mayor Idoneidad que conocen, pero que aún no han podido validar, junto con la dirección del nodo que anuncia esa cadena.

(2) En el caso de que un nodo tiene conocimiento de una cadena que sea de mayor Idoneidad que su cadena actual, puede solicitar que esa cadena se descargue desde el nodo que anuncia esa cadena (o la porción de esa cadena o segmento de la cadena de bloques que se aparta de la cadena ya validada del nodo solicitante).

(3) Al recibir los datos de la nueva cadena (que incluye todos los registros), el nodo solicitante interpretaría todos los registros en secuencia hasta que pueda confirmar que todos los bloques (y registros contenidos) son válidos, y que la idoneidad de esa cadena es de hecho mayor que cualquier otra cadena que haya validado (especialmente su cadena activa actual).

(4) Debido al hecho de que este proceso de validación puede dar lugar a que la nueva cadena que compite sea rechazada -porque es inválida, o porque el valor de idoneidad anunciado era incorrecto, o por algún otro motivo - un nodo solicitante (validante) podría simultáneamente seguir construyendo al mismo tiempo su cadena de mayor idoneidad ya validada. Cada nuevo bloque que se construye comenzará a construirse solamente después de que haya pasado el retraso de tiempo requerido (es decir, el segmento de tiempo) para el bloque anterior.

(5) La cadena actual de un nodo solamente se reemplazaría en el momento en que la validación de la cadena de la competencia se haya alcanzado con el mismo número del bloque que la cadena actual, de manera que la nueva cadena pueda usarse sin problemas para construir un bloque y el segmento de tiempo adecuado.

En al menos una modalidad, el retraso de tiempo requerido entre los bloques puede hacerse cumplir al requerir un espacio mínimo de tiempo entre bloques, o requerir que se incluya un número máximo de bloques en la cadena de bloques dentro de una determinada cantidad de tiempo. En esta modalidad, cada bloque identificaría la fecha y hora de su creación e inclusión en la cadena de bloques (es decir, su marca de tiempo), desde el primer bloque hasta el último.

Un segmento de la cadena de bloques o de la cadena de bloques se consideraría inválido (1) si la separación entre los bloques cayera por más abajo de un determinado umbral, o si el número del bloques en la cadena excediera el número del bloques esperada para las marcas de tiempo incluidos con los bloques; o (2) si los bloques más recientes al final de la cadena de bloques no se alineaban con la hora actual, ya que el nodo de construcción de bloques lo entiende. De manera similar, los nuevos bloques que se evalúan por los nodos de construcción de bloques solamente serían válidos (1) si la marca de tiempo de cada bloque está dentro de algún umbral de la hora actual según lo entiende el nodo de construcción de bloques, y (2) si la marca de tiempo del bloque no está demasiado cerca de la marca de tiempo del bloque precedente en la cadena de bloques. Cualquier cadena de bloques, segmento de la cadena de bloques o bloque inválido se descartará en lugar de una cadena de bloques, segmento de la cadena de bloques o bloque alternativo y válido.

En al menos otra modalidad, el retraso de tiempo puede aplicarse al requerir que los bloques incluyan una prueba de trabajo que sea de un nivel de dificultad informática que sería ejecutable sobre una

base amortizada dentro de la cantidad de tiempo representado por el segmento de tiempo. Un bloque sería válido solamente si (a) la dificultad representada mediante esta prueba de trabajo corresponde sobre una base amortizada a la duración del segmento de tiempo, y si (b) el bloque ha completado satisfactoriamente la prueba de trabajo, según lo demostrado por el nonce y el hash de bloque.

En al menos una modalidad, la duración obligatoria del segmento de tiempo impuesta por los métodos descritos anteriormente puede cambiar en dependencia del rendimiento informático y de la red del sistema de la cadena de bloques. Si se requiere más o menos tiempo para que se realicen los cálculos de la construcción de bloques y la validación de bloques, o para que los datos se compartan en toda la red, luego la duración del segmento de tiempo puede aumentarse o disminuirse. Tales cambios en la duración del segmento de tiempo provocarían un cambio en el espacio mínimo de tiempo entre los bloques, el número máximo de bloques que pueden incluirse en un segmento de la cadena de bloques para una determinada cantidad de tiempo, o al nivel de dificultad de un problema de la prueba de trabajo usada para separar los bloques.

La fracción  $Q$  de la duración total del segmento de tiempo usada para determinar el tamaño del bloque debe elegirse cuidadosamente de manera que sea informáticamente posible que un nodo construya nuevos bloques en tiempo real de acuerdo con el cronograma de tiempo del segmento, mientras que el nodo valida simultáneamente una cadena que compite de cierta longitud  $L$  que puede reemplazar a su cadena activa actual. El proceso de validación debe continuar a un ritmo más rápido que el proceso de construcción de bloques; de cualquier otra manera, nunca alcanzará el bloque actual.

Por ejemplo, si la fracción  $Q$  es  $1/3$ , luego puede ser posible que un nodo valide una subcadena de longitud  $L=6$  bloques dentro de aproximadamente el mismo tiempo que llevaría construir dos nuevos bloques y agregarlos a la cadena actual del nodo, asumiendo la ejecución paralela de ambos procesos. Problemas de tiempo similares también afectarían el inicio de un nuevo nodo que se uniera a la red (es decir, un nodo que no tiene ninguna cadena de bloques o datos del registro). Un nuevo nodo que inicia sesión en la red debe obtener todos los bloques y registros de la cadena de bloques que han sido aceptados por sus nodos pares en la red. Aunque puede haber alguna variación entre los nodos pares con respecto a los bloques más recientes en sus cadenas individuales, el conjunto de bloques más grandes y abrumadores que regresan al origen de la cadena debe ser idéntico.

Esto puede hacerse mediante la recuperación de datos del bloque y datos del registro a través de la red de nodos pares, o puede hacerse mediante la instalación de un empaque de datos que encapsula los datos en el bloque de manera fuera de banda (por ejemplo, al copiar tal paquete de datos desde

una unidad USB u otro medio físico, o al descargar el paquete desde algún servidor a través de la red).

En cualquier caso, un nodo debe validar que todos tales datos sean correctos y ejecutar los contratos inteligentes y procesar los datos para crear una instancia de tiempo de ejecución local del estado global del sistema. Esto probablemente se haga ejecutando todo el historial de registros y contratos inteligentes en la cadena en secuencia. Hasta que esto se haya completado, los datos deben considerarse potencialmente inválidos o incluso maliciosos. Sin embargo, una vez completada, el nodo puede tratar esta cadena de bloques de terceros y los datos del registro como válidos y verificados.

Después de la descarga inicial del bloque de reajuste del nodo nuevo y los datos relacionados, el resto de la red agregará los bloques a un ritmo constante. Como un resultado, incluso después de que el nuevo nodo haya completado la verificación y validación de los datos de reajuste, ese nuevo nodo tendrá que descargar y verificar los nuevos bloques, los registros y otros datos generados desde esa descarga inicial.

El valor de  $Q$  determinará qué tan avanzado se habrá ejecutado el resto de la red para el momento en que el nuevo nodo haya completado su verificación de la cadena de bloques y los datos relacionados, como una función de la longitud de la cadena de bloques al momento de la descarga inicial. El valor de  $Q$  también determinará cuánto tiempo tardará el nuevo nodo en alcanzar completamente el estado actual de la red después de verificar los datos iniciales.

El orden en el que pueden evaluarse los diferentes tipos de registros para la incorporación en un bloque puede limitarse por los requisitos de esta variación. Si hay algunas operaciones que *deben* ejecutarse antes de finalizar el bloque - y que pueden implicar directamente la creación de los registros adicionales y las etapas adicionales de ejecución - luego estos elementos no deben dejarse al final del proceso de creación/ejecución del bloque, y deben procesarse al comienzo del proceso.

Otros registros que pueden ser opcionales y podrían ser incluidos/interpretados/evaluados al final del proceso, de manera que puedan omitirse o cortarse en caso de que se alcance el tamaño máximo del bloque. Por ejemplo, los registros de casillero son una parte obligatoria del algoritmo de consenso de Casillero, por lo que deben incorporarse primero o al menos cerca del comienzo del proceso de creación del bloque.

Por ejemplo, en al menos una modalidad, los registros de liquidación y emparejamiento comercial (descrito en otra parte en la presente descripción) pueden considerarse partes obligatorias de cualquier bloque cuando se implementa la negociación. La implicación de realizar la conciliación y

liquidación de las operaciones al comienzo del proceso de creación de los bloques, antes de que se agreguen nuevos registros de órdenes comerciales al bloque, es que en tales circunstancias la conciliación y la liquidación de las órdenes comerciales solamente debe hacerse para las órdenes de operaciones agregadas a la cadena de bloques en los bloques anteriores.

Sin embargo, para evaluar los registros que invocan los contratos inteligentes completos de Turing, tal como los que pueden implementarse con el lenguaje Solidity, es posible que se requiera una duración variable e impredecible. Aunque cada contrato inteligente debe ser determinista en su ejecución en el punto en el que se incluye en la cadena de bloques (de manera que cada nodo de validación vea el mismo resultado), debido al “problema de detención” (proposición clásica de ciencias informáticas probada por Alan Turing), la cantidad de tiempo requerida para ejecutar tal contrato no es conocida sin ejecutar realmente el contrato.

El tiempo de ejecución de estos contratos puede estar limitado por el requisito de que las etapas informáticas individuales se paguen a través del uso de un cargo por token conocido como “gas”, de manera que incluso una contracción inteligente que implemente una lógica no terminal finalizará cuando sus etapas computacionales excedan lo que se ha pagado por adelantado. No obstante, debido a que este límite máximo representa solamente un límite superior, la cantidad real de tiempo que tardan estos contratos inteligentes será variable, incluso si tienen un límite máximo.

En un punto determinado del proceso de ejecución de tales contratos, el nodo superará el punto en el cual el tiempo de ejecución disponible para el bloque. El último registro incorporado/interpretado/evaluado en un nuevo bloque por un nodo será el registro cuya interpretación primero sobrepase el umbral de tiempo, o el registro inmediatamente precedente al registro que hace que se sobrepase el umbral de tiempo. Todos los demás registros propuestos se clasifican, se deben descartar o retrasar para su inclusión en un bloque futuro.

Otros registros, tales como las transferencias de saldo, las nuevas órdenes comerciales o los registros relacionados con propuestas, son potencialmente más predecibles en su desempeño y pueden incorporarse en el medio del proceso. Los diferentes tipos de registros se describen más abajo en más detalles. Si estos registros se vinculan a los contratos inteligentes a través de los registros de vinculación de patrones, es posible que deban tratarse como si invocaran directamente los contratos inteligentes completos de Turing y se clasificaran en la misma manera.

Para evitar que la evaluación de los contratos completos de Turing sea obligatoria, es posible que se requiera que los registros de vinculación de patrones no sean obligatoriamente ejecutables (de manera que no haya garantía de que se ejecuten). O, alternativamente, puede ser necesario tener

solamente una cantidad limitada de ciclos de cálculo disponibles para los contratos vinculados en cada bloque, y forzar los registros de vinculación de patrones para ofertar por su inclusión mediante el uso de cargos. Tal límite podría exigirse de manera conservadora asumiendo que cada contrato inteligente consumirá la totalidad de su asignación de gas, independientemente del resultado final.

### **Algoritmos de Suministro de Token**

Entre las diversas implementaciones de la cadena de bloques, existe una amplia variedad de estrategias con respecto a cómo se crean, distribuyen, administran y destruyen los tokens nativos (a menudo denominados “monedas” en las cadenas de bloques de divisas). Al menos una modalidad se refiere al “suministro” de tales tokens.

No todas las implementaciones de la cadena de bloques usan los tokens nativos. Por ejemplo, el software de la cadena de bloques que usa un consenso basado en votos autorizados, tal como la Fabric Hyperledger de IBM, no requiere la existencia de ninguna moneda o token nativo para incentivar la participación en el proceso de consenso. Fabric Hyperledger se usa por empresas que operan en un entorno de “fideicomiso parcial”, por lo que los nodos que se unen a la red se autorizan explícitamente a hacerlo por los otros nodos, y están incentivados a participar debido a los beneficios comerciales auxiliares que sus operadores reciben como un resultado de la existencia de la red, tal como la marca de tiempo de los documentos y la preservación de documentos. La contribución informática realizada por los nodos en tal red es esencialmente un obsequio, dado para beneficiar a un grupo cerrado o semicerrado de participantes sin recibir consideración directa a cambio.

Por el contrario, las redes de cadena de bloques que usan los tokens nativos típicamente están abiertas a la participación de cualquier nodo anónimo capaz de conectarse a la red a través de Internet. Estas implementaciones (por ejemplo, Bitcoin y Ethereum) usan principalmente los tokens nativos para incentivar la participación de los nodos que de cualquier otra manera no tendrían un incentivo para participar en la construcción de la cadena de bloques y, de esta manera, ayudar a asegurar la red contra el ataque de las partes no confiables.

Dentro de las redes de cadena de bloques de la prueba de trabajo, el valor de mercado de los tokens, el número de tokens emitidos y pendientes en un momento dado, la forma en que se distribuyen los tokens, y la tasa y la manera en que se crean los nuevos tokens tienen un impacto directo en la potencia informática que se aplica para resolver los problemas de prueba de trabajo. Los mineros de la prueba de trabajo gastan dinero del mundo real (moneda “por decreto” en términos de la industria) para pagar equipos de minería y electricidad para realizar los cálculos de la prueba de trabajo; recuperan sus costos vendiendo los tokens nativos generados a cambio de la moneda “por decreto”.

Históricamente, las tendencias en el valor del mercado de los tokens han tenido un impacto directo en la potencia informática total que se aporta para soportar la resolución de los problemas de la prueba de trabajo, lo que tiene un impacto directo en la seguridad general del sistema (más poder informático significa mayor seguridad).

Una caída drástica y repentina del valor de un token nativo puede dar como resultado una rápida caída de los participantes en la red, incluso mientras la capacidad física permanece intacta. Por ejemplo, si alguna vez se llevara a cabo un ataque de doble gasto en la red de Bitcoin, una forma probable de que suceda sería que un gran minero desaproveche su capacidad de extracción después de una fuga de mercado, y usar esa capacidad para extraer una cadena duplicada en paralelo y en secreto. Para poner en escena un ataque, tal un actor de este tipo (1) compraría una gran cantidad de Bitcoin; luego (2) usaría su capacidad para comenzar a extraer una cadena secreta; luego (3) vendería su Bitcoin, finalizaría el acuerdo en la cadena pública; (4) liberaría su cadena secreta más larga, lo que reemplazaría a la cadena principal, eliminaría el registro de la venta; y finalmente (5) vendería su Bitcoin por segunda vez. En otras palabras, la volatilidad de los precios de los tokens puede disminuir la seguridad de las redes de la prueba de trabajo.

En las redes de cadena de bloques de prueba de participación, cuanto mayor sea el valor de los tokens nativos en la cuenta designada de un nodo determinado, más podrá "apostar" ese nodo en una ronda determinada. Una "apuesta" más alta aumenta las probabilidades de que el nodo elegido construya el siguiente bloque y reciba la recompensa de bloque. Los nodos que puedan "apostar" un mayor número de tokens/monedas acumularán, por lo tanto, más tokens a través de las recompensas de bloques, y tendrán una influencia desproporcionada en las determinaciones del consenso.

Sin contramedidas incorporadas, las cuentas que ya son los mayores titulares de tokens nativos continuarán acumulándolas a expensas de usuarios comunes que pagan los cargos para agregar los registros a la cadena de bloques. Esta tendencia se acelera si el algoritmo que se usa para distribuir los tokens nativos al comienzo también da como resultado la concentración. La concentración conlleva riesgos e implicaciones específicos para un sistema de prueba de participación: si todos los nodos, excepto algunos, se enfrentan a una baja probabilidad de construir un nuevo bloque (y beneficiarse de las recompensas), luego los nodos tienden a salirse de la red y se niegan a participar. Menos nodos participantes aumentan la posibilidad de confabulación entre (o de control común sobre) una mayoría de nodos, lo que afecta sustancialmente la seguridad y aumenta la exposición de la red a un posible ataque de doble gasto.

### **Variación entre los Algoritmos de Suministro**

Los tipos de algoritmos que se usan para determinar cómo se distribuyen los tokens y cuántos tokens se distribuirán son variados. Por ejemplo, el primer bloque de la cadena de bloques, el “bloque de génesis”, puede contener en sí mismo una asignación de token. Este suministro inicial a veces se denomina un “premine” o un “instamine”, aunque el término “suministro inicial” es más preciso. En dependencia de cómo se generen y distribuyan los tokens subsecuentes a este evento, puede dar lugar a un alto grado de centralización de la propiedad del token.

Como parte de cada bloque, pueden generarse nuevos tokens. La cantidad total de tokens que se generarán con el tiempo puede limitarse, o puede ser potencialmente infinita.

Bitcoin genera tokens adicionales cada bloque en función de la altura del bloque de la cadena: el mismo número de nuevo Bitcoin se añaden cada bloque, excepto que cada 210 000 bloques ese número se reduce a la mitad. Se seguirán generando nuevos Bitcoins hasta que el suministro total sea igual a 21 millones. En 2019, se crearon 12.5 Bitcoins en cada bloque. Ethereum emite actualmente 3 éteres constantes por bloque, y no hay un límite integrado; a menos que se realice un cambio en el software de Ethereum y se adopte por una masa crítica de nodos, se generarán 3 éteres por bloque mientras exista Ethereum, y puede generarse un suministro potencialmente infinito de tokens de Ether.

En algunas redes de la cadena de bloques o criptomonedas, tal como Ripple, los tokens generados al comienzo comprenden el suministro máximo de tokens que se generarán; en tales casos, no hay protocolo para emitir más tokens. Algunas implementaciones de cadena de bloques destruyen los tokens. En el caso de Ripple, los tokens usados como cargos se eliminan permanentemente de la circulación, lo que hace que el suministro total de tokens disminuya gradualmente con el tiempo.

Los algoritmos de emisión y suministro implementados en el software de la cadena de bloques afectan sustancialmente la adopción de usuarios, los niveles de participación continuos y la volatilidad de precios de los tokens nativos. La cantidad total de tokens pendientes influye en la fijación de precios a través de la oferta y la demanda, y la tasa y el mecanismo de emisión de tokens influyen en el comportamiento del usuario a través de incentivos. A pesar de esto, muchas o la mayoría de las implementaciones de cadena de bloques y criptomonedas han elegido sus algoritmos de emisión y suministro sin tener objetivos claros y a largo plazo con respecto al usuario.

Por el contrario, la presente invención pretende reducir la volatilidad y mejorar la previsibilidad en cuanto al precio de los tokens nativos con el tiempo, al usar información en cadena que es accesible a cada nodo de construcción de bloques para decidir de manera determinista el número de tokens nativos a emitir o destruir en un bloque determinado.

### **Algoritmos de Estabilidad de Precios**

Los tokens nativos dentro de un sistema de la cadena de bloques sirven para incentivar las actividades de construcción de bloques de los nodos de construcción de bloques, pero también sirven para el propósito esencial de monetizar los servicios proporcionados por la cadena de bloques. Los tokens nativos son un medio para que los usuarios de la cadena de bloques compensen la red por prestar servicios. Generalmente, estos servicios incluyen la preservación de datos en cadena y el procesamiento informático asociado con la creación de un nuevo bloque. Es razonable esperar que el precio de un token nativo en relación con los activos y las divisas del mundo real externo varíe de acuerdo con la utilidad percibida de esos servicios. Sin embargo, no es necesariamente el caso de que la cantidad de tokens nativos asignados como cargos para participar en esos servicios deba ser volátil.

Esencialmente, pueden adoptarse dos enfoques distintos para estabilizar el precio o el valor fuera de la cadena de los tokens, donde cada uno puede implementarse de acuerdo con al menos una modalidad de la presente invención.

De acuerdo con el primer enfoque, un sistema de la cadena de bloques puede designar una clavija de valor para anclar el precio del token en términos de un activo o moneda del mundo real (por ejemplo, USD, SDR). Para estabilizar el precio de ese token, debe realizarse alguna intervención humana o algorítmica - por ejemplo, comprando tokens a cambio del activo fijo, o vendiendo tokens a cambio del activo fijo, o creando o destruyendo tokens, o bloqueando o desbloqueando tokens. Aunque hay algunos sistemas que han intentado usar al menos algunas de las metodologías mencionadas anteriormente (por ejemplo, los tokens de Tether), la presente invención ofrece una serie de mejoras sobre los sistemas conocidos.

De acuerdo con el segundo enfoque, un sistema de la cadena de bloques puede identificar alguna métrica de valor interna a la cadena de bloques en sí y realizar una intervención algorítmica para estabilizar esa métrica, independientemente de la información de precios de mercado externo. Un ejemplo de esta métrica de valor serían los cargos de token nativo pagados por los usuarios que desean agregar datos a la cadena de bloques, o que desean que los nodos realicen algún trabajo informático (es decir, precios de “cargo por servicio” expresados en términos del token usado para ese propósito). Este segundo enfoque es único para la presente invención, y no se encuentra entre los sistemas de la cadena de bloques conocidos.

Dos nuevos procesos y métodos, cada uno de acuerdo con al menos una modalidad de la presente invención, se describen más abajo. El primer proceso y método descrito es una implementación del

segundo enfoque de “métrica interna” introducido anteriormente, junto con severas variaciones. El segundo proceso y método descrito es una aplicación del primer enfoque de "clavija de valor" introducido anteriormente.

### **Suministro de Token a través de la Orientación de la Métrica Interna**

De acuerdo con al menos una modalidad de la presente invención, el suministro de tokens nativos puede administrarse de tal manera que los cargos de tokens nativos pagados por los usuarios que desean agregar datos a la cadena de bloques, o que desean tener nodos que realicen algún trabajo informático, serán generalmente consistentes (aunque no necesariamente fijos) con el tiempo.

Con ese fin, puede establecerse un objetivo para lo que debería ser el cargo por token promedio para agregar los registros o para realizar trabajo en la cadena de bloques (es decir, la cadena de bloques “cargo por servicio”). Este objetivo puede fijarse durante todo el período de tiempo que la red de cadena de bloques funciona, pero se definiría mejor como una función de la altura del bloque actual, de manera que la carga por servicio promedio objetivo pueda variar con el tiempo.

En al menos una modalidad, si el cargo por servicio promedio actual para los nuevos registros en la cadena de bloques cae por debajo del objetivo, luego se crearía y agregaría cierta cantidad de nuevos tokens nativos a la recompensa de bloque. Esta cantidad de tokens se determinaría de manera determinista en el momento en que se crea un bloque. Específicamente, en al menos una modalidad, la cantidad de nuevos tokens nativos se determinaría al examinar el número  $n$  de bloques inmediatamente *precedentes* al bloque en cuestión, y determinar algún promedio ponderado de la discrepancia entre el cargo por servicio promedio real de cada bloque, y el cargo por servicio objetivo en ese número del bloque. En otras palabras, se usaría un promedio móvil ponderado, pero solamente un promedio ponderado de bloques inmediatamente precedentes al bloque donde se asignarían los nuevos tokens nativos.

Al ser todos iguales, aumentar la recompensa de bloque aumentaría de esta manera la cantidad total de tokens en circulación en la cadena de bloques y aumentaría el suministro global de tokens disponibles. Esto, a su vez, puede esperarse reducir gradualmente el precio que los usuarios deben pagar para adquirir los tokens y hacer que los usuarios asignen los valores del cargo más altos al proponer los nuevos registros para agregar a la cadena de bloques.

Por otro lado, si el cargo por servicio promedio actual para los nuevos registros en la cadena de bloques se eleva por encima del objetivo, luego la recompensa de bloque disminuiría en cierta cantidad de tokens nativos. Como en el caso anterior, esta cantidad se determinaría de manera determinista en el momento en que se crea un bloque. Específicamente, en al menos una modalidad,

la cantidad de tokens mediante los cuales se reduce la recompensa se determinaría al examinar el número  $n$  de bloques inmediatamente *precedentes* al bloque en cuestión, y determinar algún promedio ponderado de la discrepancia entre el cargo por servicio promedio real de cada bloque, y el cargo por servicio objetivo en ese número del bloque. Como en el caso anterior, se usaría un promedio móvil ponderado, pero solamente para los bloques inmediatamente precedentes al bloque donde se asignarían los nuevos tokens nativos.

Si todo lo demás es igual, disminuye la recompensa de bloque en este caso disminuiría el número total de tokens en circulación en la cadena de bloques y disminuiría el suministro global de tokens disponibles. Esto, a su vez, puede esperarse aumentar gradualmente el precio que los usuarios deben pagar para adquirir los tokens y hacer que los usuarios asignen los valores del cargo más bajos al proponer los nuevos registros para agregar a la cadena de bloques.

En cualquier caso, el tamaño de la recompensa de bloque estaría al menos ligeramente desacoplado del tamaño de los cargos especificadas en los registros agregados a la cadena de bloques. Aunque un bloque que tiene los cargos más altas en total producirá una recompensa mayor que un bloque que tiene los cargos más bajas en total, la recompensa final real para el bloque sería un porcentaje de esos cargos (ya sea el 150 % de los cargos, o el 50 % de los cargos, o el 0 %), qué valor porcentual dependería de una función de los cargos asignadas en los bloques precedentes. La recompensa final puede depender del cargo por servicio promedio ponderado para los bloques precedentes, como se describió anteriormente, y en al menos una modalidad puede depender de la tasa de cambio en los cargos registradas para esos bloques.

La recompensa de bloque no puede caer por debajo de cero, por lo que el impacto máximo de reducir las recompensas de bloque sería proporcionalmente menor que el impacto máximo de aumentar la recompensa de bloque al crear los nuevos tokens para aumentar el suministro total. La recompensa de bloque no sería reducible a cero, sino a un nivel mínimo predefinido.

Dicho esto, sin embargo, la rescisión de los tokens que ya están en circulación proporciona una manera de efectuar una reducción en el número de tokens nativos en circulación más allá de este límite. En el caso del consenso de Casillero los algoritmos similares, en al menos una modalidad, si el cargo por servicio promedio aumenta demasiado por encima del objetivo (es decir, más allá de algún umbral alto definido expresamente), luego alguna porción de los tokens bloqueados o los tokens en riesgo pueden reducirse también para reducir aún más los tokens en circulación.

Retirar los tokens bloqueados de esta manera tendría el efecto de disminuir la cantidad total de tokens en circulación a expensas de disminuir el incentivo para que los participantes bloqueen o apuesten

los tokens (al imponer efectivamente una tasa de interés negativa en esos tokens), lo que afectaría negativamente la seguridad. La siguiente sección describe más abajo las posibles estrategias alternativas de rescisión de tokens que no crean este incentivo perverso.

La reducción de las recompensas de bloques y la rescisión de los tokens pueden destruir los tokens en cuestión, así como un aumento en las recompensas de bloques puede crearlos nuevos tokens que no existían antes. Lo que significa que es probable que haya una discrepancia entre los cargos cometidos por los registros incluidos en el bloque y la recompensa de bloque emitida para ese bloque. En efecto, cuando se emplea un algoritmo de suministro de tokens como el descrito aquí, el número total de tokens nativos residentes dentro del sistema de la cadena de bloques antes de que se procese un bloque puede ser mayor o menor que la cantidad de tokens nativos residentes dentro del sistema después de que se procese. De esta manera, el suministro global de tokens nativos está influenciado por el precio de los servicios de la cadena de bloques - es decir, el nivel de cargos incluido en los nuevos registros agregados a la cadena de bloques, en promedio.

### **Rescisión de Token y Selección de Métricas**

En al menos una modalidad del método de suministro de token “métrica interna” descrito anteriormente, si el cargo por servicio promedio actual cayera por debajo de la métrica objetivo por un determinado umbral, luego se anularía alguna porción de los tokens en circulación general. En dependencia de la implementación y la modalidad, se anularía un porcentaje de tokens que no han sido asignados o movidos en absoluto dentro de la cadena de bloques durante algún tiempo; o se rescataría un cierto porcentaje de tokens que se han mantenido durante un cierto período de tiempo sin haber sido usados como cargos; o en el caso del consenso de Casillero (y algoritmos similares) se anularía un porcentaje de tokens que no habían sido bloqueados o apostados durante un período de tiempo; o alguna combinación de estos.

Esta “cargo de acumulación”, “penalización por falta de pago” o “penalización por inactividad” incentivaría a los tokens inactivos a ser usados activamente, lo que ejercería una presión ascendente sobre la fijación de precios de los cargos por servicio al aumentar el número de tokens en circulación activa. En al menos una modalidad, los tokens anulados en sí mismos pueden reintroducirse en la circulación activa al agregarse, en su totalidad o en parte, a las recompensas en bloque disponibles para la distribución; alternativamente, los tokens anulados pueden destruirse, en su totalidad o en parte.

Además de incentivar un aumento inmediato en la circulación general de tokens, este enfoque con el tiempo volvería a la circulación de los tokens nativos que se habían perdido efectivamente debido

a las credenciales criptográficas (es decir, clave privada) de la cuenta que tenía esos tokens perdidos. Esto también ayudaría a reducir el riesgo de un aumento masivo de tokens que vuelven a entrar *en* circulación *en masa*, lo que tendría un gran impacto tanto en el precio externo pagado por tales tokens, así como también en la fijación de precios de cargo por servicio que experimentan los usuarios reales. A diferencia del aumento de la estabilidad que se proporciona por al menos una modalidad de la presente invención, en el caso de la cadena de bloques de Bitcoin, hay miles de millones de dólares en Bitcoin que se supone ampliamente que están permanentemente bloqueados y, por lo tanto, perdidos. Sin embargo, no es seguro que esos Bitcoins se pierdan permanentemente, y existe cierto riesgo de que esos tokens latentes vuelvan a entrar en circulación en algún momento, lo que afecta el precio del Bitcoin en el mercado.

En el caso del consenso de Casillero (o los enfoques de consenso de Distancia Hash e Idoneidad relacionados donde los tokens bloqueados o en apuestas contribuyen al valor de Idoneidad) de acuerdo con al menos una modalidad, asignar los tokens a los casilleros y bloquearlos los eximiría de la penalización por inactividad. Como un resultado, además de estabilizar el valor de los tokens nativos, habría un incentivo para que se bloqueen o estacionen mayores cantidades de tokens en un momento determinado, lo que aumentaría la seguridad del Algoritmo de Consenso.

Para determinar qué saldos de cuentas estarían sujetos a esta rescisión, los nodos de construcción de bloques se remontarían a través de la cadena de bloques hasta el bloque en el que se distribuyeron los tokens como parte de una recompensa de bloque, o hasta el bloque en el que se liberaron los tokens después de haber sido bloqueados o apostados. La cantidad de bloques entre el bloque actual y ese bloque específico se consideraría la “edad” del token.

En al menos una modalidad, si una cuenta mezcla los tokens de diferentes edades, se considerará que cada transferencia de esa cuenta ha transferido tokens de esas diferentes edades en proporción a las edades de los tokens originalmente mantenidos por la cuenta. Cuando se asignan cargos o se asignan tokens a un casillero (o se bloquean o se apuestan los tokens), luego se considera que los tokens “más antiguos” se han asignado primero, antes de que se hayan asignado los tokens más jóvenes.

Para fines de optimización, en al menos una modalidad, la antigüedad de los varios saldos de token mantenidos por una cuenta en cualquier momento dado puede rastrearse como parte de los datos de la cuenta que residen en el bloque, de manera que la edad no tiene que volver a calcularse después de la validación inicial.

En al menos una modalidad, el porcentaje de tokens anulados como parte de la “penalización por inactividad” puede determinarse mediante una función que considere los factores tales como la edad del token en combinación con la discrepancia entre el cargo por servicio objetivo y el cargo por servicio real experimentado durante los bloques  $n$  más recientes.

En otra modalidad de la presente invención, los tokens completamente latentes - los tokens que no se han incluido en ningún registro de datos escrito en la cadena de bloques después de cierta altura del bloque bien definida - se anularían a una cierta velocidad. En cada bloque, se anularía algún porcentaje de tokens inactivos; sin embargo, el tamaño de este porcentaje puede o no depender de la discrepancia actual entre el cargo por servicio promedio objetivo y el cargo por servicio promedio actual. En al menos una modalidad, la ocurrencia de esta rescisión por token latente dependería solamente de la edad del token, y se evaluaría independientemente de la “penalización por inactividad” descrita anteriormente y la “asignación de retención” descrita más abajo.

### **Asignaciones de Retención y Selección de Métricas**

En al menos una modalidad del método de suministro de tokens “métrica interna” descrito anteriormente, si el cargo por servicio promedio actual supera el objetivo por un determinado umbral, luego cierta cantidad de tokens nativos se distribuiría a las cuentas que tienen los tokens nativos. Esta “asignación de retención” podría implementarse mediante la asignación a estas cuentas de alguna porción de la recompensa que de cualquier otra manera se distribuiría a los participantes del consenso; o mediante la emisión de un nuevo valor del token no contabilizado previamente en cadena (en otras palabras, a través de la emisión de nuevos tokens nativos creados a través de la inflación); o mediante una combinación de estos.

Esta “asignación de retención” inflaría el tamaño de todos los saldos de tokens nativos de manera simultánea en toda la red por un porcentaje que se determina, al menos en una modalidad, por cuán grande es la discrepancia entre la métrica de los cargos por servicio promedio objetivo y la métrica de los cargos por servicio promedio actual entre los  $n$  bloques más recientes agregados a la cadena de bloques. Cuanto mayor sea la discrepancia, mayor será el porcentaje.

Esta asignación de retención, sin embargo, solamente se aplicaría a las cuentas con una “edad” por encima de un determinado umbral, o se aplicaría proporcionalmente a las cuentas en base a la antigüedad de los tokens que lo contienen. De esta manera, la asignación de retención también incentivaría a los titulares del token nativo a continuar reteniéndolos y no usarlos (y no venderlos para usarlos), disminuyendo potencialmente el número de tokens en circulación activa.

### **Configuración y Ajuste del Suministro de Tokens Basados en Métricas Internas**

Los enfoques anteriores de acuerdo con al menos una modalidad del método de suministro de tokens “métrica interna” de la presente invención, pueden incorporar un número de constantes y funciones que en dependencia de los detalles elegidos en el momento de la implementación tendrían un efecto material en el comportamiento del software y los incentivos que motivan a sus usuarios. Estos pueden incluir:

- La función de cargo por servicio promedio objetivo
- La(s) función(es) de aumento y disminución de recompensas de bloque
- La(s) función(es) de asignación de retención y penalización por inactividad
- La función de porcentaje de rescisión de token inactivo

Cada una de estas funciones puede implementarse de una manera que se integre directamente en el software del nodo de construcción de bloques, o pueden implementarse en su totalidad o en parte como contratos inteligentes que pueden modificarse en tiempo de ejecución por los usuarios privilegiados.

Si una función se integra en el software del nodo de construcción de bloques, todos los nodos en la red de cadena de bloques deben compartir la misma versión de la implementación de esa función. Para que la validación del bloque trabaje correctamente, se requiere que todos los nodos acepten el resultado de esta función. Una implicación de este requisito es que un porcentaje sustancial de todos los nodos - más que una mayoría - debe actualizar su software y acordar la misma versión de ese software para que ese ajuste entre en vigencia. La programación de tales sistemas puede implementarse en diferentes lenguajes de programación y/o para diferentes sistemas operativos por un experto en la técnica que sigue la descripción de las características y limitaciones del sistema que se describen en la presente descripción y que reflejan, aplican e implementan adecuadamente en la implementación de codificación específica.

Alternativamente, cualquiera de estas funciones que se implemente como un contrato inteligente en la cadena de bloques puede sustituirse por otro contrato inteligente agregado a la cadena de bloques sin requerir que se implemente ninguna versión nueva del software. Para sustituir tal un contrato inteligente de este tipo por otro contrato inteligente, una o más cuentas autorizadas firmarían el registro del nuevo contrato inteligente con instrucciones para sustituir el registro del contrato inteligente antiguo. Alternativamente, un único contrato inteligente podría programarse para delegar la responsabilidad a un contrato subordinado, cuyo contrato subordinado podría ser modificado por una mayoría de cuentas autorizadas con derecho a voto. Los bloques anteriores a este acto de

reemplazo serían válidos solamente si se usara la versión antigua de la función, mientras que los bloques posteriores a este acto de reemplazo serían válidos solo si se usara la nueva versión de la función.

