(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 840 071**

(51) Int. CI.:

***G06F 16/27*** (2009.01)
***G06F 9/54*** (2006.01)
***G06F 3/06*** (2006.01)
***G06F 7/00*** (2006.01)
***G06F 12/08*** (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **26.08.2016 E 18177129 (6)**

(97) Fecha y número de publicación de la concesión europea: **25.11.2020 EP 3418915**

(54) Título: **Métodos y aparato para una base de datos distribuida dentro de una red**

(30) Prioridad:

**28.08.2015 US 201562211411 P**
**06.01.2016 US 201614988873**
**12.05.2016 US 201615153011**
**02.06.2016 US 201662344682 P**
**08.07.2016 US 201615205688**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
**06.07.2021**

(73) Titular/es:

**SWIRLDS, INC. (100.0%)**
**3400 N Central Expwy, Ste. 470**
**Richardson, TX 75080, US**

(72) Inventor/es:

**BAIRD, LEEMON C., III**

(74) Agente/Representante:

**SÁEZ MAESO, Ana**

## DESCRIPCIÓN

Métodos y aparato para una base de datos distribuida dentro de una red

Antecedentes

Realizaciones descritas en este documento se relacionan en general con un sistema de base de datos y más particularmente con métodos y aparato para implementar un sistema de base de datos a través de múltiples dispositivos en una red.

Algunos sistemas de bases de datos distribuidas conocidos intentan lograr un consenso para valores dentro de los sistemas de bases de datos distribuidas (por ejemplo, con respecto al orden en el cual se producen las transacciones). Por ejemplo, un juego multijugador en línea puede tener muchos servidores de cálculo a los que los usuarios pueden acceder para jugar el juego. Si dos usuarios intentan recoger un ítem específico en el juego al mismo tiempo, entonces es importante que los servidores dentro del sistema de base de datos distribuida finalmente lleguen a un acuerdo sobre cuál de los dos usuarios recogió el ítem primero.

Tal consenso distribuido puede ser manejado mediante métodos y/o procesos tales como el algoritmo de Paxos o sus variantes. Bajo tales métodos y/o procesos, un servidor del sistema de base de datos es configurado como el "líder", y el líder decide el orden de eventos. Los eventos (por ejemplo, dentro de juegos multijugador) son reenviados al líder, el líder elige un orden para los eventos, y el líder radiodifunde ese orden a los otros servidores del sistema de base de datos.

Sin embargo, tales enfoques conocidos usan un servidor operado por una parte (por ejemplo, servidor de gestión central) en el que confían los usuarios del sistema de base de datos (por ejemplo, jugadores de juegos). Por consiguiente, existe una necesidad de métodos y aparato para un sistema de base de datos distribuida que no requiera un líder o un tercero de confianza para operar el sistema de base de datos.

Otras bases de datos distribuidas están diseñadas para no tener líder, pero son ineficientes. Por ejemplo, tal base de datos distribuida se basa en una estructura de datos de "cadena de bloques", que puede lograr consenso. Sin embargo, tal sistema puede ser limitado a un pequeño número de transacciones por segundo en total, para todos los participantes juntos (por ejemplo, 7 transacciones por segundo), lo cual es insuficiente para un juego a gran escala o para muchas aplicaciones tradicionales de bases de datos. Por consiguiente, existe una necesidad de un sistema de base de datos distribuida que logre consenso sin un líder, y que sea eficiente.

"Tendermint: Consensus Without Mining por Kwon J" (XP055493960) explica cómo las criptomonedas de prueba de trabajo requieren un gasto masivo de energía, son lentas para confirmar transacciones y que su seguridad es difícil de cuantificar. Propone una solución al problema de consenso de blockchain que no requiere minería mediante la adaptación de una solución existente al Problema de los Generales Bizantinos.

Resumen

De acuerdo con la invención, se proporciona un método de acuerdo con la reivindicación 1. Las características opcionales se definen mediante las reivindicaciones dependientes.

Breve descripción de los dibujos

La figura 1 es un diagrama de bloques de alto nivel que ilustra un sistema de base de datos distribuida, de acuerdo con una realización.