Hay que hacer algunas consideraciones especiales con respecto a la función promedio de cargo por servicio. La naturaleza de esta función puede tener un impacto significativo en los incentivos de los usuarios potenciales para participar en la red al ejecutar los nodos y agregar los registros a la cadena de bloques. Por ejemplo, si la función es lineal, luego el nivel de cargos por servicios crecerá a una tasa constante con el tiempo. Esto puede proporcionar estabilidad, o puede desalentar la adopción desde el principio si las recompensas de bloque disminuyen para reducir los aumentos rápidos de cargos a medida que la red comienza a proporcionar servicios públicos a los usuarios - o puede desalentar la adopción si se anulan los tokens para incentivar la velocidad de los tokens antes de que la red haya alcanzado la masa crítica de los nodos participantes. En otro orden, por ejemplo, una función que se asemeja a una función sigmoide (desplazamiento de manera que  $y \approx 0$  a  $x=0$ ) podría abordar estos problemas al proporcionar una transición suave desde un periodo de cero o bajos cargos, a una existencia estabilizada, con cargos más altos, a largo plazo. Para alinear tal transición con la red que alcanza una masa crítica de nodos, tal función también podría tener en cuenta el rendimiento actual de la red, o qué tan dramáticos y consistentes pueden ser los picos de los precios; la presión de precios más ascendente podría desencadenar la transición al nivel estable más alto.

Alternativa o adicionalmente, podrían implementarse funciones más matizadas teniendo en cuenta el comportamiento de la red después de la implementación. El reemplazo de estas funciones de suministro podría realizarse si la mayoría de los nodos de construcción de bloques acepta la modificación y actualiza su software. Una función que se implementa como un contrato inteligente podría ser, pero si un consenso se construye entre los operadores de nodos de que una implementación de contrato inteligente en particular es óptima y debe preservarse, la lógica de esa función puede implementarse por sí misma en el software del nodo de construcción de bloques, dándole un estado semipermanente que no puede reemplazarse por nada más que un amplio consenso de los nodos de construcción de bloques de la red.

### **Suministro de Token a través de Orientación de Clavija de Valor**

Un ejemplo adicional de al menos un sistema y método de la presente invención que mejora el precio del token nativo y la estabilidad del suministro se discute con referencia a la **Figura 7**.

Las implementaciones de la cadena de bloques típicamente mantienen el valor agregado de los tokens nativos dentro del sistema de la cadena de bloques constante, o lo aumentan a una velocidad

predeterminada (ya sea a una velocidad constante o con una caída gradual). Esto típicamente resulta en las criptomonedas sean deflacionarias, con valores de token que aumentan en comparación con los activos del mundo real a medida que aumenta el uso.

Al menos una modalidad de la presente invención intenta usar los datos en cadena relacionados con las operaciones entre los pares de tokens nativos y tokens del usuario como indicadores de inflación o deflación. La creación o destrucción de tokens nativos se activa de acuerdo con estos indicadores. Con referencia a la **Figura 7**, al comienzo, en la etapa 710, se emite determinada cantidad de tokens nativos. Las recompensas de bloque pueden denominarse en estos tokens nativos. En 715, los registros que se destinan a la inclusión en la cadena de bloques deben incluir cargos para incentivar la inclusión. Estos cargos deben ser en la forma de tokens nativos dentro de esa cadena de bloques. En 720, se transmite un registro a la red de nodos para su inclusión en la cadena de bloques.

En 725, los nodos primero verifican si los cargos asignados con el registro se denominan en tokens nativos. Si es así, en 728, el registro se incluye en la cadena de bloques y los cargos se agregan a la recompensa de bloque. De cualquier otra manera, si no se denomina a los tokens nativos, en 730, el nodo de construcción de bloques intenta hacer coincidir estos tokens del usuario del cargo con un registro de orden comercial que convertiría la denominación del cargo(en tokens del usuario) en tokens nativos.

En 735, se determina si se encuentra un registro comercial recíproco coincidente (o una secuencia de registros). Si se encuentra tal comercio recíproco o secuencia de comercios, entonces en la etapa 740, el nodo de construcción de bloques liquida un comercio, y convierte los cargos denominados en tokens definidos por el usuario (o moneda) en tokens nativos.

En 745, el nodo de construcción de bloques compara el comercio con operaciones dentro del número N de bloques, y calcula un porcentaje de desviación P. Luego determina un 750 si el porcentaje de desviación P está por encima del límite umbral superior o inferior. Si es así, en 755, se usa alguna función del porcentaje de desviación y de los cargos R, que pueden describirse como  $R(P, \text{cargos})$ , para calcular la contribución del registro a la recompensa de bloque.

En al menos una modalidad, esta función se implementaría para aumentar la recompensa si el precio de los tokens nativos es demasiado bajo con relación al precedente histórico, a fin de aumentar desproporcionadamente el suministro general de tokens. Si el precio negociable para los tokens nativos es demasiado alto con relación al precedente histórico, luego esta función reduciría o eliminaría la recompensa, para suprimir el suministro general de tokens. En cualquier caso, una vez

calculada la recompensa, en la etapa 790, el registro se agrega al bloque con una recompensa que es mayor o menor que el cargo del registro, en dependencia de la salida de la función R.

De cualquier otra manera, si la desviación P no supera el umbral en la etapa 750, los cargos convertidos en tokens nativos se agregan a la recompensa de bloque en 758, y el registro original se agrega a la cadena de bloques.

Si en la etapa 735, se determina que no se encuentra un registro comercial recíproco coincidente (o una secuencia de comercios), luego en 770, el historial de transacciones se escanea y analiza para determinar si los tokens de cargos de venta de volumen comercial V para los tokens nativos se han liquidado dentro de X número del bloques. Como parte de este proceso, en 775, se determina si el volumen comercial V es mayor que algún umbral mínimo permitido. De lo contrario, luego el volumen de negociación no es lo suficientemente alto y el registro original se considera inválido en 778. Como un resultado, no se incluye ni se agrega a la cadena de bloques.

Si el volumen V está por encima del umbral mínimo, luego en 780, el nodo de construcción de bloques genera los nuevos tokens nativos y los intercambia por los tokens de cargos. En 785, los tokens de cargos de tipo usuario originales se comprometen a una orden comercial que ofrece intercambiar por los tokens nativos, y con las ganancias que van a la cuenta de recompensa de bloque. Después de eso, en 790, el registro original se incluye en la cadena de bloques, contribuyendo cierto monto a la recompensa de bloque que es algo mayor o posiblemente menor que el valor del cargo, pero dentro del nivel permitido de fluctuación.

Por lo tanto, el sistema mantiene la estabilidad del precio y el suministro del tipo de token nativo.

### **Gestión y Seguridad Mejoradas de las Claves Criptográficas**

La seguridad de la propiedad de tokens, criptomonedas u otros activos mantenidos o rastreados en una cadena de bloques depende típicamente de la capacidad de esa persona de mantener en secreto la clave criptográfica privada que se usa para firmar registros, transacciones o contratos inteligentes que se originan desde esa cuenta. Tales registros, transacciones o contratos inteligentes, cuando se firman, son capaces de efectuar la asignación de esos tokens, activos o unidades de criptomoneda a las cuentas de otras personas.

Si una clave criptográfica privada se torna conocida por una persona que no sea el propietario de la cuenta asegurada por esa clave, luego esa otra persona típicamente comparte el mismo control sobre el contenido de la cuenta, incluso si no tiene la propiedad de la misma. Este control de facto se extiende hasta el punto de poder transferir esos activos fuera de la cuenta original y a una cuenta fuera del alcance de su propietario legal.

Una o más características y métodos de la presente invención resuelven este problema al reducir el riesgo de acceso subrepticio de claves por actores maliciosos. Estas características y métodos también resuelven ciertos obstáculos regulatorios causados por la incertidumbre con respecto a la custodia de las claves criptográficas. En algunas jurisdicciones, los corredores de bolsa pueden tener prohibido comercializar y administrar los activos sobre los cuales no pueden mantener un control exclusivo. Lo siguiente es un medio por el cual tales entidades pueden fundamentar una afirmación de que mantienen el control total sobre los tokens en su custodia.

En al menos una modalidad de la presente invención, a cada cuenta se le pueden asignar múltiples pares de claves criptográficas, cada una administrada por separado, y cada una puede autorizar los registros que se originan de - y modifican el estado de - la cuenta. Estos pares de claves consistirían en un componente público y un componente privado. Estos componentes públicos y privados funcionarían para permitir la creación y verificación de las firmas digitales mediante el uso de un algoritmo de firma digital estándar tal como ECDSA.

En al menos una modalidad, cada par de claves de cuenta tendrá una configuración separada relativa a las actividades que puede autorizar y los registros que puede firmar. Si un registro se firma por un par de claves que no se autoriza para generar ese registro, luego el registro es inválido, incluso si la cuenta en sí misma puede originar válidamente ese registro. Uno o más pares de claves maestras designados podrían modificar la configuración adjunta a los otros pares de claves que se asocian con una cuenta.

Los diferentes pares de claves pueden corresponder a los diferentes dispositivos que actúan como billeteras, lo que brinda al usuario o usuarios la capacidad de controlar qué actividades pueden realizarse y qué registros pueden generarse en cada dispositivo. Esto permitirá que los dispositivos más seguros realicen más actividades - y tengan acceso a mayores valores de token - que los dispositivos menos seguros.

En al menos una modalidad, por ejemplo, la billetera instalada en un teléfono inteligente puede usar una clave que se ha configurado para generar solamente registros de transferencia, y solo mover 100 tokens por día. Una computadora del hogar o de la oficina, por el contrario, puede usar una clave con un límite de 1000 tokens por día y se autoriza para firmar la mayoría de los tipos de registros. Por otro lado, una computadora air gapped usaría una clave que tenga autorización sin restricción. Sin embargo, todas estas claves estarían accediendo a la misma cuenta.

Para reducir la posibilidad de que las claves privadas se roben sin ser detectadas, y para garantizar que cada par de claves se asocie con un dispositivo a lo máximo, en al menos una modalidad, cada

clave sería utilizable una sola vez, reemplazada después de cada uso. Cuando se usa un par de claves, el registro comparte la porción pública del reemplazo en la cadena de bloques, y la clave privada se retiene por el dispositivo de billetera. Toda la configuración del par de claves original se transfiere al par de claves de reemplazo. Alternativamente, una configuración puede controlar cómo se reemplaza el par de claves frecuentemente.

En al menos una modalidad, los pares de claves de cuentas individuales se asociarían con las subcuentas, cuyo propósito es realizar un seguimiento de la configuración que se adhiere al par de claves específico, y para vincular los pares de claves a los dispositivos, y los dispositivos a las cuentas. Cuando se genera y firma un registro, el número de la subcuenta se incluye en ese registro para identificar correctamente qué par de claves se debe usar para firmarlo.

En caso de que un par de claves antiguo y ahora reemplazado firme un registro, en al menos una modalidad esto provocará que el dispositivo/subcuenta particular, o la cuenta completa, se bloquee y se vuelva inutilizable. Cada actividad en cadena y registro correspondiente a esa cuenta o a ese dispositivo/subcuenta se consideraría inválido.

El hecho de que toda la cuenta se bloquee, o solamente el dispositivo/subcuenta se bloquee, dependería de la configuración de la cuenta y del dispositivo/subcuenta. De manera similar, la duración del bloqueo dependería de la configuración. El bloqueo podría ser por un período limitado de tiempo, o podría ser indefinido, hasta que se den ciertas etapas para desbloquearlo.

La reactivación de la cuenta o del dispositivo requeriría un proceso para regenerar la clave para el dispositivo inactivo, similar a la forma en que se agregan los nuevos dispositivos y los pares de claves en primer lugar:

- (a) Un dispositivo autorizado y aún activo genera un nuevo par de claves para el dispositivo desactivado, actualiza la porción pública en la cadena de bloques a través de un mensaje de reactivación y empaqueta el par de claves en un mensaje para compartir con el dispositivo o billetera inactivos.
- (b) El nuevo mensaje clave se comparte con el dispositivo desactivado a través de un Código QR, o a través de una URL enviada por mensaje de texto/correo electrónico, a través de una conexión de enchufe de red, a través de una transmisión de radio, a través de una entrada manual o a través de otros medios.
- (c) El dispositivo recientemente reactivado instala el par de claves e inmediatamente genera un mensaje de la cadena de bloques para reemplazar el par de claves compartido por un nuevo par de claves.

En entornos de alta seguridad, la reactivación puede requerir que se usen todos o la mayoría de los dispositivos/cuentas afiliados para generar la nueva clave — incluidas las claves fuera de la línea. En este caso, en al menos una modalidad, el procedimiento de reactivación anterior se repetiría para múltiples dispositivos activos, con el dispositivo inactivo que permanece inactivo hasta que se haya agregado un número suficiente de mensajes de reactivación a la cadena de bloques.

Una implicación del método de rotación de claves y bloqueo de cuentas descrito anteriormente es que los nuevos registros propuestos por una cuenta tendrían que especificar su altura de órdenes o bloque. Si la rotación de claves es frecuente, entonces el orden en que los registros se agregan a la cadena de bloques importa, porque los registros fuera de orden pueden activar un bloqueo.

Esta característica de bloqueo de cuenta, en combinación con la función de rotación de clave descrita anteriormente, forzaría uno de dos resultados en caso de robo de clave: ya sea que el ladrón de la clave sea bloqueado rápidamente para usar la cuenta, o si el ladrón de la clave tiene éxito y se alertaría al propietario de la cuenta al ser bloqueado para usar la cuenta tan pronto como ocurra el uso no autorizado de la cuenta. Los límites en el tamaño de las transferencias de tokens, y en los tipos de actividades que un par de claves de cuenta puede autorizar, significan que si se roba una clave privada, el impacto económico está limitado.

Para habilitar esta funcionalidad, y reajustar la distribución de claves a dispositivos distintos, el siguiente método y proceso se implementaría de acuerdo con al menos una modalidad de la presente invención.

- (1) Un dispositivo de billetera genera una nueva cuenta en la cadena de bloques y produce un nuevo par de claves que está asociado con una subcuenta maestra para esa nueva cuenta. Dado que este dispositivo de billetera contiene y controla este par de claves maestras, debe considerarse la billetera maestra.
- (2) Esta billetera maestra genera un nuevo par de claves secundarias asociado con una subcuenta secundaria de la cuenta.
- (3) La billetera maestra agrega la porción pública del par de claves a la cadena de bloques pública, mientras que la billetera maestra retiene la porción privada. La billetera maestra configura ese par de claves secundarias según las necesidades del usuario y agrega esta configuración a la cadena de bloques.
- (4) La billetera maestra empaca el par de claves secundarias en un mensaje para compartir con un dispositivo de billetera secundario. El mensaje se codifica en formato binario, como código de barras de 2 dimensiones o código QR, como datos de base 64 dentro de una URL u otro formato.

(5) La billetera secundaria, a través de una conexión de enchufe de red, a través de una transmisión de radio, a través de un escaneo de un código QR, a través de la entrada manual, o por otros medios, recibe de la billetera maestra el mensaje de reemplazo de clave.

(6) Finalmente, el dispositivo de billetera secundaria instala el par de claves e inmediatamente genera un mensaje de cadena de bloques para reemplazar el par de claves compartido por un nuevo par de claves, lo que le da al dispositivo control exclusivo sobre el par de claves.

En el caso de que se requiera un nivel de seguridad más alto, en al menos una modalidad, un modo fuera de la línea permitiría que las claves secretas fuera de la línea se configuren como una medida de seguridad, y luego se mantengan en “almacenamiento en frío” hasta que sean necesarias.

Esta sería una manera de agregar claves adicionales a una cuenta, para proporcionar alguna capacidad de anulación en caso de que una cuenta sea secuestrada por el robo de sus claves en línea. Alternativamente, las claves con permisos completos con respecto a una cuenta pueden mantenerse fuera de la línea en almacenamiento en frío, mientras que los dispositivos de billetera activos funcionan mediante el uso de claves que solo tienen permisos parciales.

[0001] El método de generación de claves fuera de la línea de la presente invención puede implementar las siguientes etapas, elementos y características (aunque no todas las etapas, elementos y características son necesarios en cada modalidad).

(1) Un dispositivo de billetera que tiene el par de claves maestras para una cuenta se desconecta y se aísla en un entorno seguro, para reducir la posibilidad de hackear durante este proceso. Este dispositivo de billetera Air Gapped actuaría como la billetera maestra para los fines de este proceso.

(2) La billetera maestra genera un nuevo par de claves secundarias asociado con una nueva subcuenta fuera de la línea de la cuenta.

(3) La billetera maestra genera y firma un registro agregando la porción pública del par de claves a la cadena de bloques pública, mientras retiene la porción privada. La billetera maestra configura ese par de claves fuera de la línea según las necesidades del usuario, y genera y firma además un registro agregando esta configuración a la cadena de bloques. Si el par de claves fuera de la línea pretende ser un medio para bloquear el control total de la cuenta, entonces esta etapa de configuración puede dar al par de claves los mismos permisos que el par de claves maestras de la billetera.

(4) Debido a que la billetera maestra no está conectada a la cadena de bloques a través de ninguna red, estos registros no se agregan automáticamente a la cadena de bloques. En su lugar, se codifican visualmente mediante el uso de un código QR o de algún otro medio visual de almacenamiento de datos, y se imprimen en papel mediante el uso de una impresora conectada directamente a la billetera.

- (5) La billetera maestra luego empaqueta el par de claves fuera de la línea en un mensaje, codificado visualmente mediante el uso de un código QR o de algún otro medio visual de almacenamiento de datos. Esta codificación visual se imprime mediante el uso de una impresora conectada a la billetera, o se muestra en un monitor de computadora u otra pantalla visual conectada a la billetera.
- (6) Un dispositivo de billetera compatible, colocado en un área menos segura, escanea el código QR de los dos registros mediante el uso de una cámara conectada y luego transmite esos registros a la cadena de bloques, actualizando de esta manera la cadena de bloques para reflejar la porción pública del nuevo par de claves fuera de la línea y su configuración.
- (7) El mensaje físico impreso fuera de la línea (es decir, el código QR) que contiene el par de claves se mantiene en una instalación segura hasta que sea necesario. Por ejemplo, puede guardarse en una caja fuerte, caja de seguridad o bóveda bancaria.
- (8) En lo adelante, la billetera maestra puede volver a conectarse a la red, pero para máxima seguridad permanecerá desconectada, apagada o incluso destruida. Sin embargo, antes de que se retire de servicio, el billetera maestra de este proceso puede seguir este mismo procedimiento fuera de la línea para crear nuevamente uno o más pares de claves, en este caso con permisos reducidos. Estos pares de claves se cargarían en dispositivos de billetera conectados, que administrarían el manejo cotidiano de la cuenta, con permisos reducidos para reducir el impacto de un posible hackeo. Tenga en cuenta que este proceso depende de una característica de la presente invención donde, en al menos una modalidad, un dispositivo de billetera puede recibir un registro codificado directamente de otro dispositivo de billetera antes de que ese registro se haya agregado a la cadena de bloques. El dispositivo de billetera receptor puede entonces agregar ese registro a la cadena de bloques en nombre del dispositivo de billetera emisor, cargando el registro en la red de cadena de bloques. En el caso descrito anteriormente, los medios para transferir tal registro son mediante código QR escaneado visualmente, pero en otras modalidades el mensaje puede codificarse en formato binario, como datos de base 64 dentro de una URL, u otro formato además de un código de barras de 2 dimensiones o un código QR. Tales datos pueden compartirse con el dispositivo de billetera receptor a través de una conexión de enchufe de red, a través de una transmisión de radio, a través de un escaneo de un código QR, a través de una entrada manual, o a través de otros medios, además de a través de medios de escaneo visual de un código QR o similares.

### **Mitigación del Riesgo de Pago**

Un desafío común que enfrentan los usuarios de la cadena de bloques cuando usan tokens para realizar pagos en tiempo real, especialmente pagos minoristas, es que lleva algún tiempo finalizar o

confirmar un pago o transferencia por token en la cadena de bloques.

Primero, un registro o transacción de pago o transferencia debe aceptarse para un bloque que se agrega a la cadena de bloques. Segundo, el riesgo de que ese bloque se anule (es decir, ser parte de una bifurcación que se desecha a través de un proceso de resolución de conflictos) debe reducirse. El riesgo de que un bloque se anule disminuye a medida que se construyen más bloques sobre él, porque cada bloque adicional aumenta la idoneidad de toda la cadena.

Sin una rápida confirmación o finalización, es difícil para las transacciones comerciales, especialmente las transacciones minoristas, usar tokens de cadena de bloques como medio de pago. Tanto los clientes y como los comerciantes, típicamente no quieren esperar a que el pago se liquide antes de que los productos y servicios se transmitan del vendedor al comprador.

La presente invención ayuda en el proceso de finalizar o confirmar pagos o transferencias en al menos una modalidad al disminuir el retraso entre bloques y aumentar la tasa a la que se agregan los nuevos bloques a la cadena de bloques, a través de procesos y métodos descritos en la presente descripción.

En al menos una modalidad de la presente invención, el siguiente proceso mejora aún más la capacidad del receptor de un pago o transferencia para reducir el riesgo de que ese pago o transferencia se revierta, aumentando de esta manera la confianza de ese receptor de que se le asignará el control sobre esos tokens de una manera duradera.

(1) Una parte, el “Receptor”, genera una solicitud de pago que especifica que una determinada cantidad de tokens se transfieran a una determinada cuenta. La solicitud de pago se codifica como un mensaje que contiene el token de la cantidad solicitada y la cuenta o dirección de destino a la que se enviará el mensaje, así como también otra información que incluye, en al menos una modalidad, detalles detallados de los servicios o productos para los cuales se solicita el pago (como en una factura); detalles sobre la identidad del solicitante (por ejemplo, nombre, dirección, número de teléfono, fotografía), y el tipo de token solicitado en el caso de que haya más de un token alojado dentro del sistema de cadena de bloques.

En al menos una modalidad, la solicitud de pago incorporará además un registro de “encabezado de transacción” firmado criptográficamente, cuyo registro se requiere incluir para el procesamiento de transacciones atómicas con múltiples firmas de acuerdo con el protocolo de transacción descrito en otra parte en la presente descripción.

(2) Este mensaje de solicitud se transmite luego electrónicamente a través de varios medios posibles a la segunda parte a la que se le solicita el pago. Esta segunda parte, el “Emisor”, puede, en al menos

una modalidad, recibir este mensaje a través de un mensaje de texto móvil o SMS, o a través de un mensaje de correo electrónico, cuyo mensaje contendrá una codificación visual de la solicitud (por ejemplo, como un código QR o código de barras de dos dimensiones) en un adjunto gráfico, o un URI que codifica el mensaje en la base 64 o en otro formato de codificación. El emisor puede recibir además la solicitud de pago a través de un mensaje enviado a través de una red informática conmutada por paquetes a través de tcp/ip, udp u otro protocolo.

En al menos una modalidad, el receptor genera la solicitud de pago mediante el uso de un dispositivo informático tal como un teléfono móvil, teléfono inteligente, tableta, computadora portátil, computadora de escritorio, computadora servidor, dispositivo de punto de venta, o dispositivo de cómputo diseñado específicamente (el dispositivo del Receptor). Este dispositivo Receptor se conectaría a un monitor u otro tipo de pantalla de computadora; cuando se genera el mensaje de solicitud, se codifica como un código QR, código de barras de 2 dimensiones u otra codificación visual, y se muestra en el monitor. Luego, el Emisor usaría una cámara conectada a un dispositivo informático del Emisor para capturar una imagen del mensaje visualmente codificado y decodificar el mensaje analizando la imagen visual capturada.

En al menos una modalidad, el receptor transmite el mensaje de solicitud a través de una antena de radiofrecuencia unida al dispositivo Receptor, cuyo mensaje se recibe a través de una antena de radiofrecuencia en el dispositivo Emisor. La información puede transmitirse a través de comunicación de campo cercano u otro protocolo.

(3) El Emisor recibe el mensaje de solicitud de pago a través de un dispositivo informático que también es un dispositivo de Billetera, que almacena información de clave privada criptográfica relacionada con la cuenta o dirección de la cadena de bloques del Emisor. Este dispositivo de Billetera presenta la información de solicitud de pago al Emisor a través de un monitor, una pantalla gráfica, una terminal basado en texto u otro dispositivo de salida (por ejemplo, al leer los detalles mediante el uso de un sintetizador de voz), solicitando que el emisor autorice que se realice un pago mediante el uso de tokens asignados a la cuenta o dirección de la cadena de bloques del Emisor. El emisor puede proporcionar esta autorización presionando un botón, proporcionando aprobación verbal aceptada por reconocimiento de voz, escribiendo un comando específico mediante el uso de una interfaz de línea de comando, o por otros medios.

(4) Una vez que el Emisor proporcione la autorización, el dispositivo de Billetera genera un registro de transferencia por token que asigna los tokens indicados desde la cuenta o dirección del Emisor a la cuenta o dirección indicada en el mensaje de solicitud. Este registro de transferencia por token se

firmaría mediante el uso de la firma de clave privada que se mantiene dentro del dispositivo de billetera del Emisor. En al menos una modalidad, el registro de transferencia por token firmado estaría acompañado por metadatos adicionales que incluyen información de identidad relevante tal como (en dependencia de la modalidad) nombre, código postal, dirección, número de teléfono, número de ID del gobierno, y/u otra información relevante.

Si el mensaje de solicitud está acompañado por un registro de encabezado de transacción firmado, entonces el registro de transferencia por token incluirá una referencia al registro de encabezado de transacción de acuerdo con el protocolo de transacción descrito en otra parte en la presente descripción.

(5) Tras la firma del registro de transferencia, el dispositivo Emisor transmitiría el registro de transferencia firmado, junto con cualquier metadatos, al dispositivo Receptor. Este registro de transferencia firmado puede enviarse a través de un mensaje de texto móvil o SMS, o a través de un mensaje de correo electrónico, cuyo mensaje incluirá una codificación visual del registro de transferencia firmado (por ejemplo, como un código QR o un código de barras de 2 dimensiones) en un adjunto gráfico, o un URI que codifica el mensaje en la base 64 u otro formato de codificación. El Receptor puede recibir además el registro de transferencia firmado a través de un mensaje enviado a través de una red informática conmutada por paquetes a través de TCP/IP, UDP u otro protocolo. En al menos una modalidad, el dispositivo Emisor se conectaría a un monitor u otro tipo de pantalla de computadora; cuando se genera el registro de transferencia firmado, se codifica como un código QR, código de barras de 2 dimensiones, u otra codificación visual, y se muestra en la pantalla. Luego, el Receptor usaría una cámara conectada al dispositivo Receptor para capturar una imagen del mensaje visualmente codificado y decodificar el registro de transferencia analizando la imagen visual capturada.

En al menos una modalidad, el Emisor transmite el mensaje de solicitud a través de una antena de radiofrecuencia unida al dispositivo Emisor, cuyo mensaje se recibe a través de una antena de radiofrecuencia en el dispositivo Receptor. La información puede transmitirse a través de comunicación de campo cercano u otro protocolo.

(6) El dispositivo Receptor, tras la recepción del registro de transferencia firmado, compartirá los datos de transferencia firmados y cualquier metadatos incluido con un servidor de procesamiento separado, que es en sí mismo un nodo de construcción de bloques especializado, o que está conectado a un nodo de construcción de bloques que controla.

Este servidor de procesamiento mantiene un modelo informático de ponderación de riesgo que es

capaz de asignar una puntuación de ponderación de riesgo al registro de transferencia firmado que ha recibido. Esta puntuación de ponderación de riesgo se basa en la naturaleza y el historial de la cuenta emisora que se ha compartido.

En al menos una modalidad, el modelo de ponderación de riesgo se implementaría mediante el uso de técnicas de aprendizaje automático, que dan forma al modelo mediante el análisis de un cuerpo de datos que implican transacciones similares, o de cualquier otra manera, mediante el análisis de los datos de la cadena de bloques. Para este fin, pueden usarse técnicas de aprendizaje automático tal como redes neuronales, perceptrones multicapa, bosques aleatorios, máquinas de vectores de soporte, árboles de decisión, regresión logística, XGBoost, Máquinas Bolzman Inversas, Bosques de Aislamiento, Autocodificadores de Variación, Probabilidad de Valores Atípicos Locales, Factor de Valores Atípicos Locales, DBSCAN u otros algoritmos de aprendizaje automático.

(7) En dependencia de la puntuación de riesgo asignada, el servidor de procesamiento proporcionará una indicación inmediata a través del dispositivo Receptor al Receptor si el pago ha sido aceptado inmediatamente o si el pago requiere un procesamiento adicional. El dispositivo Receptor transmitirá esta información al Receptor a través de su interfaz del usuario.

Si el servidor de procesamiento indica que el pago ha sido aceptado, esto le indica al Receptor que es seguro entregar productos o mercancía o proporcionar servicios al Emisor, porque hay una alta posibilidad de que el pago se complete en la cadena de bloques. Si el servidor de procesamiento indica que se requiere un procesamiento adicional, entonces el Receptor puede decidir no entregar productos o servicios al Emisor hasta que se haya completado el procesamiento.

(8) El servidor de procesamiento compartirá el registro con la red de cadena de bloques y luego comenzará a construir un bloque que contiene el registro (junto con otros registros similares que haya recibido). Esto puede suceder de manera simultánea con la determinación de la puntuación de ponderación de riesgo, o después de la determinación de esta puntuación .

De acuerdo con al menos una modalidad, si el registro de transferencia apunta a un registro de encabezado de transacción, entonces el servidor de procesamiento construirá un registro de transacción alrededor del registro de transferencia según el protocolo de procesamiento de transacción descrito en otra parte en la presente descripción. Luego, el servidor de procesamiento compartirá ese registro (que encapsulará el registro de transferencia) con la red y construirá un nuevo bloque que contiene ese registro, en lugar de hacerlo con el registro de transferencia.

En el caso de que el modelo de ponderación de riesgo haga que el servidor de procesamiento decida que el pago no se acepta inmediatamente (es decir, que la puntuación de ponderación de riesgo es

demasiado baja), entonces el servidor de procesamiento continuará analizando el consenso de la cadena de bloques. Cuando el servidor de procesamiento detecta que el registro de transferencia o la transacción se ha incluido en un bloque, y que el bloque se ha construido con una cantidad suficiente de bloques nuevos adicionales, de manera que tiene una posibilidad lo suficientemente alta de no ser revocado o anulado, el servidor notificará al dispositivo receptor que el pago es finalmente aceptado.

(9) En al menos una modalidad, el servidor de procesamiento puede controlarse y gestionarse por una persona o entidad separada, el “Procesador”. El Procesador mantiene una cuenta o dirección en la cadena de bloques (la Cuenta del Procesador) que es independiente de la dirección de la cadena de bloques o de la cuenta mantenida por el Receptor (la Cuenta del Receptor).

En virtud de tal arreglo, el mensaje de solicitud de pago indicará que los tokens deben transferirse a la cuenta del Procesador, en lugar de a la cuenta del Receptor.

En al menos una modalidad, si el pago se acepta inmediatamente, significa que el procesador ha garantizado el pago. El procesador transferirá los tokens de la Cuenta del Procesador a la Cuenta del Receptor, por una cantidad igual al monto del pago menos un cargo por procesamiento, al generar un registro de transferencia a tal efecto. Por separado e independientemente, el procesador transmitirá el registro de transferencia recibido del emisor (o la transacción que lo encapsula). Si la primera de estas dos transferencias (de la cuenta del procesador al receptor) caduca y no se acepta en última instancia, el Procesador volverá a generar la transferencia hasta que se acepte, independientemente del resultado de la segunda transferencia (de la cuenta emisora a la cuenta del procesador).

En al menos una modalidad, si el pago no se acepta inmediatamente, significa que el Procesador no ha garantizado el pago. El procesador en este caso generará además un registro de transferencia desde la cuenta del procesador, asignando el monto de pago menos un cargo a la cuenta del receptor. El servidor de procesamiento transmitirá este registro a la cadena de bloques junto con el registro de transferencia y luego comenzará a construir un nuevo bloque con estos registros. Sin embargo, en este caso, si la primera de estas dos transferencias (del procesador al receptor) falla o caduca, solo se retransmitirá si la transferencia original del emisor se confirma y finaliza en la cadena de bloques. En al menos una modalidad, si los registros de transacción se implementan como parte de la modalidad, estos dos registros se combinarán en un único registro de transacción. Este registro de transacción se transmitirá e incluirá en el siguiente bloque que el servidor de procesamiento construye, de manera que ambas transferencias se incluyan en la cadena de bloques, o ninguna de las dos.

### **Definiciones de Tipos de Token y Registro**

Pueden usarse diferentes tipos de registros mediante un sistema y método de cadena de bloques de acuerdo con la presente invención. Estos registros interactúan y manipulan dos tipos de token: tokens nativos y tokens del usuario.

El “**Token Nativo**”, en el contexto de esta aplicación, se refiere a un tipo de token que es inherente a la operación de una implementación de cadena de bloques, y solo puede crearse o destruirse como parte de la operación requerida de la implementación de cadena de bloques en sí. Los cargos y recompensas se denominan preferentemente en el token nativo de acuerdo con al menos una modalidad.

El “**Token del Usuario**”, en el contexto de esta aplicación, es un tipo de token que se ha nombrado y se le ha asignado algún valor agregado a través de un registro creado por el usuario en la cadena de bloques.

El “**Tipo de token derivado**”, en el contexto de esta aplicación, es un tipo de token del usuario que de alguna manera depende de otro tipo de token del usuario, un tipo de token base. Esto permite la delegación de la creación de un nuevo tipo de token y la creación de un nuevo token de manera controlada.

El “**Registro de orden comercial**”, en el contexto de esta aplicación, es un tipo particular de registro. Una orden comercial ofrece intercambiar un tipo de token (ya sea token nativo o token del usuario) por otro tipo de token. La oferta es válida durante un período de tiempo particular, después del cual caduca si no coincide.

El “**Registro de liquidación**”, en el contexto de esta aplicación, es un registro que coincide con dos órdenes comerciales. Un registro de estado de cuenta asigna el margen a una cuenta especificada por el Nodo de Construcción de Bloques.

El “**Registro de vinculación de patrón**”, en el contexto de esta aplicación, se refiere a un tipo particular de registro que puede agregarse a la cadena de bloques. Un registro de vinculación de patrón vincula un patrón con un contrato inteligente particular - si el patrón coincide con algún evento que ocurre dentro del procesamiento de un bloque, con algún registro que aparece dentro de un bloque o con algún estado que se logra a medida que se procesa un bloque, el contrato inteligente vinculado se ejecutará como parte del proceso de construcción de bloque.

El término “**Propuesta**”, en el contexto de esta aplicación, se refiere a una declaración de hecho del mundo real que se ajusta a alguna estructura formal y es cuadrizable y recuperable dentro de un contrato inteligente. Por ejemplo, una declaración con respecto a la temperatura exterior medida en

un momento y ubicación particulares, o el nivel de lluvia experimentada en una ubicación durante un período particular, o la cantidad de huracanes nombrados durante el transcurso de un año calendario. Una propuesta puede ser además una declaración estructurada de un hecho que se refiere a un objeto específico dentro de la cadena de bloques, por ejemplo, haciendo referencia a una cuenta o dirección, o a un tipo de token. Una propuesta tendría un estado de verdad que se decide a través de procesos implementados en el software de cadena de bloques: indeciso, verdadero y falso.

El “**Registro de propuesta**”, en el contexto de esta aplicación, se refiere a un registro que introduce una propuesta a la cadena de bloques e inicia el proceso de decidir el estado de verdad de esa propuesta. Se crea un registro de propuesta con una recompensa de token que se distribuye a las cuentas que han votado en su mayoría cuando se decide la propuesta, qué decisión se toma en un momento definido por el registro de propuesta.

El “**Registro de decisión**”, en el contexto de esta aplicación, se refiere a un registro que pone en riesgo una determinada cantidad de tokens para votar a favor o en contra de una propuesta asociada con un registro de propuesta. Tras la decisión de la propuesta, los tokens en riesgo por la votación de registros en la minoría se distribuyen a las direcciones especificadas por la votación de registros en la mayoría.

### **Tipos de Tokens Nativos y Especificados por el Usuario**

Pueden definirse dos categorías amplias de tipos de token para una cadena de bloques: tipos de token nativo y tipos de token especificados por el usuario. Las instancias de estos dos tipos de tokens, denominados como tokens nativos y tokens del usuario, existen como valores asociados a cuentas o a direcciones dentro del estado global de un sistema de cadena de bloques. Casi todas las implementaciones de cadena de bloques tienen un único tipo de token nativo, que se usa para pagar cargos y generar recompensas que incentivan la participación de los usuarios. Además, dentro de al menos una modalidad de la presente invención, cualquier cantidad de tipos de token especificados por el usuario puede declararse y definirse también. En dependencia de la modalidad específica, las cuentas y direcciones de la cadena de bloques pueden almacenar valor para un solo tipo de token, o pueden almacenar valor para múltiples tipos de token. Las cadenas de bloques construidas de acuerdo con al menos una modalidad de la presente invención pueden ser capaces de contener múltiples tipos de token.

Los tokens nativos pueden acumularse en una cuenta (1) a través de una transferencia desde una cuenta separada que tenga un valor de token nativo que supere el monto de la transferencia; (2) a través de la asignación de una recompensa automática; o (3) a través de la abertura de saldos de

cuenta establecidos en el primer bloque de la cadena cuando la red de cadena de bloques se ponga en marcha por primera vez. Después de que el primer bloque establezca saldos de tokens nativos iniciales, el saldo total agregado de tokens nativos en la cadena de bloques solo puede aumentarse a través del mecanismo de recompensas de la cadena de bloques.

Los tokens del usuario funcionan bajo diferentes restricciones. En al menos una modalidad de la presente invención, se declara y define un nuevo tipo de token del usuario a través de un registro de “génesis”. Este registro de génesis puede asignar además cierta cantidad de estos tokens a una cuenta o dirección, o establecer normas para la operación y el uso del token dentro del sistema de cadena de bloques. El uso futuro de ese tipo de token en particular está restringido de acuerdo con los datos de configuración especificados como parte del registro de génesis. En al menos una modalidad, un registro de “génesis” puede ser un formato y clase de registro únicos - separados y aparte de otras clases de registro. En otras modalidades, puede ser el primer registro de transferencia aceptado en la cadena de bloques que hace referencia a ese tipo de token.

Un tipo de token del usuario se identifica por su etiqueta, que es un número o una cadena de caracteres. No hay dos tipos de token que puedan compartir la misma etiqueta. Cualquier registro que haga referencia a una etiqueta de tipo de token particular solo es válido si cumple con las restricciones establecidas de acuerdo con el historial de la cadena de bloques que precede al registro de génesis de ese tipo de token. En otras palabras, el uso de etiquetas de tipo de token se aplica dentro del proceso de construcción de bloque.

### **Registros de Génesis**

Un registro de Génesis es un tipo de registro residente dentro de un sistema de cadena de bloques. Dentro de la presente invención, los registros de génesis se agregan a la cadena de bloques para declarar y definir nuevos tipos de token del usuario. Los registros de Génesis pueden contener todos o porción de los siguientes datos (no todos los elementos y características son necesarios en cada modalidad):

- Etiqueta - la cadena única que distingue un token de otro y que identifica diferentes saldos de token dentro de una cuenta.
- Valor - el valor del token agregado creado como un resultado de la inclusión del registro de génesis; puede ser 0 en el caso de un registro de génesis que se usa para aumentar la participación asociada con un tipo de token ya declarado, en el caso de un registro que de alguna manera modifica la configuración de un tipo de token existente, o en el caso de que la cuenta local tenga autoridad para generar una cantidad ilimitada de tokens derivados.

- Tipo de token base - opcional. En al menos una modalidad, un tipo derivado hará referencia explícita a su tipo de token base.
- Etiqueta de token derivado - opcional. En al menos una modalidad, las etiquetas de los tokens derivados se restringirán de acuerdo con el campo de la etiqueta de token derivado. Este campo especificará un patrón que la etiqueta debe satisfacer para que el tipo derivado sea válido, qué patrón puede implementarse como una expresión regular o en un idioma similar.
- Cuenta local - la cuenta que recibe los tokens del usuario recientemente generados y, por lo tanto, puede distribuir estos tokens.
- Cuenta emisora - la cuenta que firma criptográficamente el registro de génesis; en el caso de un esquema de emisión de tokens “abierto”, esta es la cuenta local para la apuesta original; en el caso de un esquema de emisión de tokens “autorizado”, esta es la cuenta que tiene autoridad para emitir nuevos tokens con la etiqueta indicada. Esta es la cuenta originaria que firma criptográficamente el registro de génesis con su clave privada.
- Apuesta - en el caso de un esquema de emisión de tokens “abierto”, el valor de token nativo apostado (es decir, bloqueado) en apoyo de este token del usuario; campo no presente bajo el esquema de emisión “autorizado” (ver más abajo).
- Cargo - el monto que se incluirá entre las recompensas de bloque para incentivar la inclusión del registro en la cadena de bloques.
- Fraccional - configuración que controla si se permite que una cuenta tenga valores fraccionarios o decimales, o si solo se pueden transferir valores de números enteros para el tipo de token. Las opciones son “fraccionables” y “totales”; por defecto es “fraccional”. Una alternativa al campo “fracción de unidad” está más abajo.
- Fracción de unidad - configuración que controla cuántos tokens fraccionarios conforman una sola unidad del token. En una modalidad que implementa el campo de Fracción de Unidad en lugar del campo Fraccional, cada token representa la fracción más pequeña posible del activo representado por el token. Cada token representa una fracción igual a  $1 / \text{“fracción de unidad”}$  de una unidad del activo. La cantidad de tokens de “fracción de unidad” equivale a una unidad del activo.
- Suministro - en el caso de un esquema de emisión de tokens “autorizado”, se controla si la cuenta local o la cuenta emisora pueden aumentar arbitrariamente el valor agregado disponible para un token en particular, o si el valor se fija permanentemente al momento de la emisión inicial. Las opciones son “fija”, “ilimitada” y “emisor original”; por defecto es “fijo”.

- Nueva emisión - en el caso de un esquema de emisión de tokens “autorizado”, se controla si la cuenta local o la cuenta emisora original puede emitir tokens *derivados* subordinados - o si una cuenta que es el portador de un token puede transformar un token en nuevos tipos de tokens derivados. Las opciones son “emisor original”, “local” y “portador”; por defecto es “emisor original”.
- Subrutina de autorización - opcional. Una subrutina de contrato inteligente que se ejecutará siempre que se incluyan tokens de este tipo en cualquier registro. En al menos una modalidad, esta subrutina, cuando se invoca, recibiría como entrada todos los registros de datos sujetos a autorización como argumentos, junto con el tipo de token que originalmente se vinculó a la subrutina de autorización. La subrutina devolvería Falso en el caso de que el registro propuesto no esté autorizado, y Verdadero en el caso de que el registro esté autorizado.
- Filtros de autorización - opcional. Una lista de filtros que pueden usarse para excluir cuentas de confirmación de tener, enviar o recibir tokens del tipo dado, o que pueden usarse para requerir que las cuentas que tienen, envían o reciben tokens del tipo dado se ajusten a un perfil determinado. Los filtros consistirían en un patrón que los registros o las cuentas necesitarían cumplir, y harían referencia a los detalles de una cuenta determinada, incluidos los detalles que se adjuntan a una cuenta a través del uso de un registro de propuesta. En al menos una modalidad, cada filtro de autorización puede especificar además qué cuentas de terceros específicas necesitarían haber creado, confirmado o decidido cualquier registro de propuesta usado o al que hace referencia el filtro de autorización.
- Cascada de autorización - opcional. El valor booleano que indica si se debe invocar o no la subrutina de autorización, y/o si se deben invocar filtros de autorización para instancias de tipos de token derivados del token en cuestión. Las opciones son “Verdadero” o “Falso”, en dependencia de si la subrutina de autorización y/o los filtros de autorización de este tipo de token del usuario en particular deben aplicarse o no en el caso de tokens derivados, así como también este tipo de token en sí.

Como se indicó anteriormente, pueden seguirse dos estrategias para definir y declarar los tipos de token especificados por el usuario: un proceso para los tokens “autorizados” y un proceso para los tokens “abiertos”. Los tipos de token del usuario que pueden declararse y definirse en virtud de un esquema “abierto” estarían limitados en cantidad, y cada declaración de nuevo tipo de token requeriría una apuesta (con apuestas suplementarias que también son posibles en al menos una

modalidad). La cantidad específica  $C$  de tipos de token permitidos para declararse y activarse en cualquier momento sería constante para la implementación de la cadena de bloques. Una cuenta que declara y define un tipo de token “abierto” pondría en apuesta (es decir, bloquearía) una determinada cantidad de tokens nativos, cuyos tokens nativos permanecerían bloqueados mientras los tokens del usuario asociados permanezcan activos.

Solo se aceptará un nuevo tipo de token con una apuesta mayor que la más pequeña entre el conjunto global de tipos de token de número  $C$ , aunque cualquier apuesta de valor positivo sería suficiente si se definen menos tipos de token  $C$ . Además, en al menos una modalidad, una apuesta puede estar potencialmente sujeta a rescisión en base a la edad, disminuyendo la apuesta con el tiempo, y permitiendo que los tipos de token latentes o abandonados se eliminen del sistema cuando la apuesta cae por debajo del umbral requerido. Ocasionalmente, es posible que se fijen tokens nativos adicionales en la cuenta “local” del token para mantenerse al día con los valores de apuesta de otras declaraciones de token abiertas; de cualquier otra manera, un tipo de token es responsable de ser retirado de circulación cuando es desplazado por una cantidad suficiente de nuevos registros de génesis.

En al menos una modalidad, un tipo de token que se ha declarado y definido en virtud del proceso “abierto” descrito anteriormente puede evitar las limitaciones de este proceso — que incluye el tope en la cantidad de tokens abiertos y el requisito de apuesta — al convertirse en un tipo de token “autorizado”. Tal conversión ocurriría si se emite un registro de autorización o génesis bien formado y lo firma criptográficamente una cuenta autorizada para iniciar tal conversión, y es, al menos en una modalidad, también refrendada por la cuenta que firmó el registro de génesis que originalmente declaró y definió el tipo de token en cuestión. Alternativamente, un nuevo tipo de token que se define y declara en virtud del proceso “autorizado” descrito más abajo nunca estaría sujeto a estas limitaciones.

### **Proceso de Emisión para Tokens Autorizados**

Por el contrario de los tipos de token “abiertos”, la cantidad de tipos de token “autorizados” es ilimitada. Mientras que la cantidad de posibles tipos de token “autorizados” es ilimitada, sin embargo, cualquier generación de tokens “autorizados” nueva debe proceder a través de una trayectoria de autorización de cuenta a través de una cadena de registros de génesis que se origine con la cuenta que finalmente tenga autoridad para emitir cualquier nuevo tipo de token con cualquier etiqueta: la cuenta “raíz”, que sería la “cuenta local” especificada en el primer registro de génesis dentro de la cadena de bloques. En al menos una modalidad, esta cuenta “raíz” se designaría a través

de la emisión de un registro de génesis mediante el uso de la cadena vacía “” como una etiqueta, que el registro de génesis transmitiría a la cuenta raíz la autoridad final sobre la emisión de todas las demás cuentas.

Los tipos de token autorizados se generarían a través del siguiente proceso de “emisión” que involucra registros de génesis. Según este proceso, en al menos una modalidad, los nuevos tokens solo pueden emitirse, y los nuevos tipos de token solo pueden declararse y definirse, si estos tokens y tipos de token se emiten y declaran y definen por cuentas individuales que están explícitamente autorizadas para hacerlo.

Según el proceso descrito más abajo, dados dos tipos de token, uno puede considerarse un tipo de token “derivado” y el otro un tipo de token “base” si la existencia y la configuración del tipo de token “derivado” de alguna manera depende de la existencia y la configuración del tipo de token “base”. Se considera que un nuevo tipo de token deriva de un tipo de token base si el registro de génesis del tipo derivado de alguna manera hace referencia a un tipo de token base. Salvo los tipos de token declarados y definidos por la cuenta “raíz”, todos los tipos de token autorizados derivan de algún tipo de token base, incluidos los tipos de token base dentro de esta cadena de referencia.

La cuenta emisora indicada en un registro de génesis debe estar autorizada para emitir tokens derivados según la configuración de “nueva emisión” del token base. Adicionalmente, esta cuenta emisora debe firmar criptográficamente el registro de génesis del tipo de token derivado.

En al menos una modalidad, la etiqueta del primer tipo de token (el tipo “derivado”) se construye en la etiqueta del segundo tipo de token (el tipo “base”). En tal modalidad, se considera que un nuevo tipo de token deriva de un tipo de token base solo si la etiqueta de tipo base consiste de los mismos caracteres que forman la porción inicial de la etiqueta de tipo derivado.

En otra modalidad, un tipo de token puede considerarse la “base” de otro tipo de token “derivado” si la cuenta local del tipo de token base se cuenta como la cuenta emisora usada para emitir el tipo de token derivado, y si el registro de génesis del tipo de token derivado hace referencia al tipo de token base. Este sistema alternativo puede ser útil en el caso donde no se coloquen restricciones específicas en las etiquetas que usarán los tokens derivados.

La configuración de “nueva emisión” de un token base determina qué cuenta está autorizada para emitir un nuevo tipo de token derivado de ese token base. La cuenta emisora de un token derivado es la cuenta que está autorizada a emitir nuevos tokens derivados según la configuración de “nueva emisión” de su token base.

### **Configuración de Nueva Emisión de Tokens Autorizados**

En al menos una modalidad de la presente invención, si “nueva emisión” se establece en “emisor original” o un indicador comparable, entonces solo la cuenta emisora especificada en el registro de génesis del tipo de token base puede declarar nuevos tipos derivados que derivan de un tipo de base particular. Por ejemplo, si “*tok*” es un tipo de token base con la nueva configuración de emisión “emisor original”, y “*tok.derived*” es un tipo de token derivado de “*tok*”, entonces el registro de génesis que originalmente emite “*tok.derived*” solo es válido si su “cuenta emisora” es la misma “cuenta emisora” que firmó la cuenta de génesis “*tok*”.

Si la “nueva emisión” se establece en “local” o un indicador comparable, entonces la cuenta local de un token base puede emitir nuevos tokens derivados directamente; la cuenta local del token base sería la cuenta emisora del nuevo tipo de token derivado. Por ejemplo, si “*tok*” es un tipo de token base con la nueva configuración de emisión “local”, entonces el registro de génesis que emite un nuevo tipo de token derivado “*tok.derived*” especificaría una cuenta emisora que es la cuenta local asociada con “*tok*”.

[0002] Si la “nueva emisión” se establece como “portador” o un indicador comparable, entonces cualquier cuenta puede transformar alguna porción de los tokens del usuario que posee en un nuevo tipo derivado; esta acción sería destructiva de los tokens base originales. En este caso, una cuenta con tokens de tipo “*tok*” podría emitir nuevos tokens derivados de tipo “*tok.derived*”, siempre y cuando que la nueva configuración de emisión de “*tok*” sea “portador”.

### **Configuración de Suministro de Tokens Autorizados**

En al menos una modalidad de la presente invención, la configuración de “suministro” especificada en un registro de génesis controla si el valor total agregado disponible para un tipo de token dado puede aumentarse o no, y si es así, qué cuenta está autorizada a aumentar ese suministro.

Si se define un nuevo tipo de token con “suministro” establecido en “ilimitado” o un indicador comparable, entonces la cuenta local para ese tipo de token puede aumentar arbitrariamente el valor agregado de ese token dentro de la cadena de bloques; los registros que usan la cuenta local de ese token como una cuenta de origen (y firmado por la cuenta local) serán válidos incluso cuando excedan el valor de token almacenado en la cuenta local.

Si se define un nuevo tipo de token con “suministro” establecido en “fijo” o un indicador comparable, entonces solo se creará el valor de token especificado en el registro de génesis.

Si se define un nuevo tipo de token con “suministro” establecido en “emisor original” o un indicador comparable, entonces el valor total agregado de los tokens emitidos de ese tipo solo puede ser aumentado por la cuenta emisora de ese registro de transferencia de génesis. En al menos una

modalidad, los registros (tal como registros de transferencias o registros de órdenes comerciales) que usan tal cuenta emisora como una cuenta de origen serán válidos incluso cuando excedan el valor de token almacenado en esa cuenta emisora, siempre y cuando que la autoridad de esa cuenta emisora no esté restringida en sí por un registro de génesis más restrictivo con referencia a su tipo de token base.

Cada vez que se declare y defina un tipo de token derivado, a menos que el tipo de token base tenga su suministro establecido como “ilimitado”, el valor especificado en el registro de génesis del tipo de token derivado extraerá el suministro total disponible para el tipo de token base (es decir, el valor total agregado de todos los tokens autorizados del tipo de token base). De esta manera, el valor total agregado de todos los tokens de un tipo de token base determinado y todos sus tipos derivados no pueden superar el suministro autorizado para ese tipo de token base.

Las siguientes descripciones de escenarios ilustran la interacción de las configuraciones de “nueva emisión” y “suministro”:

(A) El tipo de token *Token\_A* se declara en un nuevo registro de génesis con el campo “nueva emisión” establecido en “local”, el campo “suministro” establecido en “ilimitado” y la cuenta local establecida en *Account\_A*.

En este caso, *Account\_A* puede, en cualquier momento, aumentar el valor total agregado de tokens *Token\_A* en la cadena de bloques. *La Account\_A* puede emitir además en cualquier momento nuevos tipos de tokens derivados que hagan referencia al *Token\_A* como un token base.

(B) *La Account\_A* declara un nuevo token con la etiqueta “*Token\_A.Child\_1*” en un nuevo registro de génesis con el token base “*Token\_A*”, el campo “nueva emisión” establecido en “local”, el campo “suministro” establecido en “emisor original” y la cuenta local establecida en *Account\_B*. El valor de token en el registro de génesis es 10,000.

En este caso, la *Account\_B* puede declarar un nuevo token derivado con el token base “*Token\_A.Child\_1*”, que tiene la etiqueta “*Token\_A.Child\_1.Grandchild*”. Sin ninguna intervención de la *Account\_A*, la *Account\_B* solo puede declarar el valor total del token 10,000 de *Token\_A.Child\_1.Grandchild*, que agota además el saldo del token de *Token\_A.Child\_1*.

La *Account\_A*, sin embargo, puede emitir nuevos tokens de *Token\_A.Child\_1* a voluntad a través de registros de génesis adicionales, reponiendo el saldo de la *Account\_B* de *Token\_A.Child\_1*, y permitiendo que *Account\_B* aumente el valor total agregado de tokens de *Token\_A.Child\_1.Gandild*.

(C) La *Account\_A* declara un nuevo token con la etiqueta “*Token\_A.Child\_2*” en un nuevo registro de génesis que tiene “*Token\_A*” como su tipo de token base, con el campo “nueva emisión”

configurado en “local”, el campo “suministro” establecido en “fijo” y la cuenta local establecida en *Account\_C*. El valor de token en el registro de génesis es 10,000.

En este caso, *Account\_C* puede emitir un nuevo registro de génesis que declara un nuevo tipo de token con la etiqueta “*Token\_A.Child\_2.Grandchild*”, que tiene un tipo de token base “*Token\_A.Chile\_2*”. El valor total máximo de tokens de *Token\_A.Child\_2.Grandchild* que *Account\_C* puede declarar es de 10,000, lo que agota además el saldo de *Token\_A.Child\_2*. Dado que el suministro de “*Token\_A.Child\_2*” es fijo, nunca se declararán más de 10,000 tokens adicionales para *Token\_A.Child\_2* y todos sus tipos de tokens derivados en total.

En al menos una modalidad, “nueva emisión” se establece en “emisor original” y “suministro” se establece en “fijo” por defecto si los ajustes se omiten en el registro de génesis.

En el caso de un nuevo tipo de token derivado, en al menos una modalidad, la “nueva emisión” solo puede establecerse en “local” si el registro de génesis del tipo de token base original establece además la configuración de “nueva emisión” en “local”. Un tipo de token derivado no puede tener una política de emisión más liberal que su tipo de token base.

Además, en el caso de un nuevo tipo de token derivado, en al menos una modalidad, “suministro” solo puede establecerse como “fijo” si el tipo de token base tiene una configuración de “suministro” de “fijo” o “emisor original”. Sin embargo, si el tipo de token base tiene una configuración de “suministro” de “ilimitado”, entonces el nuevo tipo de token derivado puede configurarse en cualquiera de “fijo”, “emisor original” o “ilimitado”. Esto significa que una cuenta que emite un nuevo token derivado solo puede especificar una configuración de “suministro” que sea al menos tan restrictiva como la política de “suministro” a la que está sujeta en sí.

Pueden emitirse además tokens “autorizados” y “abiertos” para contratos inteligentes (donde la dirección del contrato inteligente es la cuenta local en el registro de génesis). Estos contratos inteligentes, implementados como código completo de Turing, pueden transferir y emitir tokens y realizar cualquier otra acción permitida por una cuenta no contractual. Los contratos inteligentes pueden limitar o automatizar la forma en que se emiten o distribuyen los tokens.

### **Contratos de Autorización, Filtros de Autorización y Cascada de Autorización**

Los emisores de tokens y los creadores de nuevos Tipos de token pueden, ocasionalmente, desear restringir el uso y la propiedad de tokens a determinadas entidades y personas calificadas. O bien, es posible que deseen hacer cumplir ciertas normas para la transferencia y adquisición de tokens - por ejemplo, comprar limitando la cantidad total de cuentas que pueden tener un token en particular, o especificando un tiempo mínimo de retención para los tokens, o cualquier otra limitación arbitraria

que pueda colocarse mediante el uso de la lógica informática. Los casos de uso especificados en otra parte en la presente descripción enumeran una variedad de restricciones que pueden ser convenientes para los emisores de tokens.

Para especificar y hacer cumplir estas limitaciones a través del proceso de validación de bloques de la cadena de bloques, en al menos una modalidad de la presente invención, los registros de génesis pueden especificar Contratos de Autorización y Filtros de Autorización que se usan para controlar y limitar transferencias y operaciones de tokens del usuario. Los registros de génesis pueden usarse además para hacer cumplir estas limitaciones en tokens derivados a través del uso de la opción de configuración de Cascada de Autorización especificada en la presente descripción.

Si un registro no está autorizado, entonces no se agregará a la cadena de bloques, o si se agrega a la cadena de bloques, no modificará el estado global del sistema de cadena de bloques, y cualquier asignación de token que contenga no se reflejará en los saldos de token dentro del estado global.

En al menos una modalidad, cualquier registro de datos que haga referencia a un token del usuario particular estaría sujeto a autorización a través de la subrutina de autorización a la que se hace referencia o que se especifica dentro del registro de génesis que define el tipo de ese token. Los registros de génesis, registros de transferencia, registros de órdenes comerciales, registros de liquidación, registros de transacciones que encapsulan otros registros de datos que hacen referencia al token del usuario en cuestión, y cualquier otro registro de datos que haga referencia a tokens del tipo de token en cuestión requerirían tal autorización si se especifica una subrutina de autorización. Debido al costo computacional de la ejecución de la subrutina de autorización, cualquier registro de cadena de bloques sujeto a autorización puede necesitar asignar un cargo del token nativo más alto que el que de cualquier otra manera se requeriría si no estuviera sujeto a autorización. alguna porción del cargo del token nativo asignado con el registro debe usarse para recubrir el costo total de la ejecución de la subrutina de autorización.

Las subrutinas de autorización se invocan y ejecutan durante la validación de bloques, en el punto cuando los nodos de construcción de bloques evalúan si un bloque o segmento de cadena de bloques es válido para construirse. Si una subrutina de autorización devuelve Falso, el registro en cuestión aún se considera válido y cualquier bloque que incluya el registro se considera válido, siempre y cuando que el registro y el bloque sean de cualquier otra manera válidos y estén bien formados de todas las demás maneras. Sin embargo, no se realizarán modificaciones al estado global especificado dentro de este tipo de registro no autorizado, con la excepción del uso del cargo del token nativo para compensar la ejecución de la subrutina de autorización.

Antes de la validación en al menos una modalidad, el nodo de red que se entera por primera vez de un registro ejecutará sus subrutinas de autorización antes de que se creen nuevos bloques y se transmitan a la red de nodo. De esta manera, ningún registro de datos se transmitirá o incluirá en ningún bloque a menos que esté debidamente autorizado. Esto sucedería inmediatamente en el punto en que el usuario que construye un registro de datos lo envía primero a un nodo de red, y solo sucedería si el registro se proporciona al nodo por una fuente de confianza. En al menos una modalidad, solo los registros firmados con la clave privada de la cuenta propia del nodo estarían autorizados de esta manera antes de agregarse a un bloque, para evitar que los nodos se abrumen por la carga computacional de analizar demasiados registros.

Los filtros de autorización, sin embargo, en al menos una modalidad, no están completos en Turing, y no pueden ejecutar ningún procedimiento informático arbitrario. En al menos una modalidad, uno o más filtros se aplicarían a un registro antes de que se transmita a la red de cadena de bloques. Si un registro no cumple con estos filtros, no se puede incorporar en un bloque válido y no se transmitirá a la red en absoluto.

En al menos una modalidad, los filtros de autorización consistirán en definiciones de patrones del tipo encontrado dentro de los registros de vinculación de patrones (descritos más abajo). Por ejemplo, in al menos una modalidad, las definiciones de patrones pueden escribirse mediante el uso de un lenguaje de consulta declarativo, tal como XPath. De manera similar al proceso usado para identificar objetos que cumplen con las definiciones de patrón de los registros de vinculación de patrón, la definición de patrón de un filtro de autorización se aplicará al conjunto de objetos que constituyen y/o están relacionados con el registro que se autoriza. Si los objetos no se ajustan al filtro - si no hay ningún objeto conforme dentro de este conjunto de objetos - entonces el registro no está autorizado. De cualquier otra manera, está autorizado.

En el caso de un registro que funciona con uno o más tokens derivados, en al menos una modalidad, después de que se hayan aplicado los filtros de autorización y las subrutinas de autorización correspondientes a los tipos de token nombrados en el registro, la configuración de los tipos de token de base es examinada por cualquier nodo que esté evaluando y procesando ese registro. Si los tipos de token base especifican que el campo “cascada de autorización” es Verdadero, esto hará que el nodo aplique los filtros de autorización y las subrutinas de autorización adjuntas a los registros de génesis de los tipos de token base.

En al menos una modalidad de la presente invención, la subrutina de autorización de un registro de génesis no devolvería un valor Booleano, sino que devolvería la cantidad de tokens que están

autorizados para transferirse o comercializarse. De esta manera, la subrutina de autorización puede aumentar la cantidad de tokens o disminuir la cantidad de tokens que finalmente se reciben como parte de la transferencia o transacción. De esta manera, podría distribuirse una forma de interés a los titulares de tokens, o podrían imponerse cargos de mantenimiento a los titulares de tokens. Cualquier aumento en el valor del token reduciría la cantidad total de tokens emitidos o que pueden emitirse según el registro de génesis; cualquier disminución en el valor del token aumentaría los tokens disponibles para emitir. Un registro no autorizado provocaría que la subrutina arrojara un número negativo.

Alternativamente, la subrutina de autorización podría devolver una instancia completa del mismo tipo de registro que recibe como entrada. En tal caso, la subrutina de autorización puede transformar cualquier aspecto del registro, no modificarlo o, si el registro es inválido, devolver un código de error de algún tipo.

### **Registro de Transferencia**

Un registro de transferencia mueve un valor de token fuera de una cuenta o dirección originaria y a una cuenta o dirección de destino dentro del estado global de un sistema de cadena de bloques. Un registro de transferencia es aceptable para la inclusión en un nuevo bloque de cadena de bloques si está bien formado, si ha sido firmado por la clave privada de la dirección de origen y si es válido de cualquier otra manera. Para que un registro de transferencia sea válido, por ejemplo, el procesamiento del registro nunca debe dar como resultado un saldo de token negativo, o dar como resultado que el estado global de la cadena de bloques en sí mismo sea inválido.

Los registros de transferencia pueden incorporar los siguientes datos (y posiblemente otros datos también, aunque no todos los elementos y características son necesarios en cada modalidad):

- Cuenta de origen - cuenta que firma el registro de transferencia, del cual se restará el valor del token transferido
- Tipo de token de origen - el tipo de token que se transfiere fuera de la cuenta de origen
- valor del token de origen - el valor en el tipo de token de origen que se transfiere
- Cargo por token de origen - la cantidad que se incluirá entre las recompensas de bloque para incentivar la inclusión
- Tipo de token del cargo por token de origen - el tipo de token que se usa para suministrar el cargo; típicamente, esto no se especificará y se predeterminará al tipo de token nativo
- Cuenta de destino - cuenta para recibir los tokens transferidos

- Tipo de token de destino - el tipo de token recibido como un resultado de la transferencia; si no se especifica, se supone que es el mismo que el tipo de token de origen; si es diferente del tipo de token de origen, entonces un intento sería coincidir con un registro de orden comercial que puede liquidarse para permitir que se produzca la transferencia.
- Valor del token de destino - el valor en los tokens de destino que se asignarían a la cuenta de destino. Este campo es opcional y solo se especifica en el caso de que se especifique un tipo de token de destino diferente del tipo de token de origen.

Si el tipo de token de origen y el tipo de token de destino no son iguales, entonces un registro de transferencia solo puede incluirse en la cadena de bloques si se hace coincidir con un registro de orden comercial y se liquida a través de un registro de liquidación que hace referencia a la orden comercial. En dependencia de la implementación y modalidad, el registro de transferencia y el registro de liquidación pueden ser dos registros separados, o la liquidación puede representarse por ciertos campos en el registro de transferencia en sí.

En cualquier caso, la inclusión de tales registros en la cadena de bloques dará como resultado que la cuenta de destino reciba tokens del tipo de token de destino en un monto igual al valor del token de destino, y en la cuenta que originó el registro de orden comercial que recibe tokens del tipo de token de origen en un monto igual al valor del token de origen. Si un único registro comercial es insuficiente para recubrir el valor total de la transferencia, entonces los registros de liquidación harían referencia a una combinación de órdenes comerciales.

En al menos una modalidad, un nodo de construcción de bloques generaría este registro de liquidación al procesar un registro de transferencia de tipo multitoken. El nodo de construcción de bloques identificaría un registro de orden comercial coincidente — específicamente, el registro de orden comercial que ofrece comprar tokens de origen a cambio de tokens de destino a la tasa de cambio más beneficiosa entre todas las órdenes de transacción conocidas — y construiría un registro de liquidación especializado que haga referencia a esa orden comercial así como también al registro de transferencia. Cualquier margen entre el registro de transferencia y la orden comercial - lo que significa que cualquier diferencia entre el valor del token de destino y el valor del token que puede adquirirse mediante la transacción del valor del token de origen - se acumularía en el nodo de construcción de bloques.

Según esta implementación, el registro de transferencia de tipo multitoken solo se aceptará en la cadena de bloques si está acompañado por un registro de liquidación válido, y si el registro de liquidación apunta a un registro de orden comercial válido ya incorporado en la cadena de bloques

en un bloque anterior (y activo dentro del libro de órdenes), o un registro de orden comercial válido que está incluido en el mismo bloque nuevo. Además, un registro de transferencia de tipo multitoken no puede agregarse a un bloque si no hay un registro de orden comercial disponible con un precio de solicitud inferior al precio de oferta efectivo representado por el valor del token de destino (o viceversa). En otras palabras, debe generarse un margen positivo cuando se procesa una transferencia de tipo multitoken.

En al menos una modalidad, si el valor del token de destino no se especifica al mismo tiempo que se especifica un tipo de token de destino dentro de un registro de transferencia de tipo multitoken, entonces tal registro de transferencia puede coincidir con cualquier orden comercial correspondiente, siempre y cuando que la orden comercial ofrezca (o mejore) el precio de oferta o solicitud más ventajoso especificado en el libro de órdenes permanentes en el momento, excluidas las órdenes de transacciones ya coincidentes.

El valor del cargo incluido con un registro de transferencia puede ser del tipo de token nativo, o puede ser de un tipo especificado por el usuario. En dependencia de la implementación o modalidad, puede o no requerirse que los cargos se denominen en el tipo de token nativo. Una implementación que permite que una cuenta almacene valor para muchos tipos de token permite, entre otras cosas, más flexibilidad en términos de cómo se incluyen los cargos con las transacciones de transferencia. En caso de que se requiera que los cargos se denominen en tokens nativos, pero si los cargos se denominen en un tipo especificado por el usuario, entonces deben coincidir con una orden comercial y liquidarse de una manera similar a la descrita anteriormente para que se acepte el registro de transferencia.

Otra modalidad podría combinar registros de transferencia y registros de génesis en un único tipo de registro. Un nuevo tipo de token se declararía y definiría a través del primer registro de transferencia que contiene ese tipo que se acepta en la cadena de bloques. Por convención, esto actuaría como un registro de “génesis”; todas las normas descritas anteriormente se aplicarían a tal registro, y los campos descritos anteriormente también se incluirían en tal registro, pero se formatearían y presentarían en la cadena de bloques como un registro de transferencia. El primer registro de transferencia para enviar tokens de un tipo determinado actuaría como el registro de génesis para ese tipo de token.

### **Transacciones y Protocolo de Transacciones**

Una transacción, de acuerdo con al menos una modalidad de la presente invención, es una recopilación ordenada de registros y contratos inteligentes que deben agregarse todos a la cadena de

bloques juntos, o ninguno se agregará en absoluto.

Además, si una transacción y sus registros constituyentes y contratos inteligentes se agregan satisfactoriamente a la cadena de bloques, no se realizará ninguna modificación al estado global del sistema de cadena de bloques a menos que todos los registros estén completamente autorizados, y a menos que todos los contratos inteligentes invocados mientras se procesa la transacción finalicen satisfactoriamente. En otras palabras, la transformación hacia el estado global del sistema de cadena de bloques representado por cada registro y contrato inteligente solo continuará si todos los registros y contratos inteligentes dentro de la transacción son capaces de efectuar sus transformaciones individuales satisfactoriamente.

Los registros constituyentes individuales y los contratos inteligentes que comprenden una transacción siempre deben procesarse y ejecutarse en secuencia en el orden especificado por el registro de la transacción en sí mismo - sin registros intermedios o contratos inteligentes, excepto para operaciones de arbitraje u otros registros obligatorios que deben incluirse, o contratos inteligentes que deben invocarse, como una implicación necesaria de la inclusión de estos registros constituyentes en la cadena de bloques.

En al menos una modalidad, un registro de transacción en sí no está firmado; en su lugar, está asegurado por las firmas de los registros individuales que contiene. Los datos de cada uno de los registros constituyentes de la transacción incluyen referencias al registro anterior en la lista, y luego cada registro es firmado individualmente, potencialmente por diferentes partes en dependencia de qué cuenta o dirección originaria se incluya en cada registro. Bajo este arreglo, múltiples partes pueden contribuir con registros separados a la transacción, de manera que los registros de múltiples entidades y las transformaciones de estado global que representa cada registro, puedan ejecutarse de forma atómica. En tal escenario, puede seguirse el siguiente protocolo de transacciones atómicas con múltiples firmas para incluir registros firmados por múltiples partes que controlan múltiples cuentas.

(a) El primer registro de la lista es un registro de “encabezado de transacción” generado por la entidad que finalmente transmitirá la transacción a la red de cadena de bloques. Este registro de encabezado de transacción está firmado por esta primera parte.

(b) La primera parte puede entonces agregar registros adicionales a la transacción al generar nuevos registros, cada registro que haga referencia al registro anterior. La primera parte firmaría entonces cada registro independientemente. Después de agregar registros, la primera porción compartiría entonces la transacción y todos sus registros constituyentes con otras partes, mediante el uso de un método de transmisión de datos capaz de compartir datos entre dispositivos informáticos.

(c) Las partes que deseen agregar registros a la transacción agregarían registros en secuencia, con cada registro que haga referencia al registro anterior, y con cada registro que se firme independientemente mediante el uso de una clave privada que corresponda a su cuenta o dirección originaria. Cualquier entidad que agregue un nuevo registro a la transacción recibiría el valor de referencia necesario para hacer referencia a cualquier registro que esté precediendo en la lista.