La figura 2 es un diagrama de bloques que ilustra un dispositivo de cálculo de un sistema de base de datos distribuida, de acuerdo con una realización.

Las figuras 3-6 ilustran ejemplos de un *hashDAG*, de acuerdo con una realización.

La figura 7 es un diagrama de flujo que ilustra un flujo de comunicación entre un primer dispositivo de cálculo y un segundo dispositivo de cálculo, de acuerdo con una realización.

La figura 8 es un diagrama de flujo que ilustra un flujo de comunicación entre un primer dispositivo de cálculo y un segundo dispositivo de cálculo, de acuerdo con una realización.

Las figuras 9a-9c son diagramas vectoriales que ilustran ejemplos de vectores de valores.

Las figuras 10a-10d son diagramas vectoriales que ilustran ejemplos de vectores de valores que son actualizados para incluir nuevos valores.

La figura 11 es un diagrama de flujo que ilustra la operación de un sistema de base de datos distribuida, de acuerdo con una realización.

La figura 12 es un diagrama de flujo que ilustra la operación de un sistema de base de datos distribuida, de acuerdo con una realización.

La figura 13 es un diagrama de flujo que ilustra la operación de un sistema de base de datos distribuida, de acuerdo con una realización.

La figura 14 es un ejemplo de un *hashDAG*, de acuerdo con una realización.

La figura 15 es un ejemplo de un *hashDAG*, de acuerdo con una realización.

Las figuras 16a-16b ilustran un método de consenso de ejemplo para usar con un *hashDAG*, de acuerdo con una realización.

Las figuras 17a-17b ilustran un método de consenso de ejemplo para usar con un *hashDAG*, de acuerdo con otra realización.

Descripción detallada

En algunas realizaciones, un aparato incluye una instancia de una base de datos distribuida en un primer dispositivo de cálculo configurado para ser incluido dentro de un conjunto de dispositivos de cálculo que implementan la base de datos distribuida a través de una red acoplada operativamente al conjunto de dispositivos de cálculo. El aparato también incluye un procesador acoplado operativamente a la memoria que almacena la instancia de la base de datos distribuida. El procesador está configurado para definir, por primera vez, un primer evento vinculado a un primer conjunto de eventos. El procesador está configurado para recibir, por segunda vez después de la primera vez y desde un segundo dispositivo de cálculo del conjunto de dispositivos de cálculo, una señal que representa un segundo evento (1) definido por el segundo dispositivo de cálculo y (2) vinculado a un segundo conjunto de eventos. El procesador está configurado para identificar un orden asociado con un tercer conjunto de eventos con base en al menos un resultado de un protocolo. Siendo cada evento del tercer conjunto de eventos de al menos uno del primer conjunto de eventos o del segundo conjunto de eventos. El procesador está configurado para almacenar en la instancia de la base de datos distribuida el orden asociado con el tercer conjunto de eventos.

En algunas instancias, cada evento del tercer conjunto de eventos está asociado con un conjunto de atributos (por ejemplo, número de secuencia, número de generación, número de ronda, número recibido, y/o marca de tiempo, etc.). El resultado del protocolo puede incluir un valor para cada atributo del conjunto de atributos para cada evento del tercer conjunto de eventos. El valor para un primer atributo del conjunto de atributos puede incluir un primer valor numérico y el valor de un segundo atributo del conjunto de atributos puede incluir un valor binario asociado con el primer valor numérico. El valor binario para el segundo atributo (por ejemplo, un valor de incremento de ronda) para un evento del tercer conjunto de eventos puede basarse en si una relación entre ese evento y un cuarto conjunto de eventos vinculados a ese evento satisface un criterio (por ejemplo, un número de eventos fuertemente identificados por ese evento). Cada evento del cuarto conjunto de eventos es (1) un antepasado del evento del tercer conjunto de eventos y (2) asociado con un primer atributo común como los eventos restantes del cuarto conjunto de eventos (por ejemplo, un número de ronda común, una indicación de ser un primer evento de ronda R, etc.). El primer atributo común puede ser indicativo de una primera instancia de que un evento definido por cada dispositivo de cálculo del conjunto de dispositivos de cálculo está asociado con un primer valor particular (por ejemplo, una indicación de ser un primer evento de ronda R, etc.).