En al menos una modalidad, la referencia hecha por un registro dentro de una lista de transacciones sería un hash generado de manera determinista de una representación estandarizada de los datos del registro anterior en la lista. Dado que los datos de cada registro incorporan una referencia al registro anterior en la lista, incluida su firma criptográfica, ningún registro individual en la cadena puede extraerse y procesarse por separado y aparte de su colocación específica en la transacción. Cualquier modificación a cualquier elemento de la lista de registros constituyentes invalidaría toda la transacción.

(d) Después de que todas las partes hayan agregado registros a la lista de transacciones, todo el conjunto de registros se compartirá con la primera parte original. La primera parte agregará un registro más a la lista, un registro de pie de página de la transacción, y luego transmitirá la transacción y su lista de registros a la red de cadena de bloques. El registro del pie de página de la transacción debe estar firmado por la misma cuenta que ha firmado el registro del encabezado de la transacción. La existencia del encabezado de la transacción y del pie de página de la transacción garantizan que la transacción se ejecute en conjunto y que no pueda ser reformada por un actor malicioso.

(e) Cualquier nodo de construcción de bloques que incluya la transacción y sus registros constituyentes en un nuevo bloque primero necesitaría verificar la exactitud y validez de todas las referencias internas dentro de la lista de registros de la transacción, así como también la validez de los registros individuales y sus firmas. No se aceptaría ningún registro constituyente individual para su inclusión a menos que se incluya el registro de transacción mencionado, junto con todos los demás registros constituyentes, que requieren que todos sean válidos y estén firmados.

En al menos una modalidad alternativa, puede construirse un único registro de transacción ómnibus, cuyo registro encapsularía todos los registros constituyentes, sin los registros constituyentes que cada uno necesita ser firmado independientemente. En tal modalidad, si todos los registros constituyentes tienen la misma cuenta o dirección originaria, entonces el registro de transacción solo necesitaría firmarse mediante el uso de la clave privada de esa cuenta emisora. Sin embargo, si los registros constituyentes tienen cuentas o direcciones originarias diferentes, entonces el registro de la transacción tendría que estar firmado por varias cuentas o direcciones; el registro de la transacción

tendría que estar firmado una vez por cada cuenta o dirección mencionada como una cuenta o dirección originaria dentro de sus registros constituyentes.

### **Procedimiento de Determinación de la Propuesta**

Dentro de la presente invención, los registros de propuestas sirven para incorporar información externa fuera de la cadena en la cadena de bloques, para que se usen para diversos fines mediante contratos inteligentes y registros especializados que operan dentro de la cadena de bloques. Los registros de propuestas incorporan esta información en la cadena de bloques y establecen su veracidad, a través de un proceso de decisión que involucra registros de propuestas, registros de asignación de recompensas de respaldo y registros de decisiones.

En al menos una modalidad de la presente invención, la determinación de la propuesta tiene dos fases: (1) la fase inicial de obtención de recompensas, en la que los registros de propuestas y los registros de asignación de recompensas de respaldo asignan la recompensa inicial que se distribuirá a las cuentas que determinen correctamente el resultado de la propuesta al final, y (2) la fase de determinación, en la que los votos se emiten en la forma de asignaciones de token a favor o en contra de la propuesta, como se publica inicialmente en el registro de propuesta original. Estos votos se emiten en los registros de decisiones.

Según este procedimiento, la verdad o falsedad de una propuesta se decide de acuerdo con qué lado está dispuesto y es capaz de arriesgar el valor más alto en tokens. Las recompensas asignadas y distribuidas al lado ganador (y las sanciones incurridas por el lado perdedor) incentivan a los participantes a votar a favor o en contra de una propuesta cuando esos participantes tienen un conocimiento independiente específico de su veracidad o falsedad - o cuando confían mucho en otros participantes que ya han emitido su voto.

Se decide una propuesta - y el proceso de determinación de la propuesta se cierra - después de que ha transcurrido una cantidad de tiempo bien definida, típicamente medida en términos de bloques agregados a la cadena de bloques. En al menos una modalidad de la presente invención, esta cantidad de tiempo se define dentro del registro de propuesta original cuando ese registro se agrega a la cadena de bloques, iniciando el proceso de determinación. Tal registro de propuesta especificaría un momento (es decir, altura del bloque) cuando se aceptarán registros de decisión a la cadena de bloques, y también puede especificar un momento (es decir, altura del bloque) en la que los registros de decisión ya no se aceptarán a la cadena de bloques, momento en el cual se cierra el proceso.

Al menos una modalidad puede decidir si se cierra el proceso de determinación de la propuesta de acuerdo con alguna función que contempla tanto el valor diferencial entre los registros de decisión,

como la cantidad de bloques que ocurren entre los registros de decisión. Si solo se agregan registros de decisión de bajo valor al final del proceso de decisión, entonces sería apropiado reducir la cantidad de tiempo requerido para aprobar (es decir, sería apropiado disminuir la altura del bloque) antes de que se cierre el proceso de determinación. Alternativamente, la inclusión de un registro de decisión de alto valor extendería la cantidad de tiempo (aumentar la altura del bloque) antes de que se cierre el proceso de determinación. Esto evitará que los postores tardíos agobien una propuesta que tiene un cierre duro, y evitará que los postores de poco tiempo prolonguen indefinidamente una propuesta que usa inactividad para determinar el cierre.

Para proporcionar a los participantes en la información del proceso de determinación de la propuesta que es potencialmente útil para decidir si la propuesta es verdadera o falsa, de acuerdo con al menos una modalidad, los registros de decisión y los registros de la propuesta pueden incorporar evidencia, o hacer referencia a evidencia externa, en cuanto a la verdad que están afirmando. Por ejemplo, las URL de IPFS, los DID definidos por W3C o algún tipo de declaración digital verificable independientemente podrían agregarse como datos complementarios dentro de estos registros. Las referencias agregadas a registros de propuestas y registros de decisiones podrían apuntar a archivos de imágenes, archivos PDF o archivos firmados criptográficamente de cualquier tipo. Todos los participantes y posibles participantes dentro del proceso de determinación de la propuesta se beneficiarían de la inclusión de tal información debido a la oportunidad que les brindaría para evaluar tal evidencia para tomar determinaciones independientes en cuanto a la validez de la propuesta.

Una modalidad particular del procedimiento de determinación de la propuesta se describe con referencia a la **Figura 8**. En 810, se acepta un registro de propuesta en la cadena de bloques, que tiene los siguientes campos: (a) una declaración de la propuesta; (b) una asignación de la recompensa; (c) un bloque de decisión en altura  $D$ ; y (d) un bloque de cierre opcional en altura  $C$ . En al menos una modalidad, una declaración de la propuesta también puede contener un campo de caducidad de la propuesta, que es la altura del bloque más allá de la cual la declaración de la propuesta no puede ser considerada como verdadera, independientemente del resultado del procedimiento de determinación de la propuesta.

En 820, para todos los bloques hasta el bloque de decisión a la altura del bloque  $D$ , cualquier cuenta puede asignar recompensas de soporte adicionales, las cuales se agregarán a la recompensa agregada inicial disponible en el bloque  $D+1$ . En 830, tras la incorporación del bloque  $D$ , los registros de decisión inicial se transmiten a través de la red.

En 840, los registros de decisión se agregan al nuevo bloque que se está construyendo. Se invocan

todos los contratos inteligentes vinculados a una definición de patrón de coincidencia, lo que puede provocar potencialmente que se agreguen más registros de decisión. En 850, se decide si el bloque de cierre en altura C ya se ha alcanzado, pero solo si se especificó una altura de bloque de cierre en el registro de propuesta original.

Si se ha alcanzado la altura C del bloque de cierre (SÍ en la etapa 850), entonces la propuesta en 880 se decide agregando registros de decisión en cada lado de la propuesta. Se asignan recompensas y se invocan los contratos vinculados a patrones que coinciden con la decisión. De cualquier otra manera (NO en la etapa 850), se determina en 860 si se aprueba una determinada cantidad de bloques sin que se envíen votos adicionales. Si es así (SÍ en la etapa 860), entonces la propuesta se decide como se describió anteriormente para la etapa 880. De cualquier otra manera (NO en la etapa 860), entonces el número del bloque se incrementa y el proceso de construcción de un nuevo bloque se inicia en la etapa 870. Luego, el proceso regresa a la etapa 840, donde se agregan más registros de decisión al nuevo bloque. Se invocan todos los contratos inteligentes vinculados a una definición de patrón de coincidencia, lo que puede provocar potencialmente que se agreguen más registros de decisión.

En al menos una modalidad, un registro de propuesta puede especificar además si la inclusión de los registros de decisión en referencia a esa propuesta se restringirá a determinadas cuentas o direcciones, si esa restricción se aplicaría a todo el proceso de decisión, o solo al inicio del proceso de decisión (en o cerca de la altura del bloque D), y qué cuentas se permitirían para enviar registros de decisión.

En al menos una modalidad, el sistema y el procedimiento tendrían un registro de propuesta que incluye una lista opcional de cuentas específicas permitidas (o requeridas) para generar el conjunto inicial de registros de decisión correspondientes a ese registro de propuesta. Las cuentas no incluidas en esta lista solo pueden agregar registros de decisión a los bloques subsecuentes después de los primeros registros de decisión producidos cerca del bloque de decisión.

En al menos otra modalidad, esta restricción no solo se aplicaría al conjunto inicial de registros de decisión, sino que se aplicaría además a todos los registros de decisión agregados para una propuesta determinada. Solo las cuentas en esta lista podrían generar registros de decisión que pertenecen al registro de propuesta en cuestión, para todo el período de decisión. Alternativamente, el registro de propuesta puede especificar que una cantidad mínima de cuentas en esta lista debe generar registros de decisión para que el resultado del proceso sea válido.

En al menos una modalidad, un proceso de decisión simplificado sería posible de manera que un

registro de propuesta enumeraría solo una cuenta como permitida para generar registros de decisión, y de manera que el registro de propuesta establecería un bloque de cierre a la altura C que está a una corta distancia del bloque de decisión a la altura D, o incluso es el mismo bloque. Siguiendo este enfoque, un usuario puede generar una propuesta que está destinada a ser decidida por una sola cuenta (pero en cualquier caso una cuenta separada de la cuenta que generó la propuesta en primer lugar).

En al menos una modalidad, un registro de propuesta puede ser además autodeterminado, de manera que la propuesta se marca como ya se ha decidido cuando el registro se incluye en la cadena de bloques. Para habilitar este comportamiento, un campo agregado al registro de la propuesta indicaría que la propuesta es autodecida. Tal registro de propuesta no estaría sujeto al proceso de determinación de la propuesta descrito en la presente descripción, y su propuesta se consideraría automáticamente verdadera. Como un resultado, no es necesario incluir ninguna propuesta de recompensa, más allá de los cargos estándar que se usa para incentivar su inclusión en un bloque.

### **Asignación de la Recompensa de la Propuesta**

El proceso de distribución de recompensas de la propuesta a cuentas de votación exitosas de acuerdo con al menos una modalidad de la presente invención se describe con referencia a la **Figura 9**.

Para determinar el lado ganador de la propuesta (Verdadero de Falso) de acuerdo con al menos una modalidad, se agrega el valor de los tokens que votan para cada uno de los dos lados. Luego, se determina que el lado con el valor agregado más alto es el lado “ganador”. Los votantes ganadores pueden recibir un reembolso completo de sus tokens, mientras que los tokens de todos los votantes perdedores se agregan al grupo de recompensa asignado a los votantes ganadores, según al menos una modalidad de la presente invención.

En 910, la recompensa asignada para el bloque D+1,  $r_{D+1}$  es la recompensa original del registro de la propuesta más el total de todas las recompensas de soporte. A partir de 920, la recompensa de la votación  $r_{D+N}$  se distribuye a las cuentas de votación exitosas en el bloque D+N, y en los bloques subsecuentes. La recompensa de la votación inicial para el bloque D+N se establecería en el bloque anterior, con la recompensa donde  $N = 1$  es  $r_{D+1}$ .

En 930, se determina si el bloque con altura D+N es o no el último bloque que contiene los votos ganadores con respecto a la propuesta. Si es así (Sí en la etapa 930), el nodo de construcción de bloques, en la etapa 940, agregará el valor de token de todos los votos perdedores en los bloques D+N y superiores y agregará el resultado a la recompensa actual en el bloque D+N, cuyo valor  $r_{D+N}$  se había establecido previamente en el bloque D+N-1. Subsecuentemente, en la etapa 950, la

totalidad de esta recompensa combinada se distribuirá a las cuentas de votación ganadoras en el bloque  $D+N$ , proporcionalmente según los valores de token que pone en riesgo cada cuenta de votación ganadora, y el proceso finalizará.

De cualquier otra manera (NO en la etapa 930), donde el bloque  $D+N$  no es el último bloque en contener votos ganadores con respecto a la propuesta, el sistema procede a la etapa 960, donde se determina si el bloque  $D+N$  contiene algún voto a favor del lado ganador de la propuesta. Si lo hace, entonces en 970, se distribuirá una porción  $f(r_{D+N})$  de la recompensa de decisión en el bloque  $D+N$  a las cuentas que voten con el lado ganador en el bloque  $D+N$ , proporcionalmente según el valor del token de cada cuenta. Luego, la porción restante de la recompensa  $r_{D+N} - f(r_{D+N})$  se coloca en reserva, para usarse para aumentar la recompensa disponible en el siguiente bloque, bloque  $D+N+1$ . La recompensa disponible para distribución dentro de este siguiente bloque,  $r_{D+N+1}$ , será igual a la recompensa restante  $r_{D+N} - f(r_{D+N})$  más una porción del valor del token de los votos perdedores realizados en el bloque  $D+N$ , como se indica en la etapa 990. Luego, se incrementa el número del bloque ( $N = N + 1$ ) en 985 y el procesamiento regresa a la etapa 920 para el siguiente bloque.

Si en la etapa 960, se determina que el bloque  $D+N$  no contiene votos a favor del lado ganador de la propuesta (NO en la etapa 960), entonces no se asignan recompensas en el bloque  $D+N$ , y la totalidad de la recompensa no usada  $r_{D+N}$  se coloca en reserva, para usarse para aumentar la recompensa disponible para el siguiente bloque, bloque  $D+N+1$ . En este caso, la recompensa  $r_{D+N+1}$  disponible para asignarse en el bloque  $D+N+1$  equivale a la recompensa total no asignada  $r_{D+N}$ , más el valor del token de los votos perdedores realizados en el bloque  $D+N$ , como se indica en la etapa 980. Luego, se incrementa el número del bloque ( $N = N + 1$ ) en 985 y el procesamiento continúa con la etapa 920 para el siguiente bloque.

En al menos otra modalidad (no se ilustra), la relación de recompensas en forma de tokens en cada bloque puede ser una función tanto de la puntualidad del voto como del monto del token colocado en riesgo. Un monto temprano pero menor puesto en riesgo recibirá más recompensa que un monto tardío y mayor puesto en riesgo.

### **Declaraciones de la Propuesta**

Cuando el proceso de determinación de la propuesta finaliza satisfactoriamente con una decisión resultante de que una propuesta es “verdadera”, entonces la declaración de la propuesta adjunta al registro de propuesta original se agrega al estado global como un punto de datos que puede usarse en todo el sistema de cadena de bloques para guiar el flujo de lógica o modificar de cualquier otra manera el comportamiento del sistema.

Las declaraciones de la propuesta son declaraciones lógicas estructuradas que (a) pueden adjuntarse a otro objeto dentro del estado global del sistema de cadena de bloques; (b) pueden adjuntarse a cadenas arbitrarias que representan conceptos del mundo real; o (c) pueden ser independientes sin adjuntarse a ningún objeto en particular. En al menos una modalidad, una declaración de la propuesta que se adjunta a una cuenta agregaría campos adicionales a esa cuenta, a los que puede accederse a través de referencia a esa cuenta. De manera similar, en al menos una modalidad, una declaración de la propuesta que se adjunta a un tipo de token agregaría campos adicionales a ese tipo de token, a los que puede accederse a través de referencia a ese tipo de token, o mediante referencia a tokens de ese tipo.

Las declaraciones de la propuesta pueden agregar uno o más campos a los objetos a los que se adjuntan, o pueden hacer afirmaciones lógicas. En al menos una modalidad, las declaraciones de la propuesta incorporarían una o más declaraciones Booleanas que tienen cada una un valor, un operador Booleano, y una variable, campo o característica del objeto que describe la declaración Booleana. Tales declaraciones Booleanas pueden ser consultables, pueden coincidir con una definición de patrón, o pueden ser comprobables dentro de la lógica de un contrato inteligente.

En al menos una modalidad, en el caso de que una declaración de la propuesta no se adhiera a ningún objeto específico dentro del estado global del sistema de cadena de bloques, la propuesta en sí misma existe por sí misma en una estructura de datos que puede consultarse junto con otras declaraciones independientes de la propuesta.

Para reducir la hinchazón del sistema, la complejidad de los datos o la carga computacional, la manera en que se almacenan los datos de la declaración de la propuesta puede depender del proceso mediante el cual se decide la propuesta y qué cuentas participan en el proceso de determinación. En al menos una modalidad, si la cuenta que genera y firma un registro de propuesta se relaciona con el objeto al que se adjunta la declaración de la propuesta, entonces los datos de la declaración de la propuesta se almacenarán con ese objeto y serán recuperables con ese objeto. De cualquier otra manera, los datos de la declaración de la propuesta se almacenarán por separado y podrán recuperarse por separado del objeto al que se adjuntan.

En al menos una modalidad, si se genera y firma un registro de propuesta por la cuenta a la que se adjunta su declaración de la propuesta, en ese caso los datos de la declaración de la propuesta se almacenarán junto con esa cuenta en el alcance global. En al menos otra modalidad, si un registro de propuesta es generado y firmado por la cuenta local o cuenta emisora de un tipo de token, entonces la declaración de la propuesta de ese registro puede almacenarse junto con cualquier cuenta separada

a la que se adjunta, si esa cuenta separada tiene tokens de ese tipo. En otras modalidades, una declaración de la propuesta adjunta a un tipo de token se almacenaría junto con la definición y configuración de ese tipo de token, siempre y cuando que el registro de la propuesta que introduce la declaración sea la cuenta local o la cuenta emisora del tipo de token, o de los tipos de token de base o de antecesor del tipo de token.

En los tres casos, un intento de recuperar el objeto adjunto, ya sea una definición de cuenta o tipo de token, recuperará además los datos de la declaración de la propuesta que se adjunta al objeto. Sin embargo, en al menos una modalidad, una declaración de la propuesta que se adjunta a una cuenta o tipo de transacción no se almacenará y recuperará con la cuenta o tipo de transacción, a menos que el registro de propuesta que introdujo esa declaración de la propuesta se haya generado y firmado por cuentas relacionadas.

### **Registros de Vinculación de Patrones**

En al menos una modalidad de la presente invención, los registros de vinculación de patrones vinculan los contratos inteligentes ejecutables dentro de una cadena de bloques con “definiciones de patrones”. Cada tal registro puede contener todos o algunos de los siguientes campos: una cuenta o dirección originaria, una asignación de tokens usada para suministrar cargos y gas para compensar el procesamiento (ya sea como un campo combinado, o como dos campos separados), una fecha de vencimiento o altura del bloque, una frecuencia de ejecución, un bloque de inicio, un contrato inteligente o una referencia a un contrato inteligente, y una definición de patrón. Las definiciones de patrones pretenden coincidir con los (a) registros que podrían agregarse a la cadena de bloques en bloques futuros, (b) transformaciones particulares del estado del sistema, o (c) eventos que ocurren dentro del curso de un bloque que se está formando o validando. En al menos una modalidad, las definiciones de patrones se escriben mediante el uso de un lenguaje de consulta declarativo, tal como XPath.

Cuando tales definiciones de patrones se evalúan en relación con los registros o eventos de un bloque de cadena de bloques particular, si se encuentra que coinciden con uno o más de tales registros, se encuentra que coinciden con una o más transformaciones del estado del sistema realizadas por tales registros, o se encuentra que coinciden con uno o más de tales eventos, entonces se invocan los contratos inteligentes vinculados a esos patrones.

Tal evaluación se lleva a cabo por el nodo de construcción de bloques que está construyendo el bloque en cuestión, o el siguiente bloque, en dependencia de la modalidad. El nodo de construcción de bloques estará al tanto de los registros de vinculación de patrones todavía viables que se agregaron

a la cadena de bloques en algunos bloques anteriores. El nodo se iterará sobre ese conjunto de registros de vinculación de patrones, haciendo coincidir sus patrones con los registros, transformaciones de estado y eventos que se han incluido con el bloque actual, o el bloque más recientemente creado. Al encontrar una coincidencia, el nodo ejecutará el contrato inteligente al que hace referencia ese registro de vinculación.

Por ejemplo, podría definirse un patrón particular que coincida con los registros de transferencia que transfieren cantidades superiores a algún valor especificado, y que también envían tokens a alguna cuenta especificada. Un registro de vinculación de patrón que contiene tal definición de patrón provocaría que cualquier contrato inteligente vinculado se especifique en el registro de vinculación que se ejecutará en caso de que se agregue un registro de transferencia coincidente a la cadena de bloques.

Alternativamente, un registro de vinculación de patrón puede coincidir con un cambio de estado que ocurre durante el transcurso de la evaluación de un bloque. Por ejemplo, el contrato inteligente de un registro de vinculación podría ejecutarse en caso de que el saldo de alguna cuenta especificada supere algún umbral especificado.

A manera de ejemplo, en al menos una modalidad que usa una sintaxis tipo XPath para definiciones de patrones, la definición de patrón que coincide con un registro de transferencia con un valor mayor que 100 tokens puede implementarse como una cadena de la forma `‘/transfers[value>100]’`, cuya definición de patrón indica que se seleccionan objetos de transferencia que tienen un valor mayor que 100 dentro de un bloque. La cadena de definición de patrón `“/transfers[value>100 y source_token_type=“RUC”]”` indicaría que los registros de transferencia que transfieren más de 100 tokens de RUC coincidirían. En un caso más complejo, una transferencia enviada por una cuenta que ha adjuntado una propuesta de “jurisdicción” a sí misma podría coincidir con la cadena de definición de patrón `‘/transfers[account/jurisdiction!=“USA.NY”]’`. Una transición de estado de un saldo de cuenta que cae por debajo de 200 tokens RUC podría coincidir con la cadena de definición de patrón `‘/accounts/07201969/tokens/RUC[valor<200]’`, que coincidiría si el nuevo saldo RUC de la cuenta 07201969 es inferior a 200 después de que el bloque se haya procesado, pero era igual o superior a 200 antes de que se procesara el bloque.

En al menos una modalidad de este tipo de implementación de definición de patrón, el objeto raíz de la expresión puede referirse a un tipo de registro, o puede referirse a un objeto dentro del estado global. Si se refiere a un tipo de registro, entonces los registros dentro de un bloque coincidirían. Si se refiere a un objeto en el estado global, entonces una transición de estado que se produce como un

resultado del procesamiento de un bloque coincidirían. Si un registro coincide con la definición del patrón, entonces ese registro se pasa como un argumento a la subrutina de contrato inteligente vinculado. Si un objeto en el estado global coincide, entonces el registro que personifica el cambio en el estado global (es decir, la transición de estado) que ocurre dentro de ese bloque se pasa al contrato inteligente vinculado.

Un registro de vinculación de patrón puede necesitar incluir o asignar cargos del token nativo para incentivar su inclusión en un nuevo bloque, y/o pagar la ejecución del contrato inteligente al que hace referencia. Debido a que tal asignación de cargos obtendría esos cargos de una cuenta específica, el registro de vinculación de patrón necesitaría firmarse criptográficamente por la clave privada de esa cuenta.

Si se requieren cargos para incentivar un nodo de construcción de bloques para incluir un registro de vinculación de patrón en un bloque, entonces la viabilidad de esa vinculación puede limitarse por la cantidad de cargos de token nativo que incluye el registro de vinculación. Por ejemplo, un registro de vinculación de patrón puede especificar que debe evaluarse mediante nodos de construcción de bloques para el número  $G$  de bloques, para los cuales el esfuerzo computacional, el registro asigna  $X$  tokens en cargos. Cada bloque para el cual se evalúa el registro de vinculación de patrón reclamará que alguna porción de estos  $X$  tokens de cargos se asignen a su propia recompensa de bloque.

La asignación de una porción de los tokens de cargos  $X$  a la recompensa de bloque para un bloque determinado está determinada por alguna función  $f(X, G, N)$ , donde  $N$  es el bloque actual. En un caso simple, de manera que  $f(X, G, N) = X/G$ , cada bloque en el que se evalúa el patrón obtendría una porción igual de los cargos asignados. El bloque que primero incorpora el registro de vinculación de patrón en la cadena de bloques se incentiva a hacerlo si la relación  $X/G$  es alta.

Además de los tokens de cargos, cada registro de vinculación de patrón también debe asignar suficientes tokens nativos para pagar la ejecución del contrato vinculado, es decir, tokens de “gas”. Debido al “problema de detención” (diseñado por Turing y otros), no se sabe cuántas etapas informáticas se requerirán para que un contrato inteligente vinculado complete su ejecución. Cada etapa de ejecución debe consumir cierta cantidad de tokens nativos, y la cantidad total de etapas computacionales tomadas por un contrato inteligente dicta el costo total de su invocación. La ejecución de un contrato inteligente de larga duración o de cualquier otra manera pesado desde el punto de vista informático puede agotar los tokens de “gas” de suministro asignados a través del registro de vinculación para este fin. Si el registro de vinculación de patrón gasta todos sus tokens

de gas antes de que haya caducado, tal registro de vinculación perderá su viabilidad antes de su caducidad programada.

Después de incorporar un registro de vinculación de patrón a la cadena, debe evaluarse para cada bloque hasta que caduque, o hasta que se quede sin cargos de token nativo para pagar la evaluación de su patrón en relación con otros registros de bloque, o tokens de “gas” para pagar la ejecución de su contrato inteligente vinculado. Después de que un nodo de construcción de bloques haya ejecutado un contrato inteligente vinculado, el nodo de construcción de bloques agregará un registro de recompensa de vinculación al bloque, que el registro provoca los tokens de “gas” que se gastaron a través de la ejecución del contrato inteligente vinculado para asignarse a la recompensa total del bloque.

En al menos algunas modalidades, un registro de vinculación de patrón puede especificar asignaciones de tokens nativos separados para “gas” y “cargos”. Sin embargo, en otras implementaciones, una única asignación de tokens nativos puede representar todos los “gases” y “cargos” en total. A medida que se consumen los cargos y el gas, ambos extraerían la misma cantidad fungible. En algunas modalidades, un “cargo máximo” y/o un “gas máximo” puede especificarse por registro de vinculación de patrón, que establecería un límite superior para qué porción de los tokens asignados al registro de vinculación de patrón puede consumirse para cada bloque. Una subasta estilo holandés similar en el método al que se describe más abajo relativa a la negociación en cadena puede seguirse en al menos una modalidad para determinar qué registros de vinculación de patrón incorporar en la cadena de bloques y procesar para cada bloque.

El procesamiento de registros de vinculación en una cadena de bloques de acuerdo con al menos una modalidad de la presente invención se describe con referencia a la **Figura 10**.

En 1010, el nodo autor publica un registro de “vinculación de patrón” a la red de nodo, incluida una asignación de cargos. Luego, en 1020, el registro propuesto de “vinculación de patrón” se propaga en toda la red y espera la inclusión en un nuevo bloque en la cadena de bloques.

En 1030, se determina si hay suficientes cargos de inclusión en bloque asignadas al registro a través del registro propuesto de “vinculación de patrón”. Si es así, (SÍ en la etapa 1030), entonces el registro de “vinculación de patrón” se agrega al nuevo bloque, y el nuevo bloque se propaga y acepta en la cadena de bloques a través del consenso en la etapa 1040.

De cualquier otra manera, (NO en la etapa 1030), el “registro de vinculación de patrón” propuesto se abandona o demora para la inclusión en la etapa 1035, y el procesamiento regresa a la etapa 1010. En la etapa 1050, después de la inclusión de registros de vinculación de patrones en un nuevo bloque

y la aceptación de ese nuevo bloque en la cadena de bloques, sus registros se comparan con definiciones de patrones de registros de vinculación todavía viables (es decir, registros de vinculación que no caducaron y no han agotado su asignación de cargos). Si se determina en la etapa 1060 que la definición del patrón de cualquier registro de vinculación coincide con cualquier evento o registro del bloque (SÍ en la etapa 1060), entonces en la etapa 1070 se ejecutan los contratos inteligentes asociados con esos registros de vinculación. Se deduce un cargo por bloqueo de la asignación de cargos de cada registro de vinculación, y cada contrato inteligente vinculado se procesa hasta que se rescinda, o hasta que se agoten los tokens de “gas” asignados a su registro de vinculación.

En al menos una modalidad, los resultados de estas operaciones se incorporan en el siguiente bloque que se agrega a la cadena de bloques; alguna porción de estas operaciones hasta un determinado umbral debe realizarse para que ese bloque se considere válido. Alternativamente, en al menos otra modalidad, estas operaciones se realizan como un nuevo bloque que se está construyendo, de manera que los resultados se incorporan en el mismo bloque que incluye los registros, transformaciones o eventos que coinciden con los registros de vinculación.

Después de la ejecución del contrato inteligente asociado con un registro de vinculación de patrón particular, se realiza una evaluación en la etapa 1080 para determinar si los suficientes tokens de “gas” y cargos permanecen dentro de la asignación de tokens de ese registro de vinculación, y si el registro de vinculación ha caducado o de cualquier otra manera no. Si el registro de vinculación no está caducado y tiene suficientes tokens para continuar (SÍ en la etapa 1080), entonces ese registro de vinculación de patrón se considera viable y se agrega nuevamente al grupo de registros de vinculación viables, para continuar siendo evaluado contra los bloques subsecuentes en la etapa 1085.

De cualquier otra manera, el registro de vinculación de patrón se elimina del grupo (NO en 1080). El procesamiento regresa a la etapa 1010 para la creación e inclusión de nuevos registros de vinculación de patrones.

Si se determina en la etapa 1060 que la definición de patrón de cualquier registro de vinculación no coincide con ningún evento o registro del bloque (NO en la etapa 1060), entonces, en la etapa 1085, el registro de “vinculación de patrón” se agrega de nuevo al grupo de registros de vinculación de patrón viable, y continúa siendo evaluado contra los bloques subsecuentes. Luego, el proceso regresa a la etapa 1050 para repetir la comparación de registros de bloque, transformaciones y eventos con definiciones de patrones de registros de vinculación todavía viables, para el siguiente bloque.

En al menos una modalidad alternativa, sin embargo, un cargo de bloque, sin embargo, se deduciría de la asignación de token de cada registro de vinculación de patrón que se evalúa, para cada bloque dentro del cual se evalúa, independientemente de si el registro de vinculación coincide o no con cualquier registro, transformaciones o eventos dentro del bloque. En tal caso, el registro de vinculación de patrones deberá evaluarse para determinar la viabilidad (es decir, si tiene suficientes tokens en su asignación de cargos restante) independientemente de si encuentra o no una coincidencia, y, por lo tanto, puede eliminarse del grupo de registros de vinculación independientemente de si encuentra o no una coincidencia si se ha quedado sin tokens de cargos.

De manera similar, en al menos una modalidad, los registros de vinculación del patrón especificarán un campo de caducidad, que puede especificarse como una altura del bloque a la que caducará el patrón de vinculación. En tal caso nuevamente, el registro de vinculación de patrón deberá evaluarse en cuanto a si ha caducado o no, independientemente de si encuentra o no una coincidencia, y, por lo tanto, puede eliminarse del grupo de registros de vinculación independientemente de si encuentra o no una coincidencia si ha caducado.

En al menos una modalidad, la frecuencia a la cual una definición de patrón puede evaluarse en la etapa 1050 se controlaría mediante una configuración de frecuencia de ejecución especificada con el registro de vinculación de patrón. Un registro de vinculación de patrón puede definir una frecuencia tal que se evalúa en cada bloque (por defecto), o puede especificar una frecuencia alternativa, hasta un determinado límite, de manera que se toma del grupo y se evalúa cada cinco bloques, o cada 100 bloques, o lo que sea. El primer bloque en el que se evaluaría la definición del patrón se establecería mediante un campo de “bloque inicial” dentro del registro de vinculación de patrón, si se especifica tal campo.

En al menos una modalidad, en el caso de un registro de vinculación de patrón que tiene una frecuencia de ejecución que es menos de una vez por bloque, la definición de patrón se comparará con, y trabajará para filtrar, todos los registros y todas las transformaciones del estado global que han ocurrido dentro de los bloques creados desde que se evaluó previamente la definición de patrón.

### **Proceso de Negociación en Cadena**

Los tokens de un tipo mantenidos por una cuenta pueden intercambiarse por tokens de otro tipo mantenidos por otra cuenta, a través de un proceso que usa registros de órdenes comerciales y registros de liquidación agregados a una cadena de bloques.

El diagrama de flujo en la **Figura 11** ilustra un proceso de al menos una modalidad de la presente invención mediante la cual la negociación de tokens puede implementarse dentro de un software

peer-to-peer de cadena de bloques en red mediante el uso de registros de órdenes comerciales y registros de liquidación. Las órdenes comerciales se transmiten a través de la red en la forma de registros de órdenes comerciales, se incorporan en un libro de órdenes en línea y luego son emparejadas por un nodo de construcción de bloques dentro de la red, con órdenes emparejadas que se registran y liquidan de manera simultánea en la cadena de bloques a través de registros de liquidación.

El libro de órdenes se describe aquí como que se ha “sintetizado” porque se crea mediante el ensamblaje y la priorización de las órdenes de transmisión reciente junto con las órdenes permanentes en cadena que se incorporaron en bloques anteriores en la cadena de bloques, que aún no caducaron y que aún no coinciden. Después de la síntesis, el libro de órdenes puede residir en la memoria local del nodo sin sincronizarse con el resto de la red de cadena de bloques. El libro de órdenes sintetizado difiere del libro de órdenes permanente del sistema de cadena de bloques porque es local al nodo de construcción de bloques, refleja solo la información actualmente disponible para ese nodo y no se ha incorporado al estado global del sistema de cadena de bloques.

En al menos una modalidad, es la prerrogativa del nodo de construcción de bloques con respecto a qué registros de órdenes comerciales recientemente transmitidos se agregan al libro de órdenes sintetizado. El nodo puede tener en cuenta todas las órdenes de transmisión recientes de los que se ha enterado (siempre y cuando que aún no hayan caducado). No hay un límite formal en cuanto a la cantidad de estas órdenes que pueden existir.

Sin embargo, la cantidad de órdenes que se hacen permanentes al incluirse en bloques agregados a la cadena de bloques puede estar sujeta a las limitaciones de tamaño del libro de órdenes permanentes. En al menos una modalidad, más allá de un cierto número umbral de órdenes permanentes en cadena, los nuevos registros de órdenes comerciales no pueden agregarse al libro de órdenes permanentes a menos que las órdenes anteriores caduquen o coincidan completamente y se eliminen del libro de órdenes.

En al menos una modalidad, un número finito de órdenes de compra y venta puede agregarse al libro de órdenes en cadena permanente a la vez. Las nuevas órdenes comerciales solo pueden agregarse si las órdenes comerciales antiguas caducaron para hacer lugar para nuevas órdenes comerciales, o si las órdenes comerciales antiguas han coincidido completamente y eliminado del libro de órdenes, de manera que no se superen las limitaciones de tamaño del libro de órdenes permanente. Alternativamente, en al menos otra modalidad, las órdenes permanentes en cadena pueden desplazarse del libro de órdenes permanentes por nuevos registros de órdenes comerciales con

cantidades de oferta más altos o montos de solicitud más bajos que los que se encuentran dentro del libro de órdenes permanentes. Un bloque no es válido si se agregan demasiados registros de órdenes comerciales.

En cualquier caso, en al menos una modalidad, cualquier orden comercial no permanente recientemente transmitida puede coincidir en una transacción por el nodo de construcción de bloques si tal transacción produce un margen menor que el que se tendría al coincidir con una orden comercial permanente comparable ya incorporada en un bloque en la cadena de bloques. En cualquier transacción coincidente realizada por un registro de liquidación, se deben usar las órdenes comerciales de oferta más alta o de menor demanda dentro del libro de órdenes permanentes, a menos que una orden comercial de oferta más alta o de menor demanda esté disponible entre las órdenes comerciales de transmisión reciente. Siguiendo este enfoque, se evita que un nodo de construcción de bloques simplemente ignore el libro de órdenes y liquide las órdenes comerciales — órdenes comerciales potencialmente inferiores — que son conocidas solo por ese nodo de construcción de bloques.

El nodo de construcción de bloques es responsable de hacer coincidir las órdenes, que se guardan en memoria a través de registros de liquidación generados por el nodo de construcción de bloques en sí. El registro de liquidación hace referencia a todas las órdenes coincidentes, todas las cuales se eliminan del libro de órdenes sintetizados, y del libro de órdenes permanentes si también se encuentran allí. Toda orden comercial de transmisión reciente que coincida a través de una orden de liquidación debe agregarse al nuevo bloque junto con la orden de liquidación.

El nodo de construcción de bloques está incentivado a maximizar su propia recompensa de token, y probablemente intentará crear coincidencias que maximicen los cargos al elegir las operaciones que tengan los cargos más altos, todo lo demás igual. Adicionalmente, el nodo se beneficiará de cualquier margen disponible entre órdenes que coincidan; el nodo dirigirá el margen a cualquier cuenta que esté configurada para usar, que convencionalmente será la cuenta de su propietario. Para maximizar el margen, el nodo típicamente coincidirá con las órdenes comerciales que tienen las tasas de cambio más ventajosas.

El nodo también puede aprovechar las oportunidades de arbitraje dentro del libro de órdenes sintetizados al hacer coincidir cadenas de órdenes de diferentes activos o tipos, y al redirigir las ganancias a su propia cuenta de activos. Por ejemplo, si las ofertas para intercambiar A por B no existen, una oferta de B por A puede hacer coincidirse con las ofertas para A por C y C por B. Un nodo puede hacer tales cadenas coincidir para satisfacer una orden comercial, o para permitir que

una orden de transferencia de tipo multitoken se incorpore en un bloque.

En al menos una modalidad, el nodo de construcción de bloques también puede coincidir con órdenes comerciales que tienen un margen negativo (es decir, pares de órdenes comerciales donde la oferta es menor que la demanda), siempre y cuando que el nodo de construcción de bloques pueda recubrir la diferencia mediante el uso de sus propios tokens, o tokens de cargo. Puede incentivarse a un nodo de construcción de bloques a hacer esto si el cargo potencial para liquidar la transacción es lo suficientemente grande para recubrir la diferencia, o si la coincidencia es una parte de una cadena de arbitraje más grande que de cualquier otra manera es rentable.

El nodo de construcción de bloques puede elegir qué órdenes comerciales de transmisión reciente no completadas se preservan en el nuevo bloque - y de esta manera se agregan a los libros de órdenes permanentes, y qué órdenes no completadas no se incluyen. Esta opción está incentivada por los tokens de cargos incluidos en cada registro de orden comercial. La cantidad de tiempo que un registro de orden comercial figura en el libro de órdenes permanentes se determina por su caducidad especificada, que puede expresarse en términos de altura del bloque o en términos de una fecha y hora específicas. Además, el registro deberá proporcionar suficientes cargos para seguir siendo viable dentro del libro de órdenes para el período especificado.

En al menos una modalidad, el proceso de negociación en cadena descrito aquí puede combinarse con el proceso de autorización de token de la presente invención en una o más modalidades de manera que cada registro de orden comercial está sujeto a cualquier requisito de autorización configurado en los registros de génesis de los tipos de token del usuario a los que hace referencia. En tal modalidad, los registros de órdenes comerciales solo serían viables y solo se incluirían en el libro de órdenes permanentes (o en cualquier libro de órdenes sintetizado) si pasan la subrutina de autorización y los filtros de autorización que están vinculados a los tipos de token a los que hacen referencia esas órdenes comerciales. En al menos una modalidad, una falla en la etapa de la subrutina de autorización asociada con uno de sus tipos de token no impediría que se agregara una orden comercial a la cadena de bloques, pero impediría su inclusión en cualquier libro de órdenes, e invalidaría cualquier registro de liquidación que intente coincidir con ese registro de orden comercial.

Los registros de órdenes comerciales como se describió en la presente descripción funcionan como órdenes limitadas, y son equivalentes a órdenes limitadas en el sentido convencional; las órdenes comerciales permanentes sin coincidencia funcionan como órdenes limitadas en reposo en el libro de órdenes permanentes. En al menos una modalidad, los registros de transferencia como se describió

en otra parte en la presente descripción funcionan como órdenes de mercado, aunque para una sola cuenta o dirección se usa una transacción para adquirir un determinado tipo de token para sí mismo, se especificaría a sí mismo como la cuenta emisora y la cuenta de destino (alternativamente, en al menos una modalidad, especificar una cuenta de destino dentro de un registro de transferencia puede ser opcional, con la implicación de que la cuenta emisora especificada asumiría que también es la cuenta de destino si la cuenta de destino se deja sin especificar). Tales registros de transferencia de tipo multitoken solo se procesan y ejecutan si el nodo de construcción de bloques puede descubrir una trayectoria de arbitraje.

Una orden de limitación de pérdida puede implementarse como un registro de vinculación de patrón que coincide con los registros de liquidación que representan una determinada tasa de cambio, vinculada a una subrutina de contratos inteligente que ejecuta una orden de mercado por sí misma al generar un registro de transferencia como se describió anteriormente. Una orden de detención de límite puede implementarse como un registro de vinculación de patrón vinculado a una subrutina de contrato inteligente que crea internamente un registro de orden comercial.

Con referencia a **la Figura 11**, la transacción en cadena puede realizarse de la siguiente manera en al menos una modalidad. En 1110, los nodos, incluidos los nodos de billeteras y de construcción de bloques, transmiten registros de órdenes comerciales de acuerdo con al menos una modalidad. Cada orden puede incluir campos de cuenta originaria, tipos de token de compra y venta, cantidad de venta, cantidad de compra (o, alternativamente, tasa de cambio), cargo y caducidad.

En al menos una modalidad, también pueden especificarse otros campos, que incluyen un campo adicional que indica si la orden comercial puede coincidir parcialmente, o si debe coincidir completamente. Al menos una modalidad, permitiría que una orden comercial se marque explícitamente como “en reposo”, de manera que la orden no se compare antes de agregarse al libro de órdenes permanente (es decir, antes de agregarse a la cadena de bloques), y de manera que después de agregarse al libro de órdenes permanente, solo puede coincidir con las transferencias y órdenes comerciales de transmisión nuevas, no con otras órdenes comerciales permanentes.

En 1120, el nodo de construcción de bloques sintetiza un libro de órdenes a partir de registros de órdenes comerciales recientemente transmitidos y registros de órdenes comerciales permanentes en la cadena que aún son viables, no están vencidos y (al menos parcialmente) todavía no coinciden. Luego, en 1130, el nodo de construcción de bloques ejecuta subrutinas de contrato inteligentes vinculadas a patrones que coinciden con registros de órdenes comerciales, registros de liquidación u otros registros o transformaciones de estado dentro del bloque anterior. Los registros de órdenes

comerciales generados automáticamente generados automáticamente dentro de estas subrutinas de contratos inteligentes se agregan al libro de órdenes.

El nodo de construcción de bloques coincide con las órdenes en 1140 y agrega registros de liquidación al nuevo bloque. Los registros de liquidación hacen referencia a registros de órdenes comerciales coincidentes. El nodo de construcción de bloques agrega nuevos registros de órdenes comerciales no coincidentes al nuevo bloque en 1150. Los cargos por órdenes ganadas más las ganancias por margen/arbitraje se agregan a la recompensa de bloque.

En general, dos órdenes comerciales coincidirán si el tipo de token ofrecido por una orden comercial es exactamente el mismo tipo de token solicitado por la otra orden comercial, y viceversa. En al menos una modalidad, sin embargo, una orden comercial puede especificar que solicita un tipo de token particular, pero que también aceptaría cualquier tipo derivado de ese tipo de token. Si los registros de órdenes comerciales usan cadenas de etiquetas para identificar los tipos de token que se ofrecen y solicitan, pueden usar caracteres comodín para coincidir con una clase más amplia de tokens, no solo tipos de token único. Por ejemplo, FRP\_BankA puede ofrecerse a cambio de “FRP\*”, que representa todos los tipos de token que están prefijados por la cadena “FRP”, que típicamente significaría el tipo de token FRP, y todos sus tipos derivados.

En al menos una modalidad, los nodos de construcción de bloques pueden crear registros de liquidación que hagan referencia a cualquier número de registros de órdenes comerciales, y que los registros de órdenes de referencia ofrezcan realizar operaciones con cualquier número de tipos de token diferentes, siempre y cuando que, en conjunto, todos los registros de órdenes comerciales estén equilibrados y liquidados sin conflicto. Esta capacidad permite que un nodo de construcción de bloques ensamble múltiples órdenes comerciales coincidentes para cumplir con una orden grande.

Esta capacidad también permite a un nodo de construcción de bloques ensamblar cadenas de órdenes comerciales para que coincidan con una orden comercial que de cualquier otra manera no puede hacerse coincidir. Por ejemplo, cuando se agrega una orden comercial al libro de órdenes que ofrece comerciar el token A por el token B, pero no existe una oferta de orden coincidente para comerciar el token B por el token A, el nodo de construcción de bloques puede hacer coincidir la orden comercial original con dos órdenes, que comercializa B por C, y otro que comercia con C por A. Este encadenamiento de registros de órdenes comerciales también puede permitir que el nodo de construcción de bloques gane ganancias de arbitraje, de manera que puede lograrse una tasa de cambio más favorable mediante el encadenamiento de órdenes comerciales que la tasa de cambio expresada mediante órdenes comerciales independientes.

En al menos una modalidad, los tokens obtenidos a través del margen o a través del arbitraje se agregarían al grupo de recompensas del bloque denominado en tokens nativos. En tal caso, estas utilidades de margen o arbitraje tendrían que ser convertidas al tipo de token nativo al incluir con el bloque los registros adicionales de órdenes comerciales y liquidaciones que efectúan esa conversión. En al menos otra modalidad, los tokens adquiridos a través del margen o mediante arbitraje se asignan solo a la cuenta de recompensa del nodo de construcción de bloques, en su denominación original sin convertirse en tokens nativos.

Finalmente, en 1150, el bloque completo se transmite a la red, y los saldos de cuenta se actualizan dentro del estado global en todos los nodos del sistema de cadena de bloques (siempre y cuando que el bloque sea aceptado por todos los nodos). Los nodos comienzan a construirse sobre el bloque, y el proceso se repite desde la etapa 1110 para el siguiente bloque de orden comercial.

### **Operaciones en Cadena mediante el uso de las normas de la Subasta Holandesa para especificar Cargos**

De acuerdo con al menos una modalidad, la comercialización en cadena de la presente invención puede mejorarse mediante un proceso de priorización de órdenes comerciales que usa una subasta estilo holandés para seleccionar órdenes comerciales viables antes de que coincidan. Las siguientes etapas, elementos y características constituyen un método de la presente invención mediante el cual se implementa tal priorización de órdenes comerciales y esquema de coincidencia que usa registros de órdenes comerciales y registros de liquidación dentro de un sistema de cadena de bloques (aunque no todas las etapas, elementos y características son necesarios en cada modalidad).

Para comenzar, la cantidad de órdenes comerciales abiertas que constituyen el libro de órdenes permanente tiene un tope X, y solo las órdenes comerciales de oferta más alta (con respecto a al cargo del token nativo) se incluyen en el libro de órdenes en un momento dado. La “oferta más alta” en este caso se refiere al cargo de “servicio” del token nativo, no con respecto a la transacción en sí en relación con el par de tokens que se comercializan. La membresía en el libro de órdenes se decide en base a qué órdenes comerciales ofrecen las tarifas más altas para su inclusión, a través de una subasta “de estilo holandés” donde (a) la oferta más baja que elimina el estándar (es decir, la Xma oferta más alta) se aplica a todas las partes; o (b) la siguiente oferta más alta es lo que se cobra al participante, en cuyo caso la Nma oferta más alta pagará el (N-1)mo precio más alto.

Cada registro de orden comercial incluye los siguientes campos, además de otros campos relevantes que ya se han especificado anteriormente para órdenes comerciales:

(a) Caducidad: marca de tiempo o número del bloque después del cual la orden comercial quedará

fuera del libro de órdenes.

(b) Reserva del cargo: el cargo total en tokens nativos que están segregados y disponibles para aplicarse a cada bloque

(c) Oferta de cargo máximo: el cargo máxima que cualquier orden pagaría por bloque para analizarse o considerarse para coincidencia

Después de la configuración inicial, para cada bloque de cadena de bloques construido por un nodo de construcción de bloques, se sigue el siguiente procedimiento para determinar qué órdenes comerciales se incluyen en el libro de órdenes, qué registros se agregan a la cadena de bloques y cómo se asignan las tarifas:

(1) Se evalúa el conjunto de órdenes comerciales permanentes no vencidas ya agregadas a la cadena de bloques en bloques anteriores. El número  $X$  de estos registros de órdenes comerciales que especifican la oferta del cargo máximo más alto, y que tienen una porción suficiente de su saldo de reserva de cargo disponible para recubrir el cargo máximo, se agregan al bloque sintetizado. (Cualquier empate se rompe primero por reserva de cargo, luego por número del bloque y luego por un hash de su contenido)

(2) Según su propio criterio, el nodo de construcción de bloques luego agrega nuevas órdenes comerciales (es decir, registros de órdenes comerciales que aún no se han incorporado a la cadena de bloques) al libro de órdenes sintetizadas y al libro de órdenes permanentes conservado en la cadena de bloques. El nodo puede incorporar cualquier cantidad de nuevas órdenes comerciales al libro de órdenes sintetizadas, incluso sin cargo. Sin embargo:

(a) Solo aquellas nuevas órdenes comerciales que ofrecen ofertas de tarifas máximas más altas que las órdenes comerciales no conciliadas que ya están en la cadena de bloques se agregan al libro de órdenes permanentes (y, de esta manera, se agregan al nuevo bloque). Las nuevas órdenes comerciales deben especificar una oferta de cargo máximo superior a la oferta de cargo máximo más pequeña de las  $X$  órdenes comerciales permanentes sin coincidencia que ya se agregaron al libro de órdenes sintetizadas para calificar para la inclusión permanente.

(b) Si se incluye una nueva orden comercial en el libro de órdenes sintetizadas sin cumplir con el umbral de oferta de cargo máximo, solo puede incluirse si ofrece mejores condiciones de comercialización para la pareja de comercialización especificada que las órdenes comerciales permanentes no coincidentes en el libro de órdenes permanentes. Tales órdenes comerciales solo se agregan al nuevo bloque si coinciden; de cualquier otra manera, persisten en un estado “efímero” hasta que puedan reconsiderarse para su inclusión en el siguiente bloque, o hasta que venzan.

(3) A través de una subasta “estilo holandés” como se describió anteriormente, se calcula el monto real de la comisión para cada orden comercial y se deduce del saldo de reserva de comisión de cada orden comercial. Las tarifas pagadas por todas las órdenes comerciales se agregan a la recompensa total por el bloque.

Las nuevas órdenes comerciales efímeras que coinciden sin haberse incluido en el libro de órdenes permanentes no tienen que cumplir con los requisitos de cargos de la subasta estilo holandés, y no asignan ningún cargo a la recompensa de bloque a menos que se igualen. Si las nuevas órdenes comerciales efímeras coinciden, entonces el cargo realmente asignado a la recompensa de bloque será la reserva del menor cargo y el monto del cargo según lo determinado por la subasta estilo holandés.

(4) Las órdenes comerciales dentro del libro de órdenes sintetizadas coinciden y se crean registros de liquidación. Las órdenes comerciales se priorizan dentro de cada par comercial de manera que las órdenes de oferta más altas coincidirán con las órdenes de solicitud más bajas. Como se describe en otra parte de la presente descripción, sin embargo, los registros de liquidación pueden hacer referencia a dos órdenes comerciales, o más de dos órdenes comerciales, y las cadenas de órdenes comerciales pueden usarse para coincidir con órdenes comerciales que de cualquier otra manera tienen pares comerciales incompatibles.

Si las órdenes comerciales no pueden coincidir porque el precio de solicitud es más alto que el precio de oferta para ese par comercial, entonces el nodo de construcción de bloques puede usar cualquier porción del saldo de reserva de tarifas para recubrir la diferencia, o puede contribuir con sus propios tokens para recubrir la diferencia. De cualquier otra manera, tales órdenes comerciales seguirán sin coincidir.

(5) Los saldos de tokens dentro de los registros de cuenta del estado global del sistema de cadena de bloques se actualizan para reflejar las operaciones liquidadas. Cualquier token restante dentro de los saldos de reserva de cargos de las órdenes comerciales liquidadas se devuelve a las cuentas de origen de esas órdenes comerciales.

(6) Las órdenes comerciales que no coinciden se evalúan para evaluar si sus saldos de reserva de cargo todavía superan sus ofertas del cargo máximo. Esas órdenes comerciales con saldos de reserva de tarifas insuficientes se eliminan del libro de órdenes permanentes, y los saldos de reserva de tarifas restantes se devuelven a la cuenta que originó la orden comercial.

(7) Las órdenes comerciales antiguas que caducaron se eliminan del libro de órdenes permanente. Además, cualquier órdenes comerciales antiguas que previamente se encontraban en el libro de

órdenes permanentes y que no tenían una oferta de cargo máximo lo suficientemente alta como para incluirse en el libro de órdenes sintetizado también se eliminan del libro de órdenes permanentes. En ambos casos, el saldo de reserva de cargo restante se devuelve a las cuentas que originaron las órdenes comerciales.

Después de que cada bloque se procesó, en al menos una modalidad, puede realizarse una evaluación con respecto a los valores agregados de las órdenes comerciales liquidadas, agregadas y eliminadas del libro de órdenes permanentes, y que quedan en el libro de órdenes permanentes, en comparación con los valores agregados de las órdenes comerciales que quedan en el libro de órdenes permanentes cuando se completó el bloque anterior. Al comparar estos valores, que deben equilibrarse, las actividades comerciales en la cadena de bloques pueden confirmarse además como válidas.

### **Registro de Inspección dentro de los Contratos Inteligentes**

Las instrucciones de programación actuales para la lógica y el funcionamiento de contratos inteligentes en al menos una modalidad de la presente invención se ejecutan dentro de una máquina virtual similar en tipo a la Máquina Virtual Ethereum, que se introdujo por primera vez como una característica de la cadena de bloques Ethereum. Tales instrucciones de programación de contactos inteligentes se implementan mediante el uso de un lenguaje bytecode de máquina virtual completo de Turing similar en tipo al bytecode de Máquina Virtual de Ethereum. Las CPU, procesadores, computadoras, o clúster de computadoras residentes dentro de (o que constituyen) los nodos de la red de cadena de bloques instancian esta máquina virtual y ejecutan este bytecode de máquina virtual - que puede almacenarse en la memoria de la computadora o en cualquier dispositivo de memoria externo o base de datos de la computadora, que incluye la cadena de bloques - como parte de la creación de bloques, la verificación de bloques y el proceso de verificación de segmentos de la cadena de bloques, al menos, de acuerdo con una modalidad.

En al menos una modalidad, la máquina virtual aceptará un lenguaje bytecode basado en pila, que consiste en códigos de operación que funcionan para indicar a la máquina virtual que lea y escriba datos en una pila de último en entrar, primero en salir, y para realizar cálculos y otras operaciones al leer la entrada desde la parte superior de la pila y escribir la salida en la parte superior de la pila. Los códigos de operación dentro del mapa de bytecode se asignan a operaciones particulares, lo que también puede escribir datos en direcciones de memoria separadas y apartes de la pila. El bytecode de contrato inteligente puede comprender instrucciones de máquina repetibles que manipulan y actualizan datos residentes dentro del contrato inteligente, así como también datos que son accesibles a nivel mundial en la cadena de bloques.

En al menos una modalidad, la presente invención puede implementar un nuevo lenguaje de máquina virtual de bytecode, o usar (o extender) un lenguaje bytecode de máquina virtual existente - tal como el lenguaje bytecode de máquina virtual Ethereum - para incorporar ciertas nuevas instrucciones u operaciones. En al menos una modalidad, cada instrucción u operación se asignaría con una cantidad de tokens de “gas” que se consumirían cuando la operación se ejecuta dentro de un contrato inteligente.

Estas instrucciones u operaciones pueden proporcionar acceso programático a los registros y datos de la cadena de bloques, incluidos los datos del estado global, para examinar y manipular estos datos, e integrarlos en la lógica del contrato inteligente. Estas instrucciones también pueden llevar a cabo la creación de registros de datos especializados que pueden agregarse al mismo bloque dentro del cual se invoca o ejecuta de cualquier otra manera el bytecode. Estos registros de datos actualizarían el estado global del sistema de cadena de bloques a través de cualquier proceso de consenso y de intercambio de datos que se usa por esa red de cadena de bloques.

Por ejemplo, las instrucciones u operaciones de bytecode podrían agregarse a la cadena de bloques, tal como registros de génesis, registros de transferencia, registros de declaración de la propuesta y decisión, registros de órdenes comerciales, registros de liquidación, registros de coincidencia de patrones y registros de transacciones como se definen en la presente descripción, o cualquier otro registro.

Además, en al menos una modalidad, las instrucciones de lenguaje de máquina virtual de bytecode también podrían incluir lógica y operaciones usadas para inspeccionar registros de datos y cuentas que se habían agregado previamente a la cadena de bloques, o usadas para inspeccionar el estado global del sistema de cadena de bloques. Tales operaciones permitirían que tales datos se recuperen, filtren, inspeccionen, iteren, cotejen, comparen, combinen y manipulen dentro de un contrato inteligente que se había implementado mediante el uso del lenguaje bytecode.

Las instrucciones u operaciones de bytecode podrían permitir que los registros de génesis, los registros de transferencia, los registros de declaración y decisión de propuesta, los registros de órdenes comerciales, los registros de liquidación, los registros de coincidencia de patrones y los registros de transacciones se recuperen, filtren, inspeccionen, iteren, cotejen, comparen, combinen y manipulen dentro de un contrato inteligente.

Por ejemplo:

(1) Al menos una modalidad incorporaría operaciones que permitan el historial de los registros de génesis, registros de transferencia, registros comerciales y registros de liquidación que en conjunto

contribuyan a un saldo de token específico en una cuenta a recuperar. Estos registros serían accesibles y manipulables a través de operaciones en otros idiomas, lo que permite que un programa inspeccione la fuente de los tokens que posee una cuenta.

(2) Al menos una modalidad incorporaría operaciones que permitan que estos registros históricos de génesis, transferencia, transacción y liquidación se cotejen con las cuentas que originan los registros (es decir, las cuentas que originalmente firmaron los registros y contribuyeron con los tokens transferidos).

(3) Al menos una modalidad incorporaría operaciones que permitirían que los registros de declaración de la propuesta que reafirmen hechos en relación con cuentas específicas también se recuperen y pongan a disposición del contrato inteligente en tiempo de ejecución. Por ejemplo, dada una dirección de cuenta específica, y dado algún aspecto de los registros de declaración de la propuesta mediante los cuales filtrar los registros que se recuperan, los registros de declaración de la propuesta conforme se recuperarían y escribirían en la memoria de tiempo de ejecución del contrato inteligente, disponible para manipulación por otras instrucciones de bytecode. Los registros de decisión de propuesta relacionados también podrían ponerse a disposición del contrato inteligente para su análisis, cuyos registros de decisión de propuesta también se filtrarían de acuerdo con las restricciones especificadas por el contrato inteligente.

De acuerdo con al menos una modalidad, cada registro de propuesta puede incluir entre sus campos una declaración de la propuesta, que se expresaría de manera estructurada - tal como JSON o XML, o como una lista, estructura o mapa en memoria - que se presta a ser indexado y consultado. El campo de declaración de la propuesta del registro de propuesta sería inspeccionable, transitable y sujeto a manipulación (pero no modificación) dentro de la lógica de una subrutina de contrato inteligente. La declaración de la propuesta, si hace referencia a una cuenta, establecería una referencia indexada o un vínculo entre el registro de la propuesta y esa cuenta. La declaración de la propuesta también sería indexable y consultable con respecto a sus otros contenidos.

(4) En al menos una modalidad, también se implementarían operaciones de acceso directo que llevan todos estos datos a la memoria de tiempo de ejecución del contrato inteligente a la vez, sin necesidad de operaciones de recuperación de datos separadas para cada tipo de registro de datos.

Al menos una modalidad podría incluir una operación de recuperación de historial que trabajaría para acceder a la procedencia completa de determinados tokens del usuario mantenidos por una cuenta. Tal operación recuperaría todos los registros de transacciones contribuyentes, todas las cuentas originarias y alguna porción de los registros de propuesta relacionados, cuyos registros de

propuesta se filtrarían por algunas cualidades específicas de esos registros.

Mediante el uso de esta facilidad, puede implementarse un contrato inteligente para seleccionar y recuperar solo aquellos registros de propuesta que afirman un hecho en particular. De esta manera, los registros de propuesta que identifican la naturaleza de las cuentas que habían contribuido de alguna manera al saldo de una cuenta en particular podrían recuperarse e inspeccionarse, lo que permite que el contrato inteligente ejecute lógica condicional a los datos de procedencia.

(5) Al menos una modalidad incorporaría operaciones que permitirían que el contrato inteligente iterara, inspeccione y manipule de cualquier otra manera los datos agregados a la memoria por las operaciones mencionadas anteriormente.

De acuerdo con al menos una modalidad de la presente invención, la funcionalidad de Inspección de Registros descrita anteriormente se incorporaría a un nuevo lenguaje de programación completo de alto nivel de Turing - o ampliaría un lenguaje de programación completo de alto nivel existente de Turing, tal como Solidity - para especificar elementos de programación de alto nivel, incluyendo construcciones sintácticas y semánticas, palabras clave, llamadas de función, rutas lógicas, o elementos del lenguaje natural, que se compilarían en el bytecode descrito anteriormente.

En al menos una modalidad, la funcionalidad de Inspección de Registros descrita anteriormente podría incorporarse en un nuevo lenguaje de programación de alto nivel cuyos elementos son consistentes con o se asemejan al lenguaje formal usado en la escritura en lenguaje natural (por ejemplo, común dentro de la profesión legal). Este lenguaje de alto nivel puede proporcionar una sintaxis sensata para la creación de registros de datos tal como registros de génesis, registros de transferencia, registros de declaración y decisión de propuesta, registros de órdenes comerciales, registros de liquidación, registros de coincidencia de patrones y registros de transacciones como se define en la presente descripción. También se proporcionaría una sintaxis adecuada para recuperar, filtrar, inspeccionar, iterar, cotejar, comparar, combinar y manipular esos registros.

Los análogos dentro de estos idiomas de alto nivel existirían para la implementación y las operaciones del lenguaje de bytecode descrito anteriormente. Se usaría un compilador para traducir estas instrucciones de alto nivel de elementos de idioma a tal bytecode o código ejecutable, que sería ejecutable dentro de la máquina virtual de la cadena de bloques, que operaría en las CPU físicas, procesadores o computadoras, MPU o clúster de computadoras residentes o que constituyan los nodos de la red de cadena de bloques.

### **Caso de Uso 1: Ejecución de Contratos Impulsada por Eventos**

Los diversos tipos de registros descritos anteriormente y otras características, en al menos una

modalidad de la presente invención pueden usarse en una cantidad de sistemas y métodos. Uno de tales sistemas y métodos puede usar una combinación de características de al menos una modalidad para la implementación y automatización de determinados procesos legales que son impulsados por eventos del mundo real.

Por ejemplo, puede usarse para automatizar el funcionamiento de una póliza de seguro de vida o un testamento, que comprenden documentos que incluyen o representan procesos legales que se desencadenan por la muerte de una persona (el evento del mundo real). Cada uno de estos procesos puede implementarse de manera automatizada mediante el uso de una combinación de propuestas, registros de vinculación de patrones, contratos inteligentes y órdenes comerciales en la cadena de bloques de acuerdo con al menos una modalidad.

(1) Se agrega un registro de propuesta (“propuesta de morbilidad”) a la cadena de bloques, afirmando que una determinada persona ha fallecido. El bloque de decisión se define como tres meses en el futuro y el bloque de cierre una semana después del bloque de decisión. Esta propuesta permite que los terceros neutrales ofrezcan apoyo a la conclusión de los hechos que se alcanzará en forma simultánea con el bloque de cierre. También sirve como una expresión global del estado que puede ser referenciada y usada en la lógica de contratos inteligentes no relacionados.

En al menos una modalidad alternativa, la “propuesta de morbilidad” puede agregarse solo en el caso de que el difunto pase, y puede configurarse como un registro de propuesta autodeterminado, emitido por una sola entidad en la que todas las partes confían para tomar tal determinación (por ejemplo, un cuerpo gubernamental). En tal caso, dado que no se agregarían registros de decisión en respaldo de la propuesta, la confiabilidad de la entidad que emite la propuesta determinaría si tal propuesta sería suficiente para desencadenar la ejecución del contrato en este caso.

(2) Los contratos inteligentes de “resolución” pueden definirse y vincularse con la propuesta de morbilidad a través de registros de vinculación de patrones. Los registros de vinculación de patrones pueden contener definiciones de patrones que coincidan con las transiciones de estado de la propuesta de morbilidad.

(a) Podría invocarse un contrato de resolución de “renovación” en caso de que la propuesta se resuelva como falsa (es decir, en el caso de que se establezca que la persona aún no ha aprobado). Este contrato de renovación renovaría la propuesta.

(b) Uno o más contratos de resolución de “ejecución” se invocarían en caso de que la propuesta se resuelva como verdadera (es decir, en el caso de que se establezca que la persona ha fallecido de hecho). Este contrato de ejecución podría contener una lógica de proceso específica (por ejemplo, la

ejecución de un contrato inteligente de seguro de vida, o un contrato inteligente de testamento), o podría actuar como registrador de otros contratos inteligentes subordinados (un contrato “registrador”) que se ejecutarán de una sola vez en caso de que la propuesta se resuelva como verdadera, a través de la delegación del registrador.

(c) Un contrato de “registrador” podría permitir que un contrato de “testamento” sea sustituido por un contrato de “testamento” enmendado, por ejemplo, sin la necesidad de modificar ningún registro de vinculación de patrones, porque un contrato de “registrador” almacenará referencia a sus contratos inteligentes subordinados, cuya referencia puede ser reemplazada a través de la operación del contrato de “registrador” en sí.

(3) Los registros de vinculación de patrones podrían usarse para vincular estos contratos inteligentes de “resolución” con el cierre de una propuesta. Las definiciones de los patrones coinciden con los patrones de eventos expresados en una sintaxis formal: por lo tanto, el patrón de la propuesta de mortalidad que se resuelve como verdadero provocaría que se invocaran los contratos de “ejecución”, y el patrón de la propuesta de morbilidad que se resuelve como falso provocaría que se invocara el contrato de “renovación”.

(4) La lógica del proceso que debe ejecutarse en caso de fallecimiento se incorporaría en los contratos inteligentes de resolución.

### **Contrato de Seguro de Vida**

Un contrato de seguro de vida distribuiría un beneficio a un conjunto de beneficiarios, cada uno representado como una cuenta o dirección dentro de la cadena de bloques, sacando de algunos grupos de tokens.

(a) Los beneficiarios podrían ser nombrados o modificados a través de una invocación de función programática ejecutada desde una cuenta autorizada mediante el uso de credenciales de cuenta adecuadas (específicamente, firmando la instrucción de invocación con la clave privada de la cuenta).

(b) El beneficio podría definirse como un valor de token de un tipo de token particular. La cuenta propia del contrato (es decir, la cuenta controlada por el contrato en sí) necesitaría tener acceso a un valor suficiente para poder cumplir con su compromiso en caso de muerte.

(c) Un único contrato inteligente puede combinar obligaciones con respecto a múltiples personas aseguradas y, por lo tanto, puede extraer de un único grupo de valor compartido al realizar pagos.

(d) Las primas pueden pagarse directamente al contrato de seguro inteligente, lo que contribuye al grupo de valor disponible para el pago del beneficio.

(e) La utilidad puede calcularse algorítmicamente como una función del tamaño del conjunto de valor y el monto total asegurado. La utilidad puede asignarse a alguna cuenta de terceros (concebiblemente el autor del contrato), o puede distribuirse proporcionalmente a los titulares de la póliza como un reembolso, o puede mantenerse en reserva y finalmente distribuirse a todos los beneficiarios después de que todos los beneficios se hayan distribuido primero.

(f) También pueden implementarse otras rutinas como lógica dentro del contrato inteligente.

### **Un Contrato Inteligente de “Testamento”**

Un contrato inteligente de “testamento” distribuiría tokens a los herederos al fallecimiento del propietario de una cuenta (el testador).

(a) Este contrato distribuiría tokens que se mantienen dentro de la cuenta propia del contrato. Solo los tokens almacenados en la misma cuenta contractual serían distribuibles por el contrato, por lo que si los tokens se mantienen en cuentas diferentes, entonces se requerirían contratos de “testamento” diferentes. Por lo tanto, cada uno de estos contratos puede considerarse mejor como instrucciones “pagaderas al fallecimiento” para la cuenta a la que se adjunta, pero con la posibilidad de lógica compleja.

(b) En caso de que deban cambiarse herederos o asignaciones, el contrato inteligente de “testamento” puede proporcionar alguna disposición para la modificación a través de subrutinas dentro del contrato inteligente. Las partes autorizadas (es decir, cuentas o direcciones autorizadas) pueden invocar tales subrutinas para modificar las asignaciones y los heredadores; alternatively, si se implementa un contrato inteligente de “testamento” como “registrador”, entonces el contrato inteligente subordinado o contrato inteligente al que delega la responsabilidad de distribución de tokens puede ser reemplazado.

(c) Las obligaciones impositivas sucesorias pueden automatizarse al hacer que el contrato agregue una orden comercial a la cadena de bloques, cuya orden comercial convertiría porción de los tokens al tipo de token requerido para ser usado para pagar el impuesto. Las ganancias de esa transacción se dirigirían a un contrato inteligente separado que enviaría todas las ganancias a una cuenta mantenida por la autoridad gubernamental relevante para ese fin.

### **Dinámica del Incentivo dentro de la Determinación de la Propuesta**

En los ejemplos anteriores de posibles implementaciones de la presente invención, habría algún incentivo para que las diversas partes influyan en la resolución de la propuesta.

Es posible que los herederos deseen hacer que el testamento se ejecute antes de que el testador haya fallecido, y podrían intentar abrumar la propuesta de morbilidad agregando registros de decisión a

favor del resultado que favorecen. Por el contrario, los coparticipantes en un contrato inteligente de seguro de vida agrupado pueden querer evitar el pago a los beneficiarios de un difunto, para aumentar el tamaño del reembolso que pueden acumularse a sí mismos. Pueden intentar abrumar la propuesta agregando registros de decisiones contra la persona fallecida que en realidad falleció.

La anulación y competencia de incentivos dentro del sistema de propuestas garantizaría que se eviten los resultados fraudulentos, como se describe más abajo.

(a) Cada registro de decisión debe poner en riesgo los tokens nativos para respaldar un resultado particular. En caso de que la resolución de la propuesta vaya en contra de un registro de decisión, la totalidad de su valor se pierde para el originador de ese registro de decisión.

(b) Como se describió en la presente descripción, los registros de decisiones y los registros de propuestas pueden incorporar pruebas, o hacer referencia a alguna evidencia externa, en cuanto a la verdad que están afirmando, lo que les brinda a los participantes dentro del proceso de determinación de la propuesta la oportunidad de evaluar tal evidencia para tomar determinaciones independientes en cuanto a la validez de la propuesta.