El valor para un tercer atributo (por ejemplo, un número de ronda recibido) del conjunto de atributos puede incluir un segundo valor numérico con base en una relación entre el evento y un quinto conjunto de eventos vinculados al evento. Cada evento del quinto conjunto de eventos es un descendiente del evento y está asociado con un segundo atributo común (por ejemplo, es famoso) como los eventos restantes del quinto conjunto de eventos. El segundo atributo común puede estar asociado con (1) un tercer atributo común (por ejemplo, ser un primer evento de ronda R o un testigo) indicativo de una primera instancia un segundo evento definido por cada dispositivo de cálculo del conjunto de dispositivos de cálculo está asociado con un segundo valor particular diferente del primer valor particular y (2) un resultado con base en un conjunto de indicaciones. Cada indicación del conjunto de indicaciones puede ser asociado con un evento de un sexto conjunto de eventos. Cada evento del sexto conjunto de eventos puede ser asociado con un cuarto atributo común indicativo de una primera instancia un tercer evento definido por cada dispositivo de cálculo del conjunto de dispositivos de cálculo está asociado con un tercer valor particular diferente del primer valor particular y el segundo valor particular. En algunas instancias, el primer valor particular es un primer entero (por ejemplo, un primer número de ronda R), el segundo valor particular es un segundo entero (por ejemplo, un segundo número de ronda, R+n) mayor que el primer entero y el tercer valor particular es un tercer entero (por ejemplo, un tercer número de ronda, R+n+m) mayor que el segundo entero.

En algunas realizaciones, un aparato incluye una memoria y un procesador. La memoria incluye una instancia de una base de datos distribuida en un primer dispositivo de cálculo configurado para ser incluido dentro de un conjunto de dispositivos de cálculo que implementa la base de datos distribuida a través de una red acoplada operativamente al conjunto de dispositivos de cálculo. El procesador está acoplado operativamente a la memoria que almacena la instancia de la base de datos distribuida y está configurado para recibir una señal que representa un evento vinculado a un conjunto de eventos. El procesador está configurado para identificar un orden asociado con el conjunto de eventos

con base al menos en un resultado de un protocolo. El procesador está configurado para almacenar en la instancia de la base de datos distribuida el orden asociado con el conjunto de eventos.

En algunas realizaciones, un medio legible por procesador no transitorio almacena código que representa instrucciones que van a ser ejecutadas por un procesador para recibir una señal que representa un evento vinculado a un conjunto de eventos e identificar un orden asociado con el conjunto de eventos con base en una ronda asociada con cada evento del conjunto de eventos y una indicación de cuando incrementar la ronda asociada con cada evento. El código incluye además código para hacer que el procesador almacene, en una instancia de una base de datos distribuida en un primer dispositivo de cálculo configurado para ser incluido dentro de un conjunto de dispositivos de cálculo que implementa la base de datos distribuida a través de una red acoplada operativamente al conjunto de dispositivos de cálculo, el orden asociado con el conjunto de eventos. La instancia de la base de datos distribuida está acoplada operativamente al procesador.