En el caso específico de una “propuesta de morbilidad”, en al menos una modalidad, un registro de decisión que documente la muerte del asegurado podría contener un certificado de defunción digital criptográficamente firmado por una autoridad gubernamental, más declaraciones juradas firmadas criptográficamente de personas de alta reputación (por ejemplo, abogados), y posiblemente incluso pruebas fotográficas y en video a través de referencia a IPFS o alguna otra información de almacenamiento de acceso público.

(c) La mecánica interna del sistema de propuesta incentiva a las reclamaciones falsas verificable o no corroboradas a ser refutadas rápidamente por los registros de decisiones contradictorias - e incentiva de manera similar los registros de decisiones de alto valor en alineación con la realidad objetiva que se aportará temprano, antes de que se contribuyan incluso a los registros de decisiones erróneos.

El principio bajo el cual opera este mecanismo es el siguiente: dentro de un sistema radicalmente descentralizado, los participantes solo tienen la capacidad de coordinar sus elecciones y acciones por referencia a la realidad objetiva. Cualquier participante que vea una propuesta que se inclina hacia una resolución que contradiga la realidad objetiva probablemente se apresure a contribuir con registros de decisiones contradictorios para obtener el beneficio del valor asignado a esos registros de decisiones erróneos. Lo hará porque cree que los terceros descoordinados harán lo mismo de manera simultánea, reforzando su reclamo y compitiendo con ella para reclamar los tokens en riesgo.

En al menos una modalidad de este proceso, el registro de decisión inicial puede tener un valor de token bajo, o solo un valor tan alto como pueda incentivarse por la recompensa propia de la propuesta. De manera similar, los registros de decisiones de seguimiento que esperan reclamar una porción de la recompensa restante de la propuesta al reforzar el registro de decisión original pueden ser de valor relativamente bajo, ya que solo se incentivarían por una fracción de la recompensa propia de la propuesta.

Inicialmente, este umbral de bajo valor puede parecer un mal desincentivo contra los registros de decisión erróneos que se agregan para abrumar una propuesta en particular. Sin embargo, cada registro de decisión que contradiga el registro de decisión inicial (y su evidencia) aumenta el incentivo para que los registros de decisión se agreguen en respaldo del registro de decisión inicial. Cualquier token asignado a registros de decisión de alto valor que contradigan pruebas confiables atraerá registros de decisión que voten a favor de esa evidencia - y tendrá una buena posibilidad de perderse en esos registros. Por el contrario, los tokens asignados a los registros de decisión que respaldan la evidencia probablemente no se perderán, ya que puede suponerse que habrá más tokens en riesgo en respaldo de evidencia confiable que en contra de ello.

Cada participante del proceso de determinación de la propuesta, tanto en este caso de uso como en general, de acuerdo con al menos una modalidad, deben evaluar si el valor de token que se arriesgan se devolverá o perderá, en base a la posibilidad de que el resultado de la propuesta sea a su favor o no, en otras palabras, si el valor agregado asignado a su lado de la pregunta será mayor que el valor agregado asignado al otro lado cuando se resuelva la propuesta. En promedio, si hay una cantidad lo suficientemente grande de participantes, la probabilidad debe ser mayor que cualquier propuesta determinada que se resuelva para coincidir con la realidad objetiva, en lugar de cualquier otra manera, lo que debería incentivar una cantidad creciente de registros de decisiones a snowball en favor del resultado “objetivamente real” una vez que la realidad objetiva se conoce de manera verificable.

Sin perjuicio de lo anterior, de vez en cuando, las entidades que proporcionan evaluaciones confiables de la evidencia digital —participantes confiables en el proceso que votan los registros de decisión de acuerdo con su evaluación de la evidencia— pueden no estar de acuerdo con el resultado de la propuesta según lo determinado por el voto mayoritario. Algunos usuarios del sistema pueden preferir confiar en las determinaciones realizadas por esas entidades, en lugar de confiar en el proceso de determinación descentralizado. De acuerdo con al menos una modalidad, el autor de un contrato de resolución inteligente puede elegir incluir en ese contrato una inspección de los registros

de decisión de la propuesta, y hacer que el contrato ignore el resultado determinado por el voto mayoritario, si se encuentra que uno o más participantes de confianza han votado contra el consenso. O, en al menos otra modalidad, un registro de vinculación de patrón puede vincular un contrato de resolución no con un patrón que coincida con el cierre de una propuesta, sino con un patrón que coincida con la adición de un registro de decisión generado por uno o más participantes de confianza específicos.

Alternativamente, en al menos una modalidad, un registro de propuesta puede especificar un requisito de que algunos o todos los registros de decisión relacionados con la propuesta solo sean presentados por determinadas cuentas de confianza específicas, lo que permite de esta manera que la persona o entidad que crea un registro de propuesta restrinja la capacidad de las entidades no confiables de participar en el proceso de determinación de la propuesta y potencialmente socavarlo. En el caso más extremo donde la confianza en las partes individuales es mayor que la confianza en el colectivo, pueden usarse registros de propuestas autodeterminados en lugar de todo el proceso de determinación de la propuesta. Tales registros de propuesta son emitidos por una sola entidad en la que confían las partes relevantes para tomar una determinación por su cuenta sin confirmación externa. Debido a que no se agregarían registros de decisión en respaldo de la propuesta, y debido a que la propuesta se cierra y se decide en el momento en que se agrega a la cadena de bloques, la confiabilidad de la entidad que emite la propuesta determinaría si tal propuesta se aceptaría como verdadera, y por qué participantes. Debido a que no hay verificación externa, las propuestas autodeterminadas son generalmente más útiles para coordinar la acción entre las partes de confianza.

### **Caso de Uso 2: Confirmación de Identidad**

Los diversos tipos de registros descritos anteriormente y otras características en al menos una modalidad de la presente invención pueden usarse en una cantidad de sistemas y métodos que requieren la confirmación de una identidad de una entidad o persona.

Específicamente, los aspectos de la identidad del mundo real de un propietario de cuenta pueden estar disponibles en la cadena de bloques a través de un registro de propuesta que se adjunta a ese objeto. Tal registro de propuesta sería juzgado y evaluado por uno o más “verificadores” de identidad, que actúan para afirmar o rechazar el contenido del registro de propuesta mediante la generación de registros de decisión.

Al usar un formato de datos estándar dentro de las declaraciones de la propuesta, los resultados de la determinación de la propuesta pueden usarse para establecer hechos complejos que otros elementos del sistema de cadena de bloques pueden consumir. La funcionalidad compleja puede

entonces construirse sobre esa capa de abstracción. Por ejemplo, el proceso de determinación de la propuesta puede usarse para confirmar la identidad de un propietario de cuenta en particular, o para confirmar la relación entre dos propietarios de cuenta, y luego usar esa información para dirigir el comportamiento contractual.

En al menos una modalidad, un registro de propuesta (“propuesta de identidad”) puede agregarse a la cadena de bloques, afirmando que una cuenta particular se une a una identidad del mundo real particular. La información de identificación estructurada que comprende la declaración de la propuesta de identidad puede codificarse en un formato estándar (por ejemplo, formato JSON) para que sea más fácil de consumir a través de un contrato o billetera inteligente.

Esta declaración de la propuesta puede contener ciertos campos que representan no solo el nombre, sino potencialmente otros datos de identificación, tal como “fecha de nacimiento” y “dirección de residencia”. Los datos en tales campos estarían disponibles para el consumo y la manipulación por contratos inteligentes, así como también por billeteras y otros nodos en la red. Sin embargo, en determinadas circunstancias, es posible que esta información deba ocultarse porque es delicada o confidencial, lo que podría potencialmente derrotar el propósito de que los datos sean manipulados por un contrato inteligente u otro programa.

En al menos una modalidad, los datos de la declaración de la propuesta que deben ocultarse pueden tener un hash mediante el uso de un algoritmo de hash criptográfico, lo que permite que la información se distribuya ampliamente y, mientras que, se evita que se lea claramente. Sin embargo, los datos con hash pueden usarse para guiar la lógica informática; sin embargo, si los datos con los que se comparan se han hecho hash mediante el uso de el mismo algoritmo de hash. De esta manera, la declaración de la propuesta de una propuesta de identidad puede contener algunos puntos de datos en texto claro, tal como “país” o “jurisdicción legal” (que no son detallados y pueden usarse en la lógica abierta para guiar), y también puede contener algunos puntos de datos que están hash, como “nombre”, “dirección”, o “dirección de correo electrónico” (que solo se usan para confirmar detalles que ya se conocen).

Opcionalmente, en al menos una modalidad, el registro de propuesta de identidad puede incluir evidencia de la verdad de la propuesta, en la forma de registros de “identidad autosoberana” (según las especificaciones W3C), referencias a fuentes de datos de terceros independientes, datos de imágenes que representan documentos físicos, u otros datos relevantes que pueden usarse por verificadores u otros posibles participantes en el proceso de determinación de la propuesta. Esta evidencia y documentación de respaldo se incluirían en el registro de la propuesta por separado y

aparte de la declaración de la propuesta.

En al menos una modalidad, un usuario de cuenta puede cifrar porción o toda la evidencia y documentación de respaldo que se adjunta o a la que hace referencia la propuesta de identidad, de manera que solo sea accesible para verificadores de confianza. Tal evidencia y documentación de respaldo se cifrarían mediante el uso de las claves públicas de estos verificadores de confianza, quienes luego descifrarían esos detalles de manera privada sin que se divulguen públicamente. Alternativamente, los usuarios de cuentas pueden transmitir evidencia y documentación de respaldo directamente a los verificadores a través de una comunicación directa fuera de banda que no involucra la cadena de bloques; o pueden compartir la información públicamente.

Si un verificador involucrado en la evaluación de una propuesta de identidad es suficientemente confiable, ya sea tener un seguimiento lo suficientemente grande que pueda agregar automáticamente registros de decisiones de respaldo, o ser confiable sin la participación de los participantes de respaldo como se describe más abajo, entonces una propuesta de identidad que se ha confirmado públicamente en la cadena puede ser confiable incluso sin la documentación de respaldo que necesite hacerse pública.

### **Determinación de la Propuesta de Identidad**

De acuerdo con al menos una modalidad, de conformidad con el procedimiento de confirmación de propuesta descrito en otra parte en la presente descripción, una propuesta de identidad puede verificarse a través del siguiente procedimiento.

(1) Entre los participantes que votan sobre la propuesta a través de los registros de decisiones estarían las personas o empresas que actúan de manera convencional como “verificadores”, que a través de la reputación son de confianza para otros participantes de la red.

Fuera de la red de cadena de bloques, estos verificadores usarían información y procesos del mundo real para evaluar una propuesta y su evidencia de respaldo, para realizar una evaluación de la validez de la propuesta. En dependencia de las conclusiones y el nivel de certeza alcanzado con respecto a la propuesta - nuevamente, evaluados independientemente y por separado de la red de cadena de bloques - los verificadores pueden elegir generar un registro de decisión contra la propuesta o a favor de esta.

Las recompensas de la propuesta incentivan a los verificadores a evaluar las propuestas de identidad y generar registros de decisiones. Estos incentivos se mejoran si un registro de propuesta específica que solo determinados verificadores pueden presentar registros de determinación (de esta manera, aumenta la probabilidad de que se gane la recompensa). Sin embargo, en determinadas

circunstancias, estos incentivos pueden ser insuficientes.

En al menos una modalidad, puede implementarse un sistema de recompensa suplementario de manera que este se usa un registro de vinculación de patrón para transmitir valor adicional a uno o más verificadores específicos. En este escenario, un usuario con un interés en la resolución de una propuesta particular puede asignar tokens a un contrato inteligente, lo que a su vez transferiría esos tokens a un validador particular tras la ejecución. Este contrato inteligente estaría vinculado a una definición de patrón que corresponde a los registros de decisión que aún no han sido emitidos por verificadores de confianza a favor de la propuesta en cuestión. En al menos una modalidad, el registro de vinculación de patrón tendría una fecha de vencimiento o altura del bloque, después de lo cual la asignación de valor caduca y se devuelve a la cuenta original.

Con este enfoque de recompensa suplementario - implementado con registros de vinculación de patrones estándar y contratos inteligentes - los verificadores de confianza pueden ser incentivados a participar en el proceso sin tener que poner en riesgo los tokens. Sin esta recompensa suplementaria, los verificadores asumen un riesgo, aunque pequeño, de que los tokens que asignan a los registros de decisión se perderán si se asignan suficientes tokens a los registros de decisión opuestos. Debido a que la recompensa asociada con una propuesta es proporcional a los tokens colocados en riesgo, los verificadores de confianza de cualquier otra manera pueden tener que poner en riesgo los tokens sustanciales para acceder a la recompensa, además de la necesidad de hacer el trabajo de verificación del mundo real. Aunque no es necesario, una recompensa suplementaria garantizada contrarrestaría estos desincentivos.

(2) Otros participantes de la red (“participantes de respaldo”) pueden monitorear la actividad de estos verificadores de confianza. Al detectar la creación de un registro de propuesta por una cuenta que se sabe que está asociada con un verificador de confianza, estos participantes de respaldo pueden generar registros de decisión adicionales con votación en la misma dirección.

Entre estos participantes de respaldo, algunos pueden monitorear activamente la actividad de la cadena de bloques a través de un mecanismo o proceso externo, mientras que otros pueden, en cambio, crear registros de vinculación de patrones que coincidan con los registros de decisión de estas cuentas de verificador de confianza. Estas definiciones de patrones estarían vinculadas a contratos programados para generar automáticamente registros de decisión adicionales, qué contratos se invocarían después de la creación de un registro de decisión por cualquiera de las cuentas de verificador especificadas.

En al menos una modalidad, determinados participantes de respaldo tal vez deseen crear registros de

decisión de respaldo si *más de un* verificador de confianza ya ha verificado la propuesta de identidad. El simple hecho de que una definición de patrón haya coincidido sería, por sí solo, un indicador insuficiente de si un contrato vinculado debería generar registros de decisión de respaldo. En este caso, un contrato inteligente vinculado examinaría programáticamente todos los registros de decisión agregados a la propuesta - a través de la API de consulta/recuperación/transporte de la propuesta, para determinar si *otros* verificadores *de* confianza ya habían votado o no a favor de la propuesta. Solo en ese caso, el contrato inteligente agregaría registros de decisiones de respaldo.

Los participantes de respaldo están motivados por el potencial de recibir porción de la recompensa por decisión de propuesta que se reserva para los participantes de respaldo (es decir, el valor *de*  $D_{+N} - f(r_{D+N})$  como se describió en la presente descripción con respecto al proceso de determinación de la propuesta). Los participantes de respaldo enfrentan el riesgo de perder sus tokens si otros participantes cometen una mayor cantidad de tokens a favor de la idea de que el verificador de confianza está equivocado. Sin embargo, debido a su confianza en el verificador, los participantes de respaldo creen que este riesgo es comparativamente pequeño en relación con el valor esperado de su recompensa.

Los participantes que ven un beneficio al votar en oposición a un verificador particular también pueden establecer mecanismos, procesos o contratos inteligentes vinculados a patrones para agregar automáticamente registros de decisión que voten en la dirección opuesta como el verificador.

(3) Al final, cualquier verificador que ordene un seguimiento de los participantes de respaldo capaces de abrumar a la oposición debe ser considerado “confiado” por la red. Estos verificadores realizan un servicio esencial al sincronizar el conocimiento del mundo real con la cadena y permitir que esta información sea usada por otros participantes de la red de cadena de bloques para guiar el curso de los contratos automatizados (y para invocar contratos vinculados al patrón al concluir el proceso de determinación de la propuesta). En el caso de las propuestas de identidad, la conclusión del proceso de determinación puede usarse por los participantes en la red de cadena de bloques para ayudar a guiar cualquier cantidad de procesos y decisiones.

Sin embargo, tenga en cuenta que los participantes de respaldo son opcionales y sirven principalmente para reforzar la decisión tomada por el verificador. Si ningún participante de respaldo contribuye con los registros de decisión, o si el registro de propuesta original se configura para excluir a los participantes de respaldo (especificando que solo los verificadores de confianza pueden contribuir con los registros de decisión), el procedimiento de verificación de identidad procedería sin ellos. Finalmente, depende de los usuarios del sistema de cadena de bloques determinar qué

información agregada a la cadena de bloques es útil y cómo usarla. Los registros de propuesta autodecididos y los registros de propuesta que son decididos por un solo verificador de confianza pueden ser suficientes para algunos usuarios; otros usuarios pueden preferir la seguridad adicional proporcionada por “la sabiduría de la multitud” de los participantes de respaldo.

### **Acceso a Registros de Identidad a través de API**

Si la afirmación central de un registro de propuesta se representa como datos estructurados, una implementación de cadena de bloques puede optar por poner esos datos a disposición de los contratos inteligentes ejecutables y de los registros agregados a la cadena de bloques. En al menos una modalidad, las propuestas pueden recuperarse a través de un lenguaje de consulta estándar o API disponible dentro del lenguaje de contrato inteligente de la implementación de la cadena de bloques, de manera que los condicionales que inspeccionan el estado de la propuesta pueden guiar la actividad automática dentro de la máquina virtual de contrato inteligente.

(1) Un contrato de seguro de vida puede depender de que el beneficiario esté relacionado con el asegurado de alguna manera. En al menos una modalidad, un contrato inteligente de seguro de vida puede consultar los registros de la propuesta para una propuesta que reafirma determinados hechos que pertenecen a esa cuenta de beneficiario particular. Si se encuentra una propuesta confirmada que refleja la afirmación requerida (específicamente, que el titular de cuenta del beneficiario está relacionado con el asegurado) sin que de cualquier otra manera sea inválida (por ejemplo, que ese caducada o contradicha por otra propuesta), entonces el contrato inteligente puede continuar en su ejecución.

(2) Una venta o transferencia de activos en particular puede depender de que una o más partes tengan sus identidades explícitamente confirmadas, por ejemplo, para satisfacer los requisitos de “conocer a su cliente”. En al menos una modalidad, un contrato inteligente ejecutable en cadena puede configurarse para mediar esta ocurrencia, permitiendo solamente que la venta o transferencia proceda si la cuenta o cuentas participantes están vinculadas a individuos identificables que coinciden con ciertos criterios, tal como edad o nacionalidad. Tal contrato puede implementarse de manera generalizada, de manera que funcione como un mediador reusable para estos tipos de transacciones, o puede implementarse como una implementación personalizada de un solo uso para un uso de una vez.

(3) Ciertas transacciones bancarias de alto riesgo pueden mediar de esta manera. En al menos una modalidad, antes de permitir que los tokens denominados en moneda en cadena se cambien por efectivo, un banco puede requerir que tales tokens cumplan con los requisitos de un contrato que

verifica la procedencia de esos tokens. Este contrato recuperaría y manipularía datos relativos al historial de cuentas que podrían haber contribuido en el pasado al valor de los tokens que se están intercambiando, y luego recuperaría propuestas de identidad relativas a esas cuentas. Solo si una proporción suficiente de esas identidades es capaz de satisfacer ciertos requisitos, el contrato permitiría que la transacción proceda.

### **Proceso de Intercambio de Identidad**

Ocasionalmente, un titular de cuenta puede querer compartir su identidad con otro titular de cuenta de manera privada, sin compartir esos detalles con toda la cadena de bloques. Sin embargo, es posible que el titular de cuenta que recibe esta información no confíe en esta información que proviene directamente del primer titular de cuenta y que desee recibir una confirmación independiente de que la información de identidad es correcta. En al menos una modalidad de la presente invención, el siguiente proceso y método sirve para satisfacer ambos requisitos:

- (1) El primer titular de cuenta genera una propuesta de identidad que declara cierta información pública (jurisdicción del gobierno), hashes otra información (nombre, dirección, número de teléfono, etc.) y proporciona referencia a la evidencia de respaldo (foto de la ID del gobierno junto con el número de cuenta de la cadena de bloques, datos de identidad completos no hashulados) al señalar un empaque encriptado cargado a IPFS u otro sistema de almacenamiento en la nube.
- (2) El primer titular de cuenta autoriza solo a un único verificador de confianza a presentar registros de decisiones, cifra la evidencia de respaldo con la clave pública de ese verificador y agrega un cargo de incentivo para atraer a ese verificador a revisar la propuesta.
- (3) El verificador de confianza lee la propuesta de identidad y luego recupera la evidencia de respaldo de IPFS y la descifra. Luego, el verificador marca los datos de identidad despejados tomados de la evidencia de respaldo y compara ese hash con el hash encontrado en la propuesta de identidad. Finalmente, el verificador revisa el resto de la evidencia de respaldo (ID del gobierno), la compara con los datos de identidad y la compara con cualquier base de datos gubernamental o pública que tenga disponible para evaluar la veracidad de los datos de identidad.
- (4) Si el verificador de confianza considera que los datos de identidad son exactos, agrega un registro de decisión a la cadena de bloques en la afirmación. De cualquier otra manera, agrega un registro de decisión negativa. Si el verificador de confianza puede afirmar la propuesta de identidad, entonces el proceso puede avanzar.
- (5) La billetera del primer titular de cuenta crea un mensaje que contiene la información de la cuenta y toda la información de identidad, en claro. El primer titular de cuenta luego comparte este mensaje

con un segundo titular de cuenta. Este mensaje puede estar en formato binario, codificado como una URL, codificado como un código QR o codificado de cualquier otra manera, y el mensaje puede compartirse a través de una conexión de enchufe de red, a través de un SMS o de un mensaje de correo electrónico, a través de una transmisión de radio inalámbrica, o a través de una imagen capturada con una cámara.

(6) El segundo titular de cuenta, al recibir este mensaje, usará su billetera para decodificarlo. Su billetera recupera los detalles de la cuenta de la cadena de bloques, incluida la propuesta de identidad confirmada. Dado que el segundo titular de cuenta confía en el mismo verificador que el primer titular de cuenta usó para verificar su identidad, el segundo titular de cuenta puede comparar el mensaje compartido por el primer titular de cuenta con la información de la cadena de bloques.

(7) El segundo titular de cuenta hashes la información de identidad que el primer titular de cuenta había compartido en su mensaje y compare esta información con los datos de la propuesta de identidad obtenidos de la cadena de bloques. Si las dos hashes coinciden, el segundo titular de cuenta sabe que el mensaje compartido por el primer titular de cuenta contiene la misma información verificada por el verificador de confianza.

Si el segundo titular de cuenta en este caso requiere una confirmación adicional de la identidad del primer titular de cuenta, o si el segundo titular de cuenta necesita más información sobre el primer titular de cuenta, por ejemplo, para realizar la diligencia debida de “conocer a su cliente”, el segundo titular de cuenta puede solicitar esa información directamente al verificador de confianza. En al menos una modalidad, el primer titular de cuenta puede incluir con el mensaje que envía al segundo titular de cuenta una instrucción firmada criptográficamente que autoriza al verificador de confianza a compartir evidencia de respaldo y otra información con el segundo titular de cuenta. Si el segundo titular de cuenta comparte esto con el verificador de confianza, el verificador de confianza compartirá la evidencia directamente con el segundo titular de cuenta, no encriptado y fuera de banda, separado y aparte de la cadena de bloques.

### **Caso de Uso 3: Seguimiento y Título de Activos**

Los diversos tipos de registros descritos anteriormente, y otras características, en al menos una modalidad de la presente invención, pueden usarse en una cantidad de sistemas y métodos que registran, rastrean y median la propiedad de los activos.

La titularidad de un activo puede codificarse como un token en una cadena de bloques y puede rastrearse y transferirse a través de operaciones de cadena de bloques y contratos inteligentes. Sin embargo, una cadena de bloques por sí sola no proporciona suficiente infraestructura y funcionalidad

para permitir la implementación de un sistema integral de seguimiento y título de activos.

Una cadena de bloques resuelve dos problemas clave: primero, elimina la dependencia de un sistema centralizado, lo que permite que la información y el seguimiento se distribuyan entre muchos participantes; segundo, garantiza que todos los registros, incluidos los registros de transferencia, sean inmutables y potencialmente permanentes, reduciendo o eliminando la posibilidad de pérdida de datos, destrucción de registros o disputas que surjan de preguntas con respecto a la legitimidad de la documentación clave. Dicho esto, incluso cuando se usa una cadena de bloques prototípica para implementar un sistema de seguimiento, aún persisten otros problemas que deben resolverse a través de procesos y prácticas que típicamente requieren confianza en instituciones fuera de la cadena y sistemas informáticos centralizados fuera de la cadena.

Esos desafíos persistentes incluyen los siguientes: cómo reconocer un token o contrato inteligente dados como personas con autoridad en el mundo real para registrar o conmemorar un activo determinado; cómo vincular la posesión de un token o contrato inteligente que represente un activo con los derechos de propiedad del mundo real; cómo mediar adecuadamente la transferencia de propiedad de un activo dentro de una transacción compleja que puede implicar varias etapas, incluidos los pagos en moneda “por decreto”; cómo contabilizar y representar el historial de propiedad o el origen de un activo que precede el registro de ese activo en cadena (por ejemplo, historial del título). Estos desafíos son mitigados o resueltos por las funcionalidades introducidas por la invención.

El seguimiento de activos del mundo real mediante el uso de cadenas de bloques probablemente se remonta a la introducción del concepto de una “moneda coloreada”. Se han propuesto varios enfoques específicos y de exclusión mutua con respecto a las monedas coloreadas, y el concepto se ha implementado de diferentes maneras en la cadena de bloques de Bitcoin, así como también en otras cadenas de bloques, especialmente el protocolo Omni Layer. Estos enfoques son distintos del concepto del “tipo de token del usuario” de la presente invención de maneras importantes, aunque todos se relacionan con la emisión de tokens en la cadena de bloques. Con la introducción del lenguaje Solidity (usado por Ethereum y otras cadenas de bloques), se introdujo otro enfoque distinto para el seguimiento de activos; lo más notable es el protocolo ERC-20 y sus descendientes y derivados.

La presente invención, en al menos una modalidad, difiere de ambos enfoques de varias maneras notables:

(1) Los contratos inteligentes implementados mediante el uso del lenguaje Solidity (tal como los

contratos ERC-20) pueden almacenar información de token y propiedad como datos clave/de valor que se conservan en el contrato. Un contrato individual debe redactarse de manera tal que encapsula la creación y gestión de un activo/token dentro de los límites del contrato y su implementación específica.

Una desventaja de este enfoque es que todas las operaciones en ese activo/token se implementan dentro del código propio del contrato inteligente de encapsulación. Cualquier individuo que tenga tales tokens debe confiar en la exactitud de la implementación correcta, y la competencia de su creador, para garantizar que los tokens rastreados por ese contrato funcionen correctamente y de hecho sean seguros. Se han perdido fondos significativos debido a la implementación fallida o defectuosa de tales contratos inteligentes. Además, las asignaciones de propiedad de activos/tokens deben rastrearse dentro de los datos del contrato, de manera que cualquier activo con una gran cantidad de propietarios corre el riesgo de abrumar los recursos informáticos que pueden asignarse a un único contrato. De manera similar, debido a que todas las operaciones se implementan en Solidity - que se ejecuta a través de un bytecode interpretado, se limita a estructuras de datos eficientes disponibles y es poco probable que estén bien optimizadas - estas operaciones son más lentas que si las mismas operaciones se implementaran como funcionalidad integral del nodo de construcción de bloques.

Más importante aún, es costoso y difícil que los activos rastreados dentro de los contratos inteligentes sean debidamente validados. Algunas validaciones pueden no ser posibles - por ejemplo, las validaciones relacionadas con el cumplimiento de las reglamentaciones.

Al integrar la creación, transferencia y seguimiento de activos en los principios de la cadena de bloques en sí, esas operaciones se validan como parte de la validación de bloques durante los procesos de creación y aceptación de bloques.

(2) Las monedas coloreadas han sido suplantadas en gran medida por tokens basados en Solidity. Sin embargo, las implementaciones tal como Omni Layer todavía se usan para ciertas aplicaciones, y vale la pena notar las limitaciones y características que distinguen las monedas coloreadas de los “tokens del usuario” descritos de acuerdo con al menos una modalidad de la presente invención.

Las implementaciones de monedas coloreadas como Omni Layer confían en la capacidad de implementaciones de billeteras personalizadas para agregar metadatos personalizados a las transferencias de cadena de bloques, y para rastrear la propiedad de valores de “token” personalizados creados y transferidos a través de esos metadatos de transferencia. Típicamente, se crea un registro que transfiere un valor de minimis de moneda y agrega de manera simultánea

algunos metadatos únicos a ese registro de transferencia o transacción. Los valores de monedas que ya están en la cadena de bloques están marcados de alguna manera como que representan activos particulares; alternativamente, los metadatos almacenan todos los datos relevantes para el token, incluidos los valores de token, y las monedas nativas en la cadena de bloques son ignoradas.

Las monedas coloreadas se basan en el hecho de que todo el historial de una cadena de bloques está disponible para su inspección. Para determinar si una “moneda” en particular está asociada con un activo en particular, todos los registros de transferencia que contribuyan al saldo de monedas en esa dirección deben rastrearse a la etapa de “coloración” original. Este seguimiento y rastreo se realiza mediante un software de billetera personalizado, y ocurre fuera del proceso de validación realizado por nodos de construcción de bloques.

Este enfoque tiene un número de inconvenientes, especialmente que se basa enteramente en implementaciones de billeteras. El significado de los metadatos, el rastreo del historial de una moneda y la gestión general del esquema se implementa dentro de un cliente, o “billetera” que se conecta a la cadena de bloques, y no reside dentro de la cadena de bloques en sí.

El efecto práctico de esto es que todos los participantes deben usar el mismo software de cliente. Además, cualquier error en el cliente (o hackers) puede tener el efecto de socavar la integridad del esquema. No hay nada que impida que dos sistemas de monedas coloreadas de la competencia usen los mismos metadatos para codificar los mismos activos o activos diferentes. Esto es especialmente problemático si uno de tales sistemas competidores tiene intenciones fraudulentas y puede aprovechar esta confusión para fines nefastos. No hay nada en la cadena que impida que las monedas coloreadas se transfieran a una dirección que contenga monedas sin color, o monedas que hayan sido destinadas a representar activos distintos. Tampoco hay manera de evitar que los metadatos corruptos que interfieren con el funcionamiento del software de cliente que cumple con las normas se propaguen a través de la red.

Ninguno de estos problemas existe cuando los tokens se sacan de los metadatos y se trasladan a los datos reales de la cadena de bloques administrados por nodos de construcción de bloques. Las billeteras que tienen implementaciones divergentes pueden continuar acumulando datos contradictorios en la cadena de bloques. Los nodos de construcción de bloques con implementaciones divergentes, sin embargo, bifurcan la cadena de bloques, de manera que cualquier actor malicioso o sistema defectuoso se aísla naturalmente en sí sin interferir con el funcionamiento de la cadena de bloques de manera más amplia. Los datos inválidos, por definición, nunca pueden incorporarse a la cadena de bloques.

(3) En al menos una modalidad de la presente invención, los “tokens del usuario” no son una capa agregada encima de otra funcionalidad (es decir, seguimiento de tokens en el caso de “monedas coloreadas” y datos de contrato en el caso de activos implementados mediante el uso de Solidity). En cambio, los tokens del usuario son primitivos de primera clase dentro de la cadena de bloques. No es obligatorio incluir un campo de tipo de token con cada registro, pero sin incluir el campo implica que el tipo de token es “nativo”, que es un tipo de token representado por la cadena vacía (todos los tipos de token están representados por una cadena de caracteres distintos). Por lo tanto, incluso cuando está ausente, el concepto de “tipo de token” existe en todo el sistema.

A diferencia de los contratos Solidity, los tokens del usuario y los tipos de token del usuario de la invención no son datos de *contrato*, sino datos de *cuenta* integrales que son contabilizados y administrados por la operación del nodo de construcción de bloques. En dependencia de la modalidad, cada cuenta tendrá un tipo de token, o las cuentas tendrán valores para múltiples tokens; pero esto será información integral para la cuenta. Los tokens podrán transferirse de una cuenta a otra, y cualquier cuenta o contrato puede tener cualquier tipo de token.

La determinación del estado y la exactitud de cualquier valor de token del usuario se realiza y verifica mediante el funcionamiento de la cadena de bloques nativa, ya que es validada y revalidada por todos los nodos de construcción de bloques en toda la red. A diferencia de las “monedas coloreadas”, un “token del usuario” en virtud de esta invención no depende de los metadatos que se rastrean y verifican mediante factores externos; en su lugar, se verifica como parte del funcionamiento integral de la cadena de bloques en sí. A diferencia de los tokens Solidity, los tokens del usuario aquí no están restringidos por las limitaciones del contrato que lo encapsula.

En combinación con la verificación a nivel de bloque realizada por los nodos (que garantiza el cumplimiento global), las restricciones sobre el reuso de cadenas de etiquetas particulares para los tokens minimizan o eliminan la posibilidad de que los tokens con nombres confusos o competitivos se agreguen a la cadena de bloques, lo que reduce el riesgo de fraude.

(4) En al menos una modalidad, mediante la introducción del concepto de “token derivado”, la presente invención implementa un medio por el cual (a) las autoridades de control (es decir, las organizaciones humanas) pueden tomar una determinación del mundo real en cuanto a qué activos pueden enumerarse bajo qué nombres en la cadena de bloques, y evitará que los activos engañosos o fraudulentos se coticen mediante una función de control de acceso; (b) tales autoridades de control pueden delegar esa función de control de acceso a subordinados, subsidiaria, organizaciones de terceros para algún subconjunto de su autoridad, a través de una “cadena de autoridad” que emana

de la cuenta raíz; y (c) existe un vínculo claro entre los registros y las actividades en cadena y la autoridad legal de las autoridades de control personificada por para vincular esos registros y actividades a los activos del mundo real.

(5) En al menos una modalidad, al introducir el concepto de un “registro de propuesta” y un “registro de decisión”, el presente sistema y método introduce un proceso mediante el cual la información del mundo real - incluida el historial/procedencia de un activo del mundo real - puede vincularse a un token de activo de una manera confiable y verificable, y de manera que la información puede consumirse mediante código de contrato inteligente.

(6) En al menos una modalidad, al introducir registros de órdenes comerciales en cadena y registros de liquidación, el presente sistema y método permite la interconexión, combinación e intercambiabilidad de diferentes tokens de activos - incluidos los tokens de activos que no están diseñados o no tienen la intención de interactuar en absoluto - sin que las personas responsables de su emisión necesiten colaborar de cualquier manera.

### **Ejemplos de Tipos de Tokens Específicos Respaldados por Activos**

El siguiente es un ejemplo específico de cómo las herramientas y características de la invención, en al menos una modalidad, pueden usarse para administrar una clase específica de activos del mundo real, en este caso bienes inmuebles (aunque no todas las etapas, elementos y características son necesarios en cada modalidad).

(1) Un gobierno estatal o regional, una autoridad o asociación rectora emitiría algunos tokens del usuario para sí mismo, o recibiría tokens por porción de una entidad autorizada por separado, obteniendo de esta manera la autoridad para definir clases de activos dentro de su jurisdicción. Estos tokens no tendrían ningún valor independiente, pero proporcionarían una base de autoridad para la emisión de tokens derivados que representen activos del mundo real.

La configuración de “nueva emisión” se configuraría como “local”, lo que permitiría declarar los tokens derivados. La configuración de “suministro” se establecería en “ilimitado”, lo que permitiría emitir un valor máximo libre para el tipo de token. Al imaginar que el gobierno estatal es Nueva Jersey, EE.UU., una cadena de “US-NJ” podría usarse como etiqueta para el token, por ejemplo. Todos los tokens emitidos que tengan una representación de cadena que comience con “US-NJ” serían identificables como emitidos bajo la autoridad del gobierno estatal de Nueva Jersey. La cuenta de la cadena de bloques a la que se han emitido tokens “US-NJ” — y la organización en posesión de la clave privada que la controla — estaría facultada para crear tales tokens.

(2) El gobierno estatal puede delegar a municipalidades individuales la capacidad de emitir sus

propios tokens.

En este caso, la configuración de “nueva emisión” se configuraría como “local”, lo que permitiría que los tipos de tokens derivados sean emitidos por la cuenta municipal. La configuración de “suministro” se establecería como “emisor original”, lo que limita la cantidad total de tokens que la municipalidad puede emitir, incluida la cantidad total de nuevos subtipos de token, a menos que el gobierno estatal o regional incremente la cantidad de tokens. Si la municipalidad es Patterson, NJ, entonces la representación de la cadena para los tokens de la municipalidad puede ser “US-NJ.Patterson”. Si la intención de la delegación de tokens es permitir que la municipalidad emita tokens de activos en respaldo de lotes de bienes raíces, y si el estado permite que la municipalidad emita hasta 200 tokens, entonces el valor del token emitido será igual a 200. Si ese número debe aumentarse en el futuro, la cuenta controlada por el estado puede emitir más tokens a la cuenta controlada por la municipalidad (“nueva emisión” establecida como “emisor original”).

La municipalidad puede entonces emitir tokens que representen lotes indivisibles que pueden ser propiedad de entidades legales individuales. Por ejemplo, puede emitirse un tipo de token bajo la etiqueta “US-NJ.Patterson.Lot-100” que representa la propiedad en “Lot 100” dentro de la ciudad. Tal token se emitiría con “nueva emisión” establecida en “emisor original” y “suministro” también establecida en “emisor original”, con un valor de token de 1 que se transfiere en el momento de la creación, lo que en conjunto significa que el receptor del token no podría emitir ningún token derivado.

Alternativamente, la municipalidad puede elegir delegar a un desarrollador de condominios la capacidad de emitir tokens que representan unidades individuales en un edificio construido sobre un lote particular. En este caso, el token se emitiría con “nueva emisión” establecida en “local”, “suministro” establecida en “emisor original” y con un valor de token igual al número esperado de unidades. Un token emitido por el desarrollador del condominio puede tener un nombre tal como “US-NJ.Patterson.Lot-100.Unit-1A”

Cada etapa de delegación descrito anteriormente se lleva a cabo a través de la creación de un registro de transferencia en cadena, que transfiere autoridad y control desde la cuenta de origen que contiene el token base a la cuenta de destino que recibe el token derivado. Como tal, cada registro de transferencia requiere la firma criptográfica de la cuenta de origen en cada etapa. Esta cadena de firmas criptográficas establece una cadena de fideicomiso verificable que vincula al titular de un activo que representa a la Unit-1A con el Estado de Nueva Jersey.

Debido a la cadena de confianza representada por esta cadena de registros de génesis firmado

criptográficamente, puede ser suficiente que los participantes de la cadena de bloques simplemente confíen en la trayectoria de delegación que conduce a la creación de un token de activo de este tipo para confiar en su aplicabilidad en el mundo real. En otras palabras, para que un participante confíe en que un token de tipo “US-NJ.Patterson.Lot-100.Unit-1A” realmente representa la propiedad de una unidad particular en un edificio particular en Patterson, NJ, puede ser suficiente para que ese participante sepa que los tokens “US-NJ” se emiten bajo la autoridad del Estado de Nueva Jersey, y que todos los tokens son tokens autorizados. Esto es especialmente cierto para aquellas modalidades de la presente invención que aplican restricciones a las etiquetas de token, de manera que la etiqueta de un tipo de token base debe ser una subcadena que cumpla con la porción inicial de la etiqueta del tipo de token derivado. Sin embargo, si estos elementos resultan ser insuficientes para establecer confianza, los registros de “propuesta” y “decisión” que se agregan a la cadena de bloques pueden resolver cualquier problema.

Puede crearse un registro de propuesta que afirme que el token “US-NJ” es realmente emitido por el Estado de Nueva Jersey. Los metadatos incorporados en el registro de génesis original para “US-NJ” pueden incluir alguna evidencia que respalde este hecho, y los registros del mundo real fuera de la cadena podrían evidenciar esta relación. Sin embargo, un registro de propuesta que realice tal afirmación, haga referencia a tal evidencia y una determinación de propuesta exitosa a favor de esa afirmación, junto con verificadores de confianza conocidos que voten en apoyo, aumentaría la confianza de la asociación.

En al menos una modalidad, puede crearse un registro de propuesta que afirme que el activo que respalda un token particular está libre de todas las lean y de cualquier otra manera está limpio a la fecha de emisión de ese token. Los verificadores de confianza que voten a favor de tal propuesta podrían ser entidades capaces de realizar una búsqueda optimizada dentro de la jurisdicción aplicable, por ejemplo, compañías de seguros de título pagadas por propietarios de activos para asegurar el título cuando se emite el token. Tal registro de propuesta se adjuntaría al tipo de token e incluiría una declaración de la propuesta de que el activo subyacente tiene un “título verificado”. Esto aumentaría el estado del tipo de token, cuyo estado aumentado puede recuperarse dentro de un contrato inteligente a través de la API de consulta/recuperación de propuesta.

Otras características de la invención permitirían la transferencia de activos en cadena sin necesidad de mediación externa o documentación. Los registros de órdenes comerciales y los registros de liquidación, por ejemplo, pueden usarse para crear un mercado comercial abierto para activos asociados con tokens de activos.

En al menos una modalidad, los contratos de ventas contingentes autoejecutables podrían implementarse mediante el uso de contratos inteligentes. Por ejemplo, los tokens respaldados por activos pueden retenerse temporalmente dentro de un contrato inteligente de “garantía”, a la espera de los fondos del préstamo; cuando los fondos del préstamo se transfieren al contrato inteligente, esto satisfaría la contingencia, lo que potencialmente completaría la transferencia de propiedad automáticamente al transferir los fondos del préstamo al vendedor y los tokens respaldados por activos al comprador. En al menos otra modalidad, un registro de vinculación de patrón puede vincular un patrón que coincida con la venta o transferencia de un token de activo (que podría representar ciertos bienes inmuebles, por ejemplo) con una subrutina dentro de un contrato inteligente de “garantía” para indicar que se ha cumplido una contingencia de “venta de propiedad”. Las subastas autoejecutantes pueden implementarse en el uso de contratos inteligentes en al menos otra modalidad. Un contrato inteligente de “subasta” puede tener un activo pendiente de la finalización de la subasta y mediar en el manejo de ofertas y fondos de pago. Mientras la subasta esté abierta, las ofertas en la forma de tokens respaldados por monedas pueden enviarse al contrato, lo que haría un seguimiento de la cantidad de tokens de oferta aportados por cada cuenta participante. Cuando se finalice la subasta, los tokens de la oferta ganadora se reenviarán a la cuenta del propietario del token de activo original, el token de activo se reenviará a la cuenta originaria del postor ganador, y las ofertas perdedoras de los apostadores se devolverán a sus cuentas.

El hecho de que los tokens del usuario no se implementen *dentro de* los contratos en algunas modalidades, sino que sean datos primitivos simples que pueden asociarse con cualquier cuenta o contrato en cadena, significa que los contratos inteligentes nuevos y novedosos pueden crearse para fines especiales, y los activos pueden enviarse a tales contratos inteligentes de manera ad-hoc según sea necesario. Los tokens de activos no son “activos inteligentes” porque no tienen implementación con ellos; en su lugar, pueden pasarse entre contratos inteligentes que cada uno realiza funciones y operaciones diferentes en esos tokens.

#### **Caso de Uso 4: Tokens de Activos Respaldados por Divisas, Comercio de Divisas y Remesas**

En otra modalidad, el presente sistema y método pueden usarse para tokens de activos respaldados por divisas, comercio de divisas y remesas financieras.

En al menos una modalidad de la presente invención, los tokens de activos respaldados por divisas pueden crearse por instituciones de confianza que emiten “tokens del usuario”. Más abajo, se describen dos métodos que tales instituciones de confianza pueden seguir: un método “abierto” y un método “autorizado”. (Tenga en cuenta, sin embargo, que no todas las etapas, elementos y

características pueden ser necesarios en todas las modalidades).

### **Método Abierto para Emitir Tokens Respaldados por monedas**

Supongamos que existen dos bancos que desean participar en la cadena de bloques mediante la emisión de tokens de activos respaldados por divisas: Banco A y Banco B. Ambos bancos operan en un país llamado Freedonia, y ambos bancos usan Libras Freedonianas (FRP) como su moneda principal.

De acuerdo con al menos una modalidad de la presente invención, el Banco A y el Banco B emiten cada uno directamente sus propios tokens respaldados por monedas, FRP\_BankA y FRP\_BankB, respectivamente. No existirían limitaciones explícitas a la cantidad de tales tokens que los bancos puedan emitir.

El banco A agrega una orden comercial a la cadena de bloques que ofrece FRP\_BankA a cambio de FRP\_BankB de forma individual, o de forma casi individual (digamos, uno a 0.999). Igualmente, el Banco B agrega una orden comercial a la cadena de bloques que ofrece FRP\_BankB a cambio de FRP\_BankA de forma individual, o de forma casi individual. Esto permite que FRP\_BankB y FRP\_BankA sean intercambiables para la mayoría de las operaciones de cadena de bloques. Estas órdenes comerciales no coinciden entre sí o se liquidan automáticamente porque se les prohibiría hacerlo mediante algún mecanismo - potencialmente porque habría un pequeño margen entre estas ofertas (haciendo que las órdenes comerciales no sean capaces de coincidir), o porque ambos se marcarían como “en reposo”, o porque el contrato de autorización del token devolvería falso, o debido a una lista negra especificada con la orden comercial, o algún otro mecanismo en dependencia de la implementación o modalidad específica.

El Banco A y el Banco B también agregarían contratos inteligentes a la cadena de bloques, cada uno implementando una subrutina de “canje” que intercambia FRP\_BankB por FRP\_BankA de forma individual, y viceversa. Por ejemplo, la subrutina de “canje” del contrato inteligente del Banco A aceptaría cierta cantidad de FRP\_BankA y devolvería la misma cantidad de FRP\_BankB del saldo mantenido en el contrato. En al menos una modalidad, esta subrutina solo aceptaría rescates de los corresponsales del Banco A (es decir, el Banco B), y viceversa.

Se espera que tanto el Banco A como el Banco B ofrezcan moneda del mundo real fuera de la cadena a cambio de los tokens emitidos que se transfieren de vuelta a sus cuentas. Debido a las órdenes comerciales recíprocas enumeradas anteriormente, cada Banco puede “efectuar en efectivo” los tokens del otro, y los tokens de cada banco pueden usarse indistintamente en entornos comerciales. Otras organizaciones o individuos también pueden emitir tokens con etiquetas que contengan el

prefijo “FRP”. Por ejemplo, una cooperativa comercial llamada Coop A podría combinar sus recursos para emitir tokens FRP\_CoopA, y también podría distribuirlos a través de registros de transferencia (a cambio de bienes y/o servicios del mundo real o moneda dura) o a través de órdenes comerciales (ofreciendo tokens FRP\_CoopA a cambio de tokens FRP\_BankB o FRP\_BankA). Si Coop A puede mantener su propia reserva de tokens FRP\_BankB o FRP\_BankA, bloquear una porción de esos tokens en órdenes comerciales que ofrecen comercializar tales tokens a cambio de tokens FRP\_CoopA, entonces, los tokens FRP\_CoopA pueden efectivamente usarse indistintamente para FRP\_BankA y FRP\_BankB.

Al finalizar este proceso, un conjunto de tokens respaldados por FRP (“red de tokens”) estaría en circulación, capaz de actuar como sustituto de las Libras de Freedonianas en todas las actividades de la cadena de bloques.

El beneficio del enfoque anterior es que puede surgir orgánicamente de la autoasociación voluntaria de participantes independientes. Al extender crédito entre sí a través de órdenes comerciales recíprocas y subrutinas de canje, las entidades participantes sustancian su evaluación en cuanto a la riesgo de sus contrapartes; los emisores riesgosos tendrán problemas para ver sus tokens aceptados a cambio de otros tokens, o enfrentarán límites de facto a la cantidad de tokens que pueden emitir, y pueden ser cortados completamente del sistema si su riesgo es demasiado grande. Por lo tanto, los nuevos emisores pueden unirse voluntariamente de manera ad-hoc, pero deben ser patrocinados efectivamente por otros participantes, y los emisores riesgosos pueden ver su participación limitada o cancelada si pierden el respaldo de la contraparte.

Dicho esto, el enfoque anterior tiene limitaciones; principalmente en el sentido de que puede no cumplir fácilmente con los requisitos regulatorios que operan en algunas jurisdicciones. Además, hay áreas de riesgo que se encuentran en el enfoque anterior que pueden dar lugar a la pérdida de activos u otros problemas para los participantes, tal como los siguientes.

(1) Como se describió anteriormente, la autoridad supervisora o el organismo autorregulador que supervisa estos bancos y/o administra la política monetaria Freedoniana no participa en la cadena de bloques, por lo que no hay un token FRP paraguas. Como un resultado, el mercado de los tokens basados en PRF en virtud de este esquema está posiblemente en mayor riesgo de fracaso institucional que un enfoque más regulado.

Si alguna entidad que emite tokens de PRF no puede canjear sus tokens por activos del mundo real (como en una operación bancaria), los tokens emitidos por esa entidad pueden perder su valor. Los corresponsales tendrían que actuar rápidamente para rescindir las operaciones y eliminar los

contratos de canje o sus tokens se verían afectados.

Incluso en el caso de que la participación de un emisor esté limitada por los corresponsales a una fracción del valor total emitido, los tokens no canjeados en poder de los corresponsales pueden socavar su salud financiera. El valor de todos los tokens de la clase puede desacoplarse potencialmente del valor de la moneda subyacente, si se percibe que son más riesgosos que otros equivalentes de moneda.

Un riesgo sistémico representado por estas circunstancias es que el valor de todos los tokens de la clase puede desacoplarse de la moneda subyacente, lo que tiene efectos dominó en toda la cadena de bloques.

(2) Debido a que no hay restricciones técnicas impuestas sobre qué entidades pueden participar en la red de tokens de FRP, es concebible que las entidades mal calificadas puedan unirse. La participación en la red de tokens no es muy abierta - los participantes eligen a sus corresponsales, y los corresponsales de mala reputación pueden ser rechazados o eliminados de la red si hay un consenso para hacerlo entre los participantes. Sin embargo, un emisor no tendrá la capacidad de influir directamente en el riesgo que otro emisor representa para la red si los dos emisores no son corresponsales directos. Los participantes no solo aceptan el riesgo representado por sus corresponsales directos y contrapartes, sino también el riesgo representado por las contrapartes de sus contrapartes.

(3) Debido a que ninguna entidad tiene un reclamo para la cadena de “FRP” base que se usa como prefijo en las etiquetas para estos tokens, es posible que los tokens que no se han aceptado en la red de tokens usen etiquetas que sean engañosas. Por ejemplo, si la Compañía A no es aceptada como solvente por otros emisores y no puede entablar relaciones corresponsales, aún puede emitir su propio token FRP\_CompanyA. Esto puede ser engañoso para los usuarios.

Los registros de órdenes comerciales pueden usar caracteres comodín para coincidir con una amplia clase de tokens, no solo tipos de token único. Por ejemplo, FRP\_CoopA puede ofrecerse a cambio de “FRP\*”, que representa todos los tipos de token que están prefijados por la cadena “FRP”. Esto reduce la cantidad de registros de órdenes comerciales que deben enumerarse. Sin embargo, sin poder confiar en que la emisión de tokens mediante el uso de el prefijo “FRP” está de alguna manera controlada, el uso del comodín en este caso sería imposible.

(4) En virtud del escenario descrito, es posible que no se cumplan o exijan fácilmente otros requisitos regulatorios y gubernamentales, tal como los requisitos de KYC, AML y confidencialidad bancaria, o los requisitos de seguridad y solidez bien definidos, sin el cumplimiento voluntario de los emisores

de tokens. Por ejemplo, ¿pueden identificarse usuarios de tokens de alto volumen o alto valor (es decir, usuarios de alto riesgo)? ¿Se emiten tokens con suficientes activos fijos en poder del emisor? Siguiendo el método “abierto” de emitir tokens respaldados por monedas, se imponen pocos controles regulatorios a través de la cadena de bloques en sí. Las organizaciones que no están autorizadas por las autoridades reguladoras podrían emitir tokens que puedan dar la impresión de que los usuarios comunes son sólidos, pero no lo son.

### **Método Autorizado para Emitir Tokens Respaldados por monedas**

Determinadas características específicas de la invención pueden mitigar o eliminar estos riesgos e inquietudes.

Supongamos que existe un país diferente, Ruritania, cuya moneda se llama Corona (abreviado RUC). Por el contrario con Freedonia, el Banco Central Ruritano (“RCB”) desempeña un papel activo no solo en la supervisión de las actividades de cadena de bloques de las entidades participantes, sino también al ejercer control directo sobre la emisión de tokens mediante el uso de características de la invención tal como tipos de token “derivados” en combinación con contratos inteligentes y registros de órdenes comerciales. En otra jurisdicción, un rol similar podría ser desempeñado por un tipo diferente de entidad gubernamental, por una organización no gubernamental, por una asociación industrial o un organismo autorregulador, o por una corporación privada.

(1) RCB declara y define un nuevo tipo de token, con la etiqueta “RUC”, en un registro de génesis que tiene “suministro” establecido como “ilimitado” y “nueva emisión” establecido como “local”. La dirección de la cuenta local del registro de génesis es un contrato inteligente escrito por RCB para controlar y administrar la emisión de tokens respaldados por RUC. [¿por qué este contrato inteligente?]

(2) Este contrato inteligente ejecutable, del tipo que puede implementarse en un idioma como Solidity, implementa operaciones clave relacionadas con los tokens RUC.

(a) Una subrutina contractual denominada “crear” (o similar), que solo puede llamarse desde otra cuenta controlada por RCB y verificada dentro de la implementación del contrato:

Esta subrutina es responsable de declarar y asignar valor a tipos de tokens derivados de RCB. La etiqueta de tipo de token se pasaría como un argumento a la subrutina, junto con el valor del token (es decir, la cantidad de tokens que se emitirán) y la cuenta local a la que se emitirían los nuevos tokens.

La subrutina “crear” agregaría un registro de génesis a la cadena de bloques con la etiqueta, el valor y la cuenta local del tipo de token especificados, con la emisión establecida como “local” y el

suministro al “emisor original” o “fijo”. Este registro de génesis también designaría una subrutina de autorización que haría cumplir ciertas restricciones de KYC u otras restricciones de cumplimiento en el uso del token, por ejemplo, prohibiendo la transferencia por tokens a cuentas a las que no se les ha confirmado su identidad a través de una propuesta, o especificando una lista negra de cuentas. La subrutina “crear” también agregaría un registro de orden comercial que ofrece tokens RUC a cambio del nuevo tipo de moneda derivada, independientemente de cuál sea, con el contrato en sí mismo siendo la cuenta de destino para el registro de orden comercial. El valor de la orden comercial sería igual al valor total declarado para el nuevo tipo de token. La cuenta local del nuevo token se preserva como datos de contrato, almacenados junto con la etiqueta del nuevo token, de manera que pueda usarse por otras subrutinas para validar la cuenta que está invocando la subrutina. Debido a que RUC se creó con “suministro” establecido como “ilimitado”, el contrato inteligente puede emitir tantos tokens nuevos y tipos de token como sea necesario.

(b) Una subrutina contractual denominada “canje” (o similar) que permitirá que la cuenta de destino de un tipo de token derivado (y solo esa cuenta) devuelva cualquier token RUC\* a cambio del token derivado de la cuenta:

Mediante el uso de esta subrutina, los tokens derivados que son acumulados por el RCB a través de órdenes comerciales pueden distribuirse de vuelta al emisor. Es posible que los tokens derivados devueltos por la subrutina de “canje” no se devuelvan de forma individual a cambio de tokens RUC\*. RCB puede, por ejemplo, aplicar una tasa de interés efectiva y devolver más o menos tokens en dependencia de la duración del tiempo en que los tokens son mantenidos por el contrato. La relación de cuántos tokens derivados se devolverían a cambio de tokens de RUC\* sería determinada por RCB en base a sus objetivos de política: una tasa de interés negativa incentivaría un canje más rápido, lo que aumentaría el suministro de tokens en circulación, mientras que una tasa de interés positiva ralentizaría el canje, lo que disminuiría el suministro global de tokens.

(c) Un par de subrutinas (denominadas “aceptación” y “emisión”, o similares) que permiten que ciertas cuentas autorizadas (es decir, cuentas a las que se les han emitido tokens del usuario personalizados - bancos) obtengan tokens de RUC directamente a cambio de moneda fuerte, o que reciban monedas fuertes a cambio de RUC:

Estas subrutinas permitirían a un banco u otra entidad emitir o liberar más tokens RUC que el valor total de sus tokens derivados individuales.

(3) El RCB declara y define dos nuevos tipos de token, RCB.BankA y RCB.BankB, correspondientes a dos bancos que operan bajo su supervisión, el Banco A y el Banco B, respectivamente. Estos tipos

de token se crearían y emitirían a través de la subrutina “crear” del contrato, y en cada caso tendrían como cuenta local un contrato inteligente definido y controlado por el banco receptor. (Los tipos de token también pueden declararse o definirse para otro tipo de organización, no solo bancos; por ejemplo, las cooperativas de crédito, los corredores de bolsa y otras entidades corporativas también pueden recibir esta autorización).

También se emitiría un valor de token  $V$  total para el banco. Este valor  $V$  representaría el límite de la autoridad de un banco para prestar o emitir crédito a través de la emisión de tokens (ya sea a sí mismo o a otras partes) y lo más probable es que esté respaldado sobre una base fraccionaria por algún depósito independiente fuera de la cadena proporcionado al RCB. Si el banco emite el mismo tipo de token a cambio de depósitos en moneda, entonces el valor  $V$  también restringiría el monto de los depósitos que el banco puede emitir. Otro token, sin embargo, podría emitirse alternativamente a cambio de depósitos, por ejemplo, un segundo token derivado emitido específicamente al banco, o tokens RUC prestados o proporcionados de cualquier otra manera por RCB con el fin de aceptar depósitos.

Según los requisitos normativos, estos bancos deberán canjear cualquier token de RUC\* presentado por consumidores, empresas u otros bancos a cambio de efectivo o equivalente.

Al emitir diferentes tipos de tokens derivados para cada banco, el riesgo sistémico representado por un emisor determinado se identifica inmediatamente: es el valor de su tipo de token en circulación. Si un banco determinado está en riesgo de ser intervenido, el RCB puede intentar aislar este riesgo bloqueando la conversión del token derivado en el token base a través de órdenes comerciales. Alternativamente, podría alterar la subrutina de autorización de manera que bloquee las operaciones, potencialmente agregando la etiqueta de token a una lista negra mantenida por ese contrato. Si la RCB decide detener las operaciones de un tipo de token derivado particular (al finalizar o no renovar los registros de órdenes comerciales o bloquear las operaciones a través de la subrutina de autorización), entonces cualquier persona que tenga esos tokens ya no podrá realizar transacciones comerciales con esos tokens. No habría forma de usar el token RCB\* de tal banco para realizar un pago. El RCB puede entonces gestionar una liquidación ordenada que incluye posibles pagos de seguro de depósito a personas que todavía tienen los titulares de tokens RCB\* afectados.

(4) Los propios contratos inteligentes de los bancos (que serían iguales o similares para todos los bancos) encapsularían determinadas operaciones requeridas para controlar el uso de tokens RUC\*.

Por ejemplo, el contrato gestionado por el Banco A incluiría la siguiente funcionalidad.

Una subrutina contractual inteligente denominada “canje”, abierta a todas o casi todas las cuentas,

permitiría que las cuentas recibieran RUC a cambio de tokens RUC.BankA de forma individual.

En al menos una modalidad, una inspección de la procedencia del token por esta subrutina puede limitar qué cuentas lo invocan con éxito, dado que el uso y el historial de contribución de los tokens serían rastreables hasta algunos requisitos mínimos. (Por ejemplo, si el RUC atravesó la cuenta de un mal actor conocido registrado en una lista dentro del contrato inteligente, entonces el canje se bloquearía).

Una subrutina contractual denominada “retiro”, que solo puede llamarse desde una cuenta controlada única y separada del Banco A, permitiría al Banco A reabastecer sus tenencias de RUC.BankA al retirarlos de la RCB a través de la subrutina “canje” implementada por la RCB en su contrato, a cambio de tokens de RUC o RUC\* en poder del banco.

Una subrutina contractual denominada “depósito”, que solo puede llamarse desde una cuenta controlada única y separada del Banco A, permitiría al Banco A depositar sus participaciones de RUC.BankA (y potencialmente otros tokens de RUC\*) con RCB a través de la subrutina de “aceptación” del contrato inteligente de la RCB.

Una subrutina contractual llamada “emisión” permitiría al Banco A mover tokens RUC.BankA a otra cuenta, por ejemplo, a cambio de un depósito de efectivo fuera de la cadena, o para financiar un préstamo.

Una subrutina contractual llamada “aceptación” permitiría que el Banco A reciba tokens RUC.BankA (o posiblemente cualquier token RUC\*) - por ejemplo, como repago de un préstamo, o a cambio de un retiro de efectivo.

En al menos una modalidad, una inspección de la procedencia del token dentro del contrato inteligente puede restringir la cantidad de transferencias realizadas por tokens, o los tipos de cuentas que contribuyen al saldo del token (por ejemplo, si el propietario de la cuenta ha sido verificado o no). La procedencia de los tokens sería accesible a través de elementos del lenguaje completo de Turing usado para implementar el contrato inteligente. El contrato podría atravesar el gráfico de registros y cuentas que contribuyen al saldo de la cuenta de origen, y también inspeccionar los registros de propuesta y decisión relacionados con estas cuentas.

### **Mecanismo de Pagos**

El mecanismo mediante el cual estos registros y contratos inteligentes permitirían pagos y comercio, en al menos una modalidad de la presente invención, es el siguiente:

(1) Un banco, que desea emitir tokens respaldados por monedas, solicitaría a una autoridad central, la RCB en este caso, la autoridad para emitir tokens derivados del token RUC emitido por la RCB.

La cantidad de tokens emitidos al banco dependería, en parte, de los depósitos de reserva que el banco realice con el RCB. En este caso, el RCB invocaría la subrutina de “creación” de su contrato inteligente para autorizar al Banco A a emitir tokens RUC.BankA.

(2) Un consumidor intercambiaría efectivo (ya sea efectivo físico o efectivo en una cuenta bancaria) por una cantidad equivalente de tokens emitidos por un banco. Por ejemplo, el consumidor podría intercambiar 10 RUC físicos por 10 tokens RUC.BankA. El Banco A invoca la subrutina de “emisión” en su contrato inteligente, a favor de la cuenta de la cadena de bloques del consumidor.

(3) Un comerciante puede ofrecer un widget para la venta a un precio de 10 RUC. Dado que el comerciante desea aceptar tokens para el pago, el comerciante fija el precio del widget igual a 10 tokens RUC.

(4) El consumidor compraría este widget transfiriendo 10 tokens RUC.BankA al comerciante con un tipo de token de destino RUC. Cuando se procesa la transferencia, se liquida una transacción automática de manera que los 10 tokens RUC.BankA se convierten en 10 tokens RUC, que se agregan a la cuenta de la cadena de bloques del comerciante. Los 10 tokens RUC.BankA se agregan a la cuenta de la cadena de bloques de la RCB.

(5) El comerciante puede gastar estos tokens RUC para compras adicionales, o el comerciante puede depositar el monto en una cuenta bancaria. Debido a que él realiza operaciones bancarias en el Banco B, el comerciante transfiere los tokens al Banco B en una invocación de la subrutina de “aceptación” del contrato inteligente del banco. Luego, el banco acredita la cuenta bancaria del comerciante.

(5) El banco puede entonces invocar la subrutina de “retiro” en su propio contrato inteligente, lo que da como resultado que el contrato inteligente envíe los 10 tokens RUC a la subrutina de “canje” en el contrato inteligente RCB. Luego, el RCB devuelve al banco una cantidad equivalente de tokens RUC.BankB. Alternativamente, especialmente en el caso de que la RCB no tenga ningún token de RUC.BankB en su posesión, el banco puede enviar los 10 tokens de RUC a la subrutina de “aceptación” en el contrato inteligente de la RCB, lo que a su vez aumentaría la cantidad de tokens que el banco está autorizado a emitir por la RCB.

(6) Alternativamente, en caso de que el Banco B no canjee sus tokens RUC a cambio de los tokens RUC.BankB a un ritmo lo suficientemente rápido, el RCB puede enviar sus tokens RUC.BankB a la subrutina de “canje” en el contrato inteligente del banco. Si el banco no puede canjear estos tokens, esto podría dar lugar a una revisión más detallada del banco por parte del RCB.

En al menos una modalidad adicional, cada banco generaría registros de órdenes comerciales recíprocos que ofrecieran intercambiar los tokens que ellos mismos emitían a cambio de RUC. En

este caso, por ejemplo, el Banco A generaría una orden de transacción que ofreciera tokens RUC.BankA a cambio de tokens RUC, de forma individual o de forma casi individual, y el Banco B generaría una orden de transacción que ofreciera tokens RUC.BankB a cambio de tokens RUC, de forma individual o de forma casi individual. En este escenario, el comerciante puede fijar el precio de su widget en términos del tipo de token de su banco preferido, en este caso RUC.BankB, en lugar de RUC. Un pago de un cliente que posee tokens RUC.BankA a un comerciante que solicita tokens RUC.BankB se lograría a través de una cadena de registros de órdenes comerciales vinculados al registro de transferencia mediante el uso de un registro de liquidación, en el método descrito en otra parte en la presente descripción.

### **Controles de Cumplimiento Adicionales**

Pueden implementarse controles de cumplimiento adicionales al proporcionar herramientas de análisis a los empleados del banco y otras personas que administran interacciones con clientes que usan estos productos de cadena de bloques. Por ejemplo, la misma API que está disponible para contratos inteligentes con el fin de explorar el historial de cuenta, las cadenas de transferencias, la procedencia del token y los registros de propuesta y decisión estarían disponibles para billeteras y otras aplicaciones de GUI de cliente creadas para interactuar con la cadena de bloques. El personal de cumplimiento de un banco u otra organización recibiría una herramienta para explorar este historial.

Adicionalmente, los documentos de identidad proporcionados a los verificadores de confianza confirmaron la identidad de una cuenta a través del proceso de determinación de la propuesta podrían ponerse a disposición del personal del banco para confirmar la identidad de una persona que busca canjear u obtener tokens, o realizar alguna otra actividad que requiera confirmación de identidad.

Lo que conoce un contrato inteligente en la cadena es el resultado del consenso de determinación pública, y exactamente qué verificadores de confianza pueden haber contribuido con los registros de decisión a la cadena de bloques. La subrutina de autorización de un tipo de token (u otra subrutina restrictiva) puede implementarse para requerir un proceso de determinación de la propuesta para encontrar a favor de una propuesta de identidad, y/o que determinados verificadores de confianza contribuyeron a los registros de decisión a favor. Sin embargo, no tendrá acceso a materiales de prueba confidenciales.

Cualquier acción que requiera actividad fuera de la cadena - por ejemplo, cualquier intento de canjear u obtener tokens para monedas fuertes - puede incorporar una revisión humana que interactúe con los datos de la cadena de bloques para recuperar detalles de la cuenta (historial de la propuesta

asociado con la cuenta, historial de tokens). El cliente del banco puede proporcionar el nombre, la dirección a través de una ID emitida por el estado, y eso puede confirmarse como parte de la propuesta mediante la manipulación de la información de identificación física y la comparación con la propuesta.

Si se requiere una profundización para revisar la documentación de prueba original usada para confirmar la identidad, la comunicación fuera de banda entre el banco y el verificador (sujeto a NDA y otros términos comerciales) puede hacer que el verificador revele datos no cifrados al banco a través de una conexión de red encriptada o de cualquier otra manera segura. Los verificadores de confianza podrían responder a las solicitudes de datos de múltiples bancos - sujeto a los términos contractuales acordados tanto por los bancos como por los clientes del banco - y ayudarían a hacer que la identidad de las personas sea portátil en múltiples instituciones.

Este enfoque desacoplaría algún aspecto del proceso de KYC de una institución en particular y permitiría que se comparta información importante de cumplimiento entre instituciones que están vinculadas entre sí por el mismo cliente o la misma transacción. Podría realizarse un proceso similar para compartir información de la fuente de fondos entre instituciones.

### **Remesas Internacionales**

Dada la existencia de tokens respaldados por monedas como se describió anteriormente, las personas podrían enviar dinero internacionalmente. Un emisor individual en Ruritania, por ejemplo, podría enviar dinero a un beneficiario individual en Freedonia, efectivamente sin intermediarios.

[0003] En al menos una modalidad, un método de la presente invención para facilitar las remesas internacionales puede implementar las siguientes etapas, elementos y características (aunque no todas las etapas, elementos y características son necesarios en cada modalidad). En esta descripción, y a lo largo de esta sección, el ejemplo de monedas, bancos y personas de Ruritania y Fredonia se usa solo con fines ilustrativos; el proceso descrito sería el mismo si se reemplazaran estos elementos.

(1) Como requisito previo, debe establecerse un mercado activo para el intercambio de tokens de PRF y RUC en la cadena mediante el uso de registros de órdenes comerciales y registros de liquidación. Los compradores y vendedores de tokens posiblemente estarían buscando tales intercambios en apoyo del comercio entre los países.

(2) El emisor Ruritaniano compraría tokens de RUC por el monto que desea enviar. El pago puede realizarse en efectivo físico o a través de transferencia bancaria, y puede comprarse en un banco, de una compañía de remesas autorizada, de un cambio de monedas, de una persona o a través de alguna otra entidad. Los tokens comprados se transfieren a una cuenta controlada por el emisor.

En determinados entornos regulados, el emisor solo tendría permitido comprar tokens de un distribuidor legalmente autorizado y tendría que proporcionar documentación para cumplir con los requisitos de KYC y Confidencialidad Bancaria. Es posible que se requieran registros de propuestas y registros de decisiones para establecer definitivamente la identidad en la cadena.

(3) El emisor le pregunta a su beneficiario Freedoniano la dirección de la cuenta del beneficiario. Si el beneficiario aún no tiene una cuenta, el beneficiario debe crear una.

(4) El emisor agrega un registro de orden comercial a la cadena de bloques que ofrece el valor de sus tokens de RUC comprados a cambio de un número/fracción particular de tokens denominados en tokens de FRP (moneda beneficiaria).

Debido al enfoque laissez-faire de Freedonia, pueden introducirse tokens FRP fraudulentos a la cadena de bloques en cualquier momento, haciendo imposible el uso del comodín “\*” para coincidencia (esto se menciona anteriormente). El emisor tendría que elegir solo uno de los tokens de FRP en la red de tokens (digamos, FRP\_BankA) para nombrar en la orden comercial. Sin embargo, si no hay operaciones coincidentes para ese token en particular, el nodo de construcción de bloques puede combinar una tercera o cuarta orden comercial para atravesar el “gráfico” del token y liquidar la transacción.

La relación de tokens denominados en moneda extranjera con respecto a tokens denominados en moneda local representados por este registro de orden comercial necesitaría asemejarse a la relación de otras transacciones similares si se va a emparejar y liquidar; en otras palabras, el precio ofrecido necesitaría ser competitivo para el nodo de construcción de bloques para incentivarse a incorporar la orden. Será responsabilidad del emisor o del software del emisor (por ejemplo, una aplicación móvil de billetera) garantizar que la relación en oferta sea ventajosa para los nodos de construcción de bloques para incentivarlos a encontrar una coincidencia.

La cuenta de liquidación/destino incluida en el registro de orden comercial (y subsecuentemente el registro de liquidación, tal vez) sería la cuenta del beneficiario.

(5) Alternativamente, en lugar de generar tal orden comercial, el emisor puede realizar una transferencia directamente al beneficiario sin crear ningún registro de orden comercial. El emisor generaría un registro de transferencia con la cuenta de destino establecida como la cuenta del beneficiario y la moneda de destino establecida como la moneda del beneficiario.

En este caso, si el emisor no especifica un valor del token de destino, el nodo de construcción de bloques determinaría la cantidad específica de tokens denominados en moneda local que el beneficiario recibiría en base a la orden comercial con mejor precio disponible, bajo algunas

restricciones impuestas por el algoritmo de construcción de bloques y verificación. Si el emisor especifica un valor del token de destino, la transferencia coincidiría con una orden de transferencia con precio adecuado del libro de órdenes, o no se completaría.

(6) Si el beneficiario desea convertir los tokens denominados en moneda local a la moneda fuerte real, el beneficiario vendería esos tokens para la moneda fuerte, ya sea en una sucursal bancaria o en alguna otra ubicación. Por ejemplo, el beneficiario podría canjear el token en la institución que emitió originalmente el token.