En algunas realizaciones, una instancia de una base de datos distribuida en un primer dispositivo de cálculo puede estar configurada para ser incluida dentro de un conjunto de dispositivos de cálculo que implementa la base de datos distribuida a través de una red acoplada operativamente al conjunto de dispositivos de cálculo. El primer dispositivo de cálculo almacena múltiples transacciones en la instancia de una base de datos distribuida. Un módulo de convergencia de base de datos puede ser implementado en una memoria o un procesador del primer dispositivo de cálculo. El módulo de convergencia de base de datos puede ser acoplado operativamente con la instancia de la base de datos distribuida. El módulo de convergencia de base de datos puede estar configurado para definir, por primera vez, un primer evento vinculado a un primer conjunto de eventos. Cada evento del primer conjunto de eventos es una secuencia de bytes y está asociado con (1) un conjunto de transacciones de múltiples conjuntos de transacciones, y (b) un orden asociado con el conjunto de transacciones. Cada transacción del conjunto de transacciones es de las múltiples transacciones. El módulo de convergencia de base de datos puede estar configurado para recibir, por segunda vez después de la primera vez y desde un segundo dispositivo de cálculo del conjunto de dispositivos de cálculo, un segundo evento (1) definido por el segundo dispositivo de cálculo y (2) vinculado a un segundo conjunto de eventos. El módulo de convergencia de base de datos puede estar configurado para definir un tercer evento vinculado al primer evento y al segundo evento. El módulo de convergencia de base de datos puede estar configurado para identificar un orden asociado con un tercer conjunto de eventos con base al menos en el primer conjunto de eventos y el segundo conjunto de eventos. Cada evento del tercer conjunto de eventos es de al menos uno del primer conjunto de eventos o del segundo conjunto de eventos. El módulo de convergencia de base de datos puede estar configurado para identificar un orden asociado con las múltiples transacciones con base al menos en (1) el orden asociado con el tercer conjunto de eventos y (2) el orden asociado con cada conjunto de transacciones de los conjuntos múltiples de transacciones. El módulo de convergencia de base de datos puede estar configurado para almacenar en la instancia de la base de datos distribuida el orden asociado con las múltiples transacciones almacenadas en el primer dispositivo de cálculo.

En algunas realizaciones, una instancia de una base de datos distribuida en un primer dispositivo de cálculo puede estar configurada para ser incluida dentro de un conjunto de dispositivos de cálculo que implementa la base de datos distribuida a través de una red acoplada operativamente al conjunto de dispositivos de cálculo. Un módulo de convergencia de base de datos puede ser implementado en una memoria o un procesador del primer dispositivo de cálculo. El módulo de convergencia de base de datos puede estar configurado para definir, por primera vez, un primer evento vinculado a un primer conjunto de eventos. Cada evento del primer conjunto de eventos es una secuencia de bytes. El módulo de convergencia de base de datos puede estar configurado para recibir, por segunda vez después de la primera vez y desde un segundo dispositivo de cálculo del conjunto de dispositivos de cálculo, un segundo evento (1) definido por el segundo dispositivo de cálculo y (2) vinculado a un segundo conjunto de eventos. Cada evento del segundo conjunto de eventos es una secuencia de bytes. El módulo de convergencia de base de datos puede estar configurado para definir un tercer evento vinculado al primer evento y al segundo evento. El módulo de convergencia de base de datos puede estar configurado para identificar un orden asociado con un tercer conjunto de eventos con base al menos en el primer conjunto de eventos y el segundo conjunto de eventos. Cada evento del tercer conjunto de eventos es de al menos uno del primer conjunto de eventos o del segundo conjunto de eventos. El módulo de convergencia de base de datos puede estar configurado para almacenar en la instancia de la base de datos distribuida el orden asociado con el tercer conjunto de eventos.

En algunas realizaciones, los datos asociados con una primera transacción pueden ser recibidos en un primer dispositivo de cálculo desde un conjunto de dispositivos de cálculo que implementan una base de datos distribuida a través de una red acoplada operativamente al conjunto de dispositivos de cálculo. Cada dispositivo de cálculo del conjunto de dispositivos de cálculo tiene una instancia separada de la base de datos distribuida. Un primer valor de orden de transacción asociado con la primera transacción puede ser definido por primera vez. Datos asociados con una segunda transacción pueden ser recibidos desde un segundo dispositivo de cálculo del conjunto de dispositivos de cálculo. Un conjunto de transacciones puede ser almacenado en la instancia de la base de datos distribuida en el primer dispositivo de cálculo. El conjunto de transacciones puede incluir al menos la primera transacción y la segunda transacción. Puede ser seleccionado un conjunto de valores de orden de transacción que incluyen al menos el primer valor de orden de transacción y un segundo valor de orden de transacción por segunda vez después de la primera vez. El segundo valor de orden de transacción puede ser asociado con la segunda transacción. Una variable de estado de base de datos puede ser definida con base en al menos el conjunto de transacciones y el conjunto de valores de orden de transacción.

En algunas realizaciones, un método incluye recibir un primer evento de una instancia de una base de datos distribuida en un primer dispositivo de cálculo de un conjunto de dispositivos de cálculo que implementan la base de datos distribuida a través de una red acoplada operativamente al conjunto de dispositivos de cálculo. El método incluye además definir, con base en el primer evento y un segundo evento, un tercer evento. El tercer evento está vinculado a un conjunto de eventos. Un valor de orden puede ser definido para un cuarto evento con base al menos en parte en un valor de participación colectiva asociado con el conjunto de eventos que cumplen un criterio de valor de participación. El valor de orden puede ser almacenado en una instancia de la base de datos distribuida en un segundo dispositivo de cálculo del conjunto de dispositivos de cálculo. En algunas realizaciones, el método incluye además calcular el valor de participación colectiva con base en una suma de un conjunto de valores de participación. Cada valor de participación del conjunto de valores de participación está asociado con una instancia de la base de datos distribuida que definió un evento del conjunto de eventos.

En algunas realizaciones, un método incluye recibir un primer evento de una instancia de una base de datos distribuida en un primer dispositivo de cálculo de un conjunto de dispositivos de cálculo que implementan la base de datos distribuida a través de una red acoplada operativamente al conjunto de dispositivos de cálculo. El método incluye además definir, con base en el primer evento y un segundo evento, un tercer evento y determinar un primer conjunto de eventos con base al menos en parte en el tercer evento. Cada evento del primer conjunto de eventos es a) identificado por un segundo conjunto de eventos y b) asociado con un primer número de ronda. Un valor de participación colectiva asociado con el segundo conjunto de eventos cumple con un primer criterio de valor de participación y cada evento del segundo conjunto de eventos (1) está definido por una instancia diferente de la base de datos distribuida y (2) es identificado por el tercer evento. Un número de ronda para el tercer evento puede ser calculado con base en una determinación de que una suma de valores de participación asociados con cada evento del primer conjunto de eventos cumple con un segundo criterio de valor de participación. Un número de ronda para el primer evento corresponde a un segundo número de ronda mayor que el primer número de ronda. El método incluye además determinar un tercer conjunto de eventos con base en el tercer evento. Cada evento del tercer conjunto de eventos es a) identificado por un cuarto conjunto de eventos que incluye el tercer evento y b) del primer conjunto de eventos. Cada evento del cuarto conjunto de eventos está definido por una instancia diferente de la base de datos distribuida y un valor de participación colectiva asociado con el cuarto conjunto de eventos cumple con un tercer criterio de valor de participación. Un valor de orden es definido luego para un cuarto evento con base en un valor de participación colectiva asociado con el tercer conjunto de eventos que cumplen un cuarto criterio de valor de participación y el valor del orden puede ser almacenado en una instancia de la base de datos distribuida en un segundo dispositivo de cálculo.

En algunas realizaciones, un conjunto de valores de participación incluye un valor de participación (1) asociado con cada instancia de la base de datos distribuida que define un evento del segundo conjunto de eventos y (2) proporcional a una cantidad de criptomoneda asociada con esa instancia de la base de datos distribuida. El valor de participación colectiva asociado con el segundo conjunto de eventos que es basado en una suma de valores de participación del conjunto de valores de participación.

En algunas realizaciones, al menos uno del primer criterio de valor de participación, el segundo criterio de valor de participación, el tercer criterio de valor de participación o el cuarto criterio de valor de participación se define con base en un valor de participación colectiva de la base de datos distribuida. Además, en algunas realizaciones, el conjunto de dispositivos de cálculo que implementan la base de datos distribuida por primera vez está asociado con un conjunto de entidades de confianza y el conjunto de dispositivos de cálculo que implementan la base de datos distribuida por segunda vez después de la primera vez está asociado con un conjunto de entidades que incluye entidades no del conjunto de entidades de confianza.

Como se usa en este documento, un módulo puede ser, por ejemplo, cualquier ensamblaje y/o conjunto de componentes eléctricos acoplados operativamente asociados con la realización de una función específica, y puede incluir, por ejemplo, una memoria, un procesador, trazas eléctricas, conectores ópticos, software (que se ejecuta en hardware) y/o similares.

Como se usa en esta especificación, las formas singulares "un", "uno, una" y "el, la" incluyen referentes plurales a menos que el contexto dicte claramente otra cosa. De este modo, por ejemplo, el término "módulo" está previsto para significar un único módulo o una combinación de módulos. Por ejemplo, una "red" está prevista para significar una única red o una combinación de redes.

La figura 1 es un diagrama de bloques de alto nivel que ilustra un sistema 100 de base de datos distribuida, de acuerdo con una realización. La figura 1 ilustra una base de datos 100 distribuida implementada a través de cuatro dispositivos de cálculo (dispositivo 110 de cálculo, dispositivo 120 de cálculo, dispositivo 130 de cálculo, y dispositivo 140 de cálculo), pero debe entenderse que la base de datos 100 distribuida puede usar un conjunto de cualquier número de dispositivos de cálculo, incluyendo dispositivos de cálculo que no se muestran en la figura 1. La red 105 puede ser cualquier tipo de red (por ejemplo, una red de área local (LAN), una red de área amplia (WAN), una red virtual, una red de telecomunicaciones) implementada como una red por cable y/o red inalámbrica y usada para acoplar operativamente dispositivos 110, 120, 130, 140 de cálculo. Como se describe con más detalle en este documento, en algunas realizaciones, por ejemplo, los dispositivos de cálculo son ordenadores personales conectados entre sí a través de un Proveedor de Servicios de Internet (ISP) y el Internet (por ejemplo, red 105). En algunas realizaciones,

puede ser definida una conexión, a través de red 105, entre dos dispositivos 110, 120, 130, 140 de cálculo cualesquiera. Como se muestra en la figura 1, por ejemplo, puede ser definida una conexión entre el dispositivo 110 de cálculo y uno cualquiera del dispositivo 120 de cálculo, dispositivo 130 de cálculo, o dispositivo 140 de cálculo.

En algunas realizaciones, los dispositivos 110, 120, 130, 140 de cálculo pueden comunicarse entre sí (por ejemplo, enviar datos a y/o recibir datos de) y con la red a través de redes intermedias y/o redes alternativas (no se muestran en la figura 1). Tales redes intermedias y/o redes alternativas pueden ser de un mismo tipo y/o un tipo diferente de red que la red 105.

Cada dispositivo 110, 120, 130, 140 de cálculo puede ser cualquier tipo de dispositivo configurado para enviar datos sobre la red 105 para enviar y/o recibir datos de uno o más de los otros dispositivos de cálculo. Se muestran ejemplos de dispositivos de cálculo en la figura 1. El dispositivo 110 de cálculo incluye una memoria 112, un procesador 111, y un dispositivo 113 de salida. La memoria 112 puede ser, por ejemplo, una memoria de acceso aleatorio (RAM), un búfer de memoria, un disco duro, una base de datos, una memoria de solo lectura programable y borrable (EPROM), una memoria de solo lectura borrable eléctricamente (EEPROM), una memoria de solo lectura (ROM) y/o así sucesivamente. En algunas realizaciones, la memoria 112 del dispositivo 110 de cálculo incluye datos asociados con una instancia de una base de datos distribuida (por ejemplo, instancia 114 de base de datos distribuida). En algunas realizaciones, la memoria 112 almacena instrucciones para hacer que el procesador ejecute módulos, procesos y/o funciones asociadas con el envío a y/o recepción de otra instancia de una base de datos distribuida (por ejemplo, instancia 124 de base de datos distribuida en dispositivo 120 de cálculo) un registro de un evento de sincronización, un registro de eventos de sincronización anteriores con otros dispositivos de cálculo, un orden de eventos de sincronización, un valor para un parámetro (por ejemplo, un campo de base de datos que cuantifica una transacción, un campo de base de datos que cuantifica un orden en el cual se producen los eventos, y/o cualquier otro campo adecuado para el cual pueda ser almacenado un valor en una base de datos).

La instancia 114 de base de datos distribuida puede, por ejemplo, estar configurada para manipular datos, incluyendo almacenar, modificar y/o eliminar datos. En algunas realizaciones, la instancia 114 de base de datos distribuida puede ser una base de datos relacional, base de datos de objetos, base de datos postrelacional, y/o cualquier otro tipo adecuado de base de datos. Por ejemplo, la instancia 114 de base de datos distribuida puede almacenar datos relacionados con cualquier función y/o industria específica. Por ejemplo, la instancia 114 de base de datos distribuida puede almacenar transacciones financieras (del usuario del dispositivo 110 de cálculo, por ejemplo), incluyendo un valor y/o un vector de valores relacionados con el historial de propiedad de un instrumento financiero particular. En general, un vector puede ser cualquier conjunto de valores para un parámetro, y un parámetro puede ser cualquier objeto de datos y/o campo de base de datos capaz de tomar diferentes valores. De este modo, una instancia 114 de base de datos distribuida puede tener un número de parámetros y/o campos, cada uno de los cuales está asociado con un vector de valores. El vector de valores es usado para determinar el valor real para el parámetro y/o campo dentro de esa instancia 114 de base de datos.

En algunas instancias, la instancia 114 de base de datos distribuida también puede ser usada para implementar otras estructuras de datos, tales como un conjunto de pares (clave, valor). Una transacción registrada por la instancia 114 de base de datos distribuida puede ser, por ejemplo, agregar, eliminar, o modificar un par (clave, valor) en un conjunto de pares (clave, valor).

En algunas instancias, puede ser consultado el sistema 100 de base de datos distribuida o cualquiera de las instancias 114, 124, 134, 144 de base de datos distribuida. Por ejemplo, una consulta puede consistir en una clave, y el resultado regresado del sistema 100 de base de datos distribuida o instancias 114, 124, 134, 144 de base de datos distribuida puede ser un valor asociado con la clave. En algunas instancias, el sistema 100 de base de datos distribuida o cualquiera de las instancias 114, 124, 134, 144 de base de datos distribuida también puede ser modificado a través de una transacción. Por ejemplo, una transacción para modificar la base de datos puede contener una firma digital por una parte que autoriza la transacción de modificación.

El sistema 100 de base de datos distribuida puede ser usado para muchos propósitos, tales como, por ejemplo, almacenar atributos asociados con diversos usuarios en un sistema de identidad distribuida. Por ejemplo, tal sistema puede usar una identidad del usuario como la "clave", y la lista de atributos asociados con los usuarios como el "valor". En algunas instancias, la identidad puede ser una clave pública criptográfica con una clave privada correspondiente conocida por ese usuario. Cada atributo puede, por ejemplo, estar firmado digitalmente por una autoridad que tenga el derecho a hacer valer ese atributo. Cada atributo también puede, por ejemplo, estar encriptado con la clave pública asociada con un individuo o grupo de individuos que tienen derecho a leer el atributo. Algunas claves o valores también pueden tener adjunta a ellos una lista de claves públicas de las partes que están autorizadas a modificar o eliminar las claves o valores.

En otro ejemplo, la instancia 114 de base de datos distribuida puede almacenar datos relacionados con los Juegos Multijugador Masivo (MMGs), tal como el estado actual y propiedad de ítems de juego. En algunas instancias, la instancia 114 de base de datos distribuida puede ser implementada dentro del dispositivo 110 de cálculo, como se muestra en la figura 1. En otras instancias, la instancia de la base de datos distribuida es accesible por el dispositivo de cálculo (por ejemplo, a través de una red), pero no es implementada en el dispositivo de cálculo (no se muestra en la figura 1).

El procesador 111 del dispositivo 110 de cálculo puede ser cualquier dispositivo de procesamiento adecuado configurado para realizar y/o ejecutar la instancia 114 de base de datos distribuida. Por ejemplo, el procesador 111 puede estar configurado para actualizar la instancia 114 de base de datos distribuida en respuesta a recibir una señal del dispositivo 120 de cálculo, y/o hacer que una señal sea enviada al dispositivo 120 de cálculo, como se describe con más detalle en este documento. Más específicamente, como se describe con más detalle en este documento, el