Antes de canjear los tokens a cambio de monedas, la institución emisora podría inspeccionar la procedencia de los tokens que se canjean, para confirmar que no tienen una fuente sospechosa y que no pasaron por ninguna cuenta que sea de mala fe conocida. Esta capacidad de inspeccionar es un beneficio único de las cadenas de bloques, debido al hecho de que cada nodo tiene acceso al historial completo de transacciones de la cadena de bloques.

La invención aumenta esta capacidad al poner más actividad en cadena (transacciones) y al agregar registros de propuesta y decisión, que ponen metadatos verificados y verificables sobre una cuenta o transacción también en la cadena.

### **Facilitar las Operaciones de Compañías de Remesas con Licencia**

Lo anterior describe el proceso mediante el cual un emisor individual del país de Ruritania puede enviar dinero a un beneficiario individual en el país de Freedonia. Funciona porque ambas personas pueden obtener cuentas de cadena de bloques. Sin embargo, si un individuo no puede obtener y mantener una cuenta de cadena de bloques (quizás debido a problemas de seguridad o acceso a tecnología), las personas que envían y reciben dinero pueden trabajar a través de procesos estándares de la compañía de remesas con licencia que se mejoran mediante el uso de tokens de cadena de bloques habilitados por la invención. Específicamente, estos tokens respaldados por moneda mejoran el proceso de liquidación que se lleva a cabo detrás de escena.

En al menos una modalidad, un método de la presente invención para facilitar las operaciones de liquidación de las compañías de remesas con licencia puede implementar las siguientes etapas, elementos y características (aunque no todas las etapas, elementos y características son necesarios en cada modalidad). En esta descripción, nuevamente, el ejemplo de monedas, bancos y personas de Ruritania y Fredonia se usa solo con fines ilustrativos; el proceso descrito sería el mismo si estos elementos fueran reemplazados.

(1) El emisor Ruritania visita la ubicación física o en línea de una compañía de remesas con licencia (LRC) con la intención de enviar dinero. El emisor individual completa un formulario electrónico o

físico que detalla la información del emisor y del beneficiario y el monto de la remesa, y envía ese formulario a la LRC. El beneficiario nuevamente es un individuo que reside en Freedomia y recibe Libras Freedomianas.

(2) La LRC, a través de algún programa o subrutina, inspecciona los registros del estado de las órdenes comerciales y los registros de liquidación que coinciden con los tokens de ROC\* y FRP\* para determinar un tasa de cambio probable. A través de algún otro programa o subrutina, la LRC examina el historial reciente de la cadena de bloques para determinar los cargos de token que se cobran actualmente para calcular el costo probable de token de crear los registros necesarios en la cadena de bloques. Mediante algún algoritmo, la LRC aumenta la tasa de cambio (para evitar pérdidas y asegurar algunas ganancias) y cotiza un cargo y una tasa de cambio para transmitir el ROC al beneficiario, pagadero como FRP.

(3) El emisor individual le ofrece a la LRC el monto requerido en moneda fuerte, junto con cualquier información de identificación y documentación que la LRC considere necesaria (por ejemplo, ID del gobierno). La LRC, que ya posee tokens de ROC\*, crea un registro de pedido para intercambiar tokens de ROC\* por tokens de FRP\*, buscando adquirir tantos tokens de FRP\* como sea necesario para pagar al beneficiario individual identificado por el emisor. Esto es para garantizar que se fijen las ganancias y la tasa de cambio.

(4) La LRC ya estaría en posesión de tokens ROC\* porque (a) hubiera comprado esos tokens a cambio de moneda fuerte; o (b) habría recibido algunos tokens ROC\* derivados de un banco (como los tokens ROC.BankA) en la forma de préstamo.

(5) La LRC genera un número de reclamación único que el emisor proporcionaría a su beneficiario de Freedomian.

(6) La LRC tendría un arreglo previo con uno o ambos de Freedomian Bank A o Freedomian Bank B, de acuerdo con el cual el banco estaría obligado a canjear el número de reclamación único del beneficiario y pagar los fondos de la remesa en efectivo al beneficiario individual después de confirmar la identidad del beneficiario. El beneficiario individual visitaría el Freedomian Bank A o Bank B y presentaría el número de reclamación y la información de identificación. El banco buscaría el número de reclamación en un sistema de base de datos en red mantenido por la LRC, para confirmar el monto y la identidad del receptor. Después de pagar al beneficiario el monto de la remesa en efectivo (FRP), el banco registrará la remesa como pagada en la base de datos de la LRC, haciendo de manera simultánea un reclamo de reembolso.

(7) La LRC entonces reenviaría el monto de la moneda del PRC adeudado al banco en tokens de

PRC\* a través de la cadena de bloques.

(8) La LRC usaría el efectivo cobrado del emisor para comprar más tokens ROC\*, reponiendo su suministro.

Si la LRC opera a través de agentes de terceros en lugar de operar sus propias ubicaciones o a través de Internet, puede requerir que sus agentes adquieran y tengan una determinada cantidad de tokens ROC\* ellos mismos.

Inmediatamente después de generar una transacción de remesas, la LRC exigiría al agente que reenvíe los tokens de ROC\* como reembolso. Si los tokens no se envían a la LRC, la capacidad del agente para generar transacciones de remesas sería limitada.

Si no fuera a requerir que el agente reenviara los tokens inmediatamente después de la creación de una transacción de remesa, la LRC tendría que entonces extender el crédito del agente. El LRC arriesgaría al agente a no depositar o de cualquier otra manera no reenviar la moneda adeudada a la LRC por el agente.

El hecho de que el agente utilice tokens ROC\* para reembolsar a la LRC crea una oportunidad para que el agente reciba crédito de terceros no relacionados, o para que use el efectivo del emisor para adquirir tokens de fuentes de terceros. Por ejemplo, un agente que opera un cajero automático podría cargar efectivo en la máquina y recibir tokens ROC\* como reembolso de la red de cajeros automáticos cada vez que se realizan retiros. O bien, un agente que cobra cheques podría recibir tokens ROC\* como reembolso por cheques enviados electrónicamente al banco u otra entidad de compensación.

### **Caso de Uso 5: Cumplimiento de Valores**

Los reguladores de valores tienen amplia autoridad y poder para registrar, regular y supervisar la emisión, venta, comercialización, transferencia y compensación de valores dentro de sus diversas jurisdicciones. En dependencia de la jurisdicción en la que operen, los requisitos aplicados por varios reguladores difieren. Además, dentro de una jurisdicción específica, los requisitos pueden diferir para diferentes tipos de valores, o diferentes cantidades de valores, o para valores emitidos a o mantenidos por diferentes tipos de inversores o diferentes números de inversores.

Las cadenas de bloques y los sistemas de libros mayores distribuidos se han usado para emitir tokens que se asemejan a valores o que pueden caracterizarse como valores en virtud de la ley de valores en determinadas circunstancias. Sin embargo, debido a la incapacidad dentro de esos sistemas de proporcionar los controles detallados requeridos para garantizar el cumplimiento de las leyes y regulaciones de valores, estos tokens se han sometido a un escrutinio regulatorio cada vez mayor.

Además, dada la naturaleza global de las cadenas de bloques y otros sistemas de libros mayores distribuidos en todo Internet, ha demostrado que es difícil para los emisores de tokens garantizar que se respeten los límites jurisdiccionales.

En al menos una modalidad, la presente invención resuelve una cantidad de desafíos regulatorios enfrentados por las cadenas de bloques y otros sistemas de libros mayores distribuidos cuando se trata de la emisión, venta, transferencia y comercialización de tokens que se parecen o pueden caracterizarse como valores, o de cualquier otra manera pueden caer bajo un esquema regulatorio que reafirma el control integral sobre la emisión, venta, transferencia, asignación, uso o comercialización de tales tokens. Las siguientes etapas, elementos y características comprenden un método de la presente invención mediante el cual se facilita la emisión en cumplimiento de los tokens de seguridad en una cadena de bloques (aunque no todas las etapas, elementos y características son necesarios en cada modalidad).

(1) Con el fin de garantizar que se apliquen las normas adecuadas para el funcionamiento de cualquier token emitido a través de un sistema de libros mayores distribuidos donde el token pueda caracterizarse potencialmente como un valor, en al menos una modalidad, la jurisdicción de las cuentas participantes puede establecerse mediante el uso de registros de propuesta y determinación. Un usuario que desee conservar tales tokens establecería su jurisdicción (o la jurisdicción de su cuenta) mediante la emisión de un registro de propuesta que afirme que su cuenta está en una jurisdicción específica. Este registro de propuesta puede estar compuesto por una propuesta de identidad completa (como se describió en el caso de uso de verificación de identidad especificado en la presente descripción), o simplemente podría dirigirse a la cuestión de jurisdicción.

El procedimiento descrito como parte del caso de uso de verificación de identidad se seguiría para determinar la veracidad del registro de propuesta. Después de esta, las subrutinas contractuales inteligentes pueden inspeccionar el registro de la propuesta para determinar si se ha determinado o no una jurisdicción para una cuenta en particular.

En al menos una modalidad, la subrutina de autorización especificada para un token puede escribirse para detener o impedir la transferencia, asignación o comercialización de un token en caso de que una cuenta de beneficiario o receptor no se verifique como perteneciente a jurisdicciones autorizadas para retener el token. De manera similar, la subrutina de autorización puede hacer cumplir diferentes normas y restricciones relacionadas con la transferencia, asignación o negociación de tokens, en dependencia de a qué jurisdicciones pertenecen las cuentas que participan en la transacción.

Alternativamente, en al menos una modalidad, las mismas restricciones pueden implementarse

mediante el uso de filtros de autorización en lugar de subrutinas de autorización, al menos en parte. Tales filtros describirían patrones que solo coincidieran con cuentas que se ha confirmado que pertenecen a determinada jurisdicción, o que no pertenecen a determinadas jurisdicciones. Solo las cuentas con una declaración de la propuesta de confirmación de jurisdicción adjunta no coincidirían con el filtro y, de esta manera, se les permitiría conservar los tokens en cuestión.

(2) En ciertas jurisdicciones, las normas con respecto a la emisión, venta, transferencia, asignación o comercialización de valores se relajan en caso de que la cantidad de personas que poseen un valor sea limitada. En al menos una modalidad, un token que califica como un valor podría aprovechar un estándar tan relajado si la cantidad de personas que tienen un valor estuviera limitada a través de la subrutina de autorización implementada para ese token. La subrutina de autorización especificada para ese token realizaría una consulta contra los datos de la cadena de bloques para determinar la cantidad de cuentas que actualmente tienen un valor positivo para ese token.

Alternativamente, la subrutina de autorización actuaría para recuperar los datos de todos los registros de propuesta de reafirmación de identidad que hacen referencia a las cuentas que actualmente tienen un valor positivo para ese token. Esta información de identidad luego se analizaría para determinar la cantidad de titulares individuales de ese token, donde los titulares se contarían según las normas establecidas para la jurisdicción en cuestión.

La subrutina de autorización especificada para el token entonces bloquearía o impediría la creación de cualquier registro que pueda causar que la cantidad de individuos o la cantidad de cuentas que tienen el token superen la cantidad mínima permitida por las jurisdicciones dentro de las cuales se permite que se conserve el token. La subrutina de autorización, garantizaría de esta manera que el token nunca supere los límites operativos que permiten que el token sea evaluado por las autoridades reguladoras en virtud de un estándar relajado.

En al menos una modalidad, los filtros de autorización pueden implementar estas restricciones en lugar de subrutinas de autorización. Tales filtros de autorización proporcionarían un medio para filtrar por el conteo o la cantidad de cuentas que tienen tokens de un tipo en particular, y proporcionarían un medio para filtrar por si una cuenta en particular ya tenía esos tokens o no. La transferencia por tokens se permitiría solo si la cantidad de titulares de cuenta fuera inferior a un umbral determinado, o si la cuenta receptora ya tenía los tokens.

(3) En determinadas jurisdicciones, se imponen normas menos restrictivas sobre la emisión, transferencia, asignación o comercialización de valores que son mantenidos por personas que tienen algún ingreso mínimo anual o patrimonio neto financiero, o que de cualquier otra manera son capaces

de demostrar o afirmar algún nivel mínimo de sofisticación o perspicacia financiera. Estos individuos o entidades a menudo se denominan “inversores acreditados”. Siempre y cuando que la asignación, transferencia o comercialización de un token pueda restringirse a cuentas que solo tengan tales “inversores acreditados”, entonces ese token, incluso si se caracteriza como un valor, puede estar sujeto a estas normas menos restrictivas.

De acuerdo con al menos una modalidad, cualquier cuenta que desee adquirir o comerciar en tales tokens generaría un registro de propuesta (incluida una recompensa) que afirme que tal cuenta es mantenida por inversores acreditados. Algunas entidades o grupos de entidades calificados (“verificadores de acreditación”), al tomar conocimiento del registro de la propuesta, evalúan si tal cuenta está en poder de uno o más inversores acreditados. Por lo tanto, los verificadores de acreditación cumplen un propósito similar al de los verificadores de identidad en el proceso de verificación de identidad.

Incentivados por la recompensa, los verificadores de acreditación agregan registros de determinación al libro mayor distribuido. Otros participantes también pueden agregar registros de determinación adicionales, lo que resulta en la confirmación o denegación de la propuesta. Por lo tanto, las recompensas se asignan según el procedimiento de distribución de recompensas de la propuesta.

De acuerdo con al menos una modalidad, si se determina que una cuenta está en poder de inversores acreditados - lo que significa que, si la cuenta está vinculada a una propuesta que afirma satisfactoriamente que la cuenta está en poder de inversores acreditados - se considera una “cuenta acreditada”. Si no existe una propuesta que afirme satisfactoriamente que la cuenta es mantenida por inversores acreditados, entonces se considera una “cuenta no acreditada”.

Puede aplicarse una restricción a la asignación, transferencia o comercialización de un token mediante la implementación de esa restricción dentro de la subrutina de autorización del token, de manera que el token solo pueda mantenerse por cuentas acreditadas. La subrutina de autorización tendría referencia al registro de datos (o, en el caso de un registro de transacción, el conjunto de registros vinculados a esa transacción) que está destinada a autorizar o rechazar, y por lo tanto tendría referencia al tipo de token, la cuenta de origen, la cuenta de receptor y el valor de token especificado por ese registro, así como también otros datos relevantes.

La subrutina realiza una consulta contra los datos de la cadena de bloques para recuperar cualquier propuesta y registros de determinación relacionados con la(s) cuenta(s) del receptor especificada(s) en los registros de datos. La subrutina de autorización entonces bloquea o deshabilita la creación de cualquier registro nuevo que dé como resultado una cuenta no acreditada (es decir, cualquier cuenta

que no esté vinculada a ninguna propuesta que afirme satisfactoriamente que la cuenta es una cuenta acreditada) con el token en cuestión.

Por ejemplo, antes de que un registro de transferencia, un registro de orden comercial, un registro de liquidación o un registro de transacción se acepten como válidos, si el token del usuario que se transfiere o comercializa está restringido por una subrutina de autorización, entonces esa subrutina de autorización se usa para autorizar el registro antes de que pueda agregarse a la cadena de bloques/libro mayor. La subrutina usa la oportunidad para determinar si la(s) cuenta(s) del receptor especificada(s) en el/los registro(s) de transferencia, el/los registro(s) de orden comercial o el/los registro(s) de liquidación son cuentas acreditadas o no. Si se determina que no lo son, la subrutina no permitiría la inclusión de esos registros.

Alternativamente, en al menos una modalidad, la misma restricción puede implementarse mediante el uso de filtros de autorización en lugar de una subrutina de autorización. Tales filtros describirían patrones que solo coincidieran con cuentas a las que se ha adjuntado una declaración de la propuesta de acreditación. Cualquier cuenta que no tenga la declaración de la propuesta de acreditación adjunta no coincidiría con el filtro y se rechazaría.

### **Automatización de la Emisión de Tokens Restringidos**

En al menos una modalidad, puede construirse un contrato inteligente que emita automáticamente nuevos tokens de los tipos restringidos anteriores — restringidos por jurisdicción, o por número o por tipo de titular de cuenta, o por otra calificación — aprovechando el proceso de emisión de “token autorizado” detallado anteriormente.

Este contrato inteligente sería la “cuenta local” para un tipo de token que representa la totalidad de esta familia particular de tokens. El registro de génesis que origina este tipo de token tendría su campo de “nueva emisión” establecido en “local”, lo que permite que el contrato inteligente emita tokens derivados subordinados. El registro de génesis también tendría su campo de “suministro” establecido como “ilimitado”, lo que significa que este contrato inteligente tendría permitido emitir automáticamente tantos tokens nuevos como sea necesario.

El contrato inteligente implementaría una subrutina denominada “autorizar” (o similar), que actuaría como la subrutina de autorización para cualquier token del usuario creado por el contrato inteligente. Esta subrutina en sus argumentos se pasaría al registro propuesto (o, en el caso de una transacción, un conjunto de registros) que pretende autorizar o rechazar, y por lo tanto tendría referencia al tipo de token, la cuenta de origen, la cuenta de Receptor, el valor de token especificado por el registro y otros datos relevantes.

Como se describió anteriormente, esta subrutina de autorización recuperaría e inspeccionaría los registros de datos del libro mayor distribuido para determinar la cantidad de personas o el tipo de personas que tendrían el token si el registro o conjunto de registros propuestos se agregaran a la cadena de bloques (u otros datos que califiquen), y en base a la naturaleza de esos datos bloquearían o no permitirían la creación de cualquier registro que pueda superar las restricciones impuestas por las jurisdicciones dentro de las cuales el token está permitido. Por ejemplo, la subrutina puede no permitir la inclusión de registros de transferencia, orden comercial, liquidación o transacción si después de su inclusión el token sería retenido por individuos no acreditados, o demasiados individuos, o individuos en la jurisdicción incorrecta, etc., en dependencia de la restricción que se imponga.

En al menos una modalidad, el contrato inteligente también implementaría una subrutina llamada “emisión” (o similar), que subrutina aceptaría como argumentos una etiqueta, valor (es decir, conteo), fraccionamiento, y una cadena de caracteres que representa una jurisdicción o una lista de cadenas de caracteres que representan múltiples jurisdicciones (tales cadenas podrían ser códigos de país ISO 3166, por ejemplo). Al ser invocada por primera vez, esta subrutina generaría y agregaría al libro mayor un nuevo registro de génesis, con las siguientes características:

- “Etiqueta” según se especifica en los argumentos
- “Valor” según se especifica en los argumentos
- “Fraccional” según se especifica en los argumentos
- “Suministro” establecido en “emisor original” ▪ “Nueva emisión” establecido en “emisor original”
- “Subrutina de autorización” establecida para la subrutina de “autorizar” implementada por el mismo contrato inteligente.
- “Cuenta Local” a cualquier cuenta (o contrato inteligente) que haya invocado la subrutina (es decir, la cuenta que firmó la instrucción de invocación con su clave pública). Como un resultado, los nuevos tokens, tras la creación, se agregarán a la cuenta (o contrato inteligente) que haya invocado la subrutina.

Alternativamente, en al menos una modalidad, además de o en lugar de la subrutina de autorización especificada como porción del registro de génesis, la subrutina “emisión” agregaría filtros de autorización al registro de génesis. Estos filtros de autorización, por ejemplo, pueden definir patrones que se filtran y excluyen, por ejemplo, registros que envían tokens a cuentas que no tienen una declaración de la propuesta de jurisdicción que cumpla con los requisitos adjunta, o a cuentas que no tienen una declaración de la propuesta de acreditación que cumpla con los requisitos adjunta, o a

cuentas que no tienen tokens del mismo tipo cuando demasiadas cuentas ya tienen tokens del mismo tipo.

En al menos una modalidad, cuando se invoca por primera vez, la subrutina de “emisión” registraría la cuenta que invocó la subrutina (es decir, la cuenta que firmó la instrucción de invocación con su clave pública), qué cuenta se usaría para inicializar una lista de cuentas permitidas para expandir la emisión del tipo de token en cuestión (mediante la emisión de tokens adicionales) y destruir los tokens (mediante la eliminación de estos de circulación); esta lista de cuentas autorizadas se mantendría internamente mediante el contrato inteligente en sí mismo dentro de sus datos privados. La subrutina de “emisión”, tras la segunda invocación de un token determinado, ampliaría la emisión del tipo de token en cuestión al emitir tokens adicionales. Lo haría a través de un registro de génesis que asigna esos tokens a la misma cuenta que invoca la subrutina, pero solo si esa cuenta está incluida en la lista de cuentas permitidas para afectar la cantidad pendiente del token en cuestión (es decir, el suministro de ese token). Alternativamente, si el contrato inteligente tiene un saldo del token (por ejemplo, debido a una cantidad del token que se ha destruido previamente), puede emitir porción de los tokens a través de una transacción de transferencia desde su propia cuenta.

De manera similar, el contrato inteligente implementaría una subrutina llamada “destruir” (o similar), que reduciría la emisión del tipo de token en cuestión al eliminar de la circulación los tokens pagados en la subrutina. Lo haría al asignar esos tokens a su propia cuenta. Pero solo lo haría si la cuenta que invoca la subrutina se incluye en la lista de cuentas permitidas para afectar la cantidad pendiente del token.

El contrato inteligente también implementaría una subrutina llamada “update\_accounts” (o similar), que actualizaría la lista de cuentas permitidas para emitir tokens adicionales o destruir tokens.

Una variación de este contrato inteligente, en al menos una modalidad, implementaría una subrutina restrictiva de “autorizar” y una subrutina de “emisión” como se describió anteriormente, pero no administraría la creación o destrucción de tokens después de la emisión original. En su lugar, este contrato inteligente principal se extendería por cualquier contrato inteligente que actúe como la cuenta local para la emisión del nuevo token.

El contrato inteligente principal en este caso inspeccionaría la cuenta local para garantizar que sea un contrato inteligente que no invalide la implementación de subrutina de “autorizar” con una implementación diferente. Tal enfoque requeriría un contrato inteligente para inspeccionar problemáticamente la implementación de otro contrato inteligente (específicamente, si un contrato inteligente se extiende a otro, y si está anulando o remediante el uso de las mismas subrutinas

implementadas en el contrato inteligente principal).

Con el fin de garantizar que la subrutina de autorización correcta permanezca en vigencia, en al menos una modalidad, los registros de génesis secundarios (es decir, registros de génesis emitidos para aumentar el suministro después de la emisión inicial) pueden restringirse por el sistema de manera que no puedan nombrar una subrutina de autorización alternativa, y no puedan aflojar las restricciones impuestas por los filtros de autorización. Todos los tokens emitidos de un tipo determinado estarían sujetos a la misma subrutina de autorización y filtros que los especificados en el registro de génesis original para ese tipo.

El campo “suministro” del registro de génesis original usado para emitir el nuevo token se configuraría como “ilimitado” en lugar de “emisor original”.

### **Ofertas Públicas**

En muchas jurisdicciones, la emisión, la transferencia, la asignación, el uso y la comercialización de valores están altamente regulados en situaciones donde esos valores están disponibles para el público en general. Los reguladores de valores en tales jurisdicciones a menudo confían en organizaciones autorreguladoras de la industria, intercambios u otras entidades privadas para supervisar la emisión de valores dentro de estas jurisdicciones, y para garantizar que las normas con respecto a la emisión, transferencia, asignación y comercialización de valores dentro de la jurisdicción en cuestión.

Un token que califica como un valor, si se ofrece al público en general, puede tener que ser controlado en una manera que permita a los reguladores de valores, organizaciones autorregulatorias, intercambios u otras entidades diputadas (en conjunto, “organizaciones que hacen cumplir las reglas”) monitorear el cumplimiento y hacer cumplir las normas relacionadas con los valores. La emisión de tales tokens en al menos una modalidad de la presente invención puede restringirse a través del método de “tipo de token autorizado” especificado en otra parte en la presente descripción, y controlado por una organización oficial que hace cumplir las normas de una manera descrita en algunos detalles más abajo. Tras la emisión, la transferencia, asignación, uso y comercialización de tales tokens se controlarían a través de contratos inteligentes interoperativos, cuyos contratos inteligentes pueden implementar subrutinas de autorización que actúen para restringir tal actividad. Tales tokens también pueden controlarse mediante el uso de la característica de “filtro de autorización” de la presente invención, descrita en la presente descripción.

En al menos una modalidad de la presente invención, a una organización que hace cumplir las normas se le otorgaría autoridad sobre la emisión de nuevos tokens a través de un registro de génesis creado por una cuenta de autoridad superior. Este registro de génesis proporcionaría a la organización que

hace cumplir las normas el control sobre un tipo de token base, del cual obtendría privilegios clave con respecto a cómo se emiten y usan los nuevos tokens. Solo se emitirían tales tokens a las organizaciones que hacen cumplir las normas con autoridad oficial.

En al menos una modalidad, los registros de génesis que otorgan autoridad a organizaciones que hacen cumplir las normas pueden configurar algunos o todos sus campos de la siguiente manera:

- “Etiqueta” - el prefijo que se agregará a la etiqueta de todos los nuevos tipos de token, indicando la organización responsable de hacer cumplir las normas para todos esos tipos de token (por ejemplo, “REO”).
- “Valor” - cualquier número adecuado, muy probablemente 1; el valor es efectivamente irrelevante, debido al hecho de que estos tokens existen solo para transmitir autoridad a la organización que hace cumplir las reglas, y no se distribuirán.
- “Cuenta local” - el contrato inteligente usado por la organización que hace cumplir las normas para emitir nuevos tokens derivados conforme al mandato de la organización.
- “Cuenta Emisora” - la cuenta de autoridad superior que otorga a la organización que hace cumplir las normas su autoridad para emitir tokens autorizados en el libro mayor distribuido.
- “Suministro” - “ilimitado”.
- “Nueva emisión” - “local”, qué valor permitirá que la autoridad de la organización que hace cumplir las normas emita.
- “Subrutina de autorización” - puede usarse como control para habilitar/deshabilitar la emisión, transferencia, uso y comercialización de tokens emitidos por la organización que hace cumplir las normas, independientemente de su capacidad.
- “Filtros de autorización” - un medio alternativo que puede usarse en lugar de “Subrutina de Autorización” para habilitar/deshabilitar la emisión, transferencia, uso y comercialización de tokens emitidos por la organización que hace cumplir las normas, independientemente de lo que puedan ser.
- “Cascada de Autorización” - “verdadero”

El registro de génesis inicial anterior proporcionaría a la organización que hace cumplir las normas, a su vez, para generar registros de génesis adicionales, que se usarían para autorizar a las entidades que operan bajo su jurisdicción a emitir sus propios tokens de seguridad. Los tipos de tokens derivados declarados y definidos de esta manera por la organización que hace cumplir las normas se crearían a través de registros de génesis que tienen todas o algunas de las siguientes características, en al menos una modalidad:

- “Etiqueta” - el nombre bajo el cual se comercializaría el nuevo tipo de token (por ejemplo, “REO.TOKEN”)
- “Valor” - la cantidad de tokens que se crearán en la primera emisión
- “Cuenta Local” - el contrato inteligente al que se emitirán primero los tokens y desde el cual se distribuirán los tokens después de la emisión inicial.
- “Cuenta Emisora” - la cuenta o contrato inteligente controlado por el REO, cuya cuenta o contrato inteligente fue autorizado por el registro de génesis original para emitir nuevos tokens.
- “Suministro” - “emisor original”
- “Nueva Emisión” - “emisor original”
- “Subrutina de autorización” - puede usarse como control para habilitar/deshabilitar la transferencia, uso y comercialización del nuevo token derivado en cuestión. También puede hacer referencia a la subrutina de autorización del token base, de manera que una suspensión global de la actividad de la organización que hace cumplir las normas también suspendería la actividad de todos los tokens derivados que ha autorizado.
- “Filtros de Autorización” - pueden usarse como alternativa a la Subrutina de autorización como control para habilitar/deshabilitar la transferencia, uso y comercialización del nuevo token derivado en cuestión.
- “Cascada de Autorización” - “verdadero”

En al menos una modalidad, la organización que hace cumplir las normas emitiría tokens derivados solo a contratos inteligentes que se implementan en cumplimiento de los estándares establecidos por la organización que hace cumplir las reglas. La organización que hace cumplir las normas puede promulgar políticas de gobernanza del mundo real con respecto a la emisión de nuevos tokens, cuyas políticas pueden traducirse al código e implementarse dentro de subrutinas “emisión” implementadas por todos los contratos inteligentes que actúan como cuentas de origen para los tokens derivados recientemente emitidos. Si estos contratos inteligentes son inmutables, entonces una inspección manual o asistida por computadora realizada por la organización que hace cumplir las normas antes de que se autoricen los tokens derivados garantizaría que el contrato inteligente cumpla con esas políticas. Además, los controles que la organización que hace cumplir las normas puede requerir que se implementen en la subrutina de autorización podrían verificarse mediante inspección manual.

Para facilitar la aplicación de sus políticas de gobernanza, la organización que hace cumplir las reglas, en al menos una modalidad de la presente invención, puede generar registros de génesis que

requieren que tanto (a) la subrutina de autorización de todos los tokens base pase, como (b) la subrutina de autorización de cada token derivado específico pase, antes de que un registro pueda incorporarse en la cadena de bloques. Esto puede lograrse por un lado a través del uso de una configuración de “cascada de autorización” dentro de los registros de génesis como se describió en la presente descripción, o por el otro al hacer que las subrutinas de autorización de tipos de token derivados invocan las subrutinas de autorización de tipos de token base.

De esta manera, como se describió anteriormente, en al menos una modalidad, el comportamiento y el uso de tokens de valores emitidos bajo la autoridad de una organización que hace cumplir las normas puede regularse, y las políticas de gobernanza promulgadas por una organización que hace cumplir las normas pueden hacerse cumplir. Los tokens derivados emitidos por una organización que hace cumplir las normas están restringidos en su uso por subrutinas de autorización y filtros de autorización especificados en los registros de génesis que autorizan estos tokens derivados. Estas subrutinas de autorización y filtros de autorización harían cumplir las normas de comercialización y transferencia, así como también otras políticas de gobernanza impuestas por la organización que hace cumplir las normas.

Las reglamentaciones y normas específicas que pueden exigirse mediante el uso de estas herramientas son variadas e ilimitadas, y pueden dejarse a las organizaciones específicas que usan las herramientas descritas en la presente descripción. Sin embargo, de todas maneras es posible proporcionar ejemplos de mecanismos específicos que pueden emplearse para implementar políticas específicas.

(1) Mediante la inspección de la jurisdicción residente de las cuentas que intentan comerciar con el token derivado, las subrutinas y filtros de autorización pueden limitar la negociación a cuentas residentes solo en jurisdicciones sancionadas oficialmente. Las jurisdicciones prohibidas por la organización que hace cumplir las normas pueden ser excluidas de la actividad comercial para los tokens autorizados.

(2) El comportamiento y las actividades de todos los tokens derivados emitidos en virtud de la autoridad de la organización que hace cumplir las normas pueden controlarse de manera centralizada al otorgar una única subrutina de autorización maestra implementada por la organización que hace cumplir las reglas, la autoridad para controlar la actividad comercial de todos los tokens derivados emitidos en virtud de su competencia, todo de una sola vez. Las subrutinas de autorización individual de tokens derivados complementarían su propia lógica invocando esta subrutina de autorización global, rechazando cualquier registro nuevo que corra en conflicto con esta subrutina maestra. O

bien, esta subrutina maestra podría agregarse como la subrutina de autorización del bloque de génesis original que otorga autoridad a la organización que hace cumplir las reglas, con la cascada de autorización activada.

Se podría implementar una suspensión de operaciones globales para todos los tokens autorizados por una organización que hace cumplir las normas en particular mediante el uso de un indicador Booleano almacenada dentro de los datos de este contrato inteligente maestro. Al cambiar el valor almacenado en este indicador, la subrutina de autorización maestra arrojaría alternativamente verdadero y falso, controlando si la actividad comercial global está habilitada o desactivada en toda la red de nodos para los tokens derivados.

Una suspensión dirigida de la transferencia, uso o comercialización de un token derivado particular - que se ocupa de toda la circulación del token - es posible agregando ese token a una lista de tokens “suspendidos” mantenidos por el contrato inteligente que contiene la subrutina de autorización. Cualquier transferencia, orden comercial o registro de liquidación que haga referencia a tal token se rechazaría debido al hecho de que el token aparece en tal lista.

Alternativamente, tales suspensiones pueden lograrse mediante el uso de registros de propuestas en combinación con filtros de autorización incluidos con el bloque de génesis original que otorga autoridad a la organización que hace cumplir las normas. Para suspender la comercialización de todos sus tokens derivados, por ejemplo, la organización que hace cumplir las normas puede emitir un registro de propuesta autodecidida cuya declaración de la propuesta se adjunta al tipo de token base propio de la organización, declarando la comercialización suspendida. Un filtro de autorización asociado con ese tipo de token base bloquearía la actividad hasta que esa propuesta haya caducado. Puede usarse una declaración de la propuesta similar que se adjunta a los tipos de tokens derivados individuales para poner fin al comercio de esos tokens derivados individuales.

(3) La reautorización explícita de un token - por ejemplo, en espera del cumplimiento total de los requisitos de informes periódicos - puede lograrse de acuerdo con al menos una modalidad a través del uso de una fecha de cargo para cada tipo de token (donde la fecha de cargo se implementa como un número del bloque o altura del bloque particular). Cuando se cumplen los requisitos de cumplimiento para un período de tiempo determinado, la organización que hace cumplir las normas actualiza la fecha de cargo.

Tal fecha de cargo puede adjuntarse a un token derivado mediante el uso de un registro de propuesta autodefinida creado por la organización que hace cumplir las reglas. Si la fecha de cargo o la altura del bloque no se actualizan antes de que pase esa fecha, entonces la subrutina o filtro de autorización

devolvería un valor “falso”, lo que haría que la actividad de intercambio y transferencia de ese token se detuviera automáticamente. Alternativamente, la fecha podría ser un punto de datos almacenado dentro de un contrato inteligente que puede actualizarse a través de una llamada de subrutina.

(4) El cambio de normas y el cambio de los requisitos normativos pueden adaptarse al hacer que este contrato inteligente global sea intercambiable a través de un contrato de giro o un contrato de registro. Las subrutinas de autorización que controlan cada uno de los tipos de tokens derivados harían referencia al contrato de giro o registrador, que a su vez haría referencia al contrato inteligente global y la subrutina de autorización global a través de la dirección del contrato inteligente global.

Cuando se requiere autorización, la subrutina de autorización individual invocaría la subrutina de autorización implementada en el contrato giratorio al que se hace referencia, que a su vez invocaría la subrutina de autorización implementada en el contrato inteligente global al que se hace referencia. El contrato dinámico mantendría una lista de cuentas permitidas para redirigir su referencia al contrato inteligente global, de manera que el contrato inteligente global pueda cambiarse por una implementación alternativa. Las subrutinas se implementarían dentro del contrato dinámico que permite este intercambio contractual y que permite una actualización de la lista de cuentas permitidas para hacer tal cambio.

(5) Cualquier otra limitación, regla o requisito puede imponerse sobre la emisión, transferencia, asignación, uso y comercialización de cualquier otro tipo o categoría de token de seguridad a través de un proceso comparable que representa la naturaleza única de esa limitación, regla o requisito en particular. La emisión de tokens puede restringirse a la subrutina de un contrato que ha sido explícitamente autorizado para emitir tokens autorizados a través de una cadena de registros de génesis. La naturaleza y las cualidades de las personas que controlan las cuentas participantes pueden verificarse a través del proceso de determinación de la propuesta, y pueden tenerse en cuenta mediante un contrato inteligente a través de la inspección de registros de la propuesta y registros de determinación. Cualquier regla específica con respecto a la emisión, transferencia, asignación, uso y comercialización de un tipo de token puede codificarse en la implementación de contratos inteligentes y subrutinas y filtros de autorización.

Se entiende que las disposiciones descritas anteriormente son simplemente ilustrativas de las muchas modalidades específicas posibles, que representan aplicaciones de la presente invención. Pueden diseñarse fácilmente numerosos y variados arreglos de acuerdo con los principios de la presente invención sin apartarse del espíritu y el alcance de la invención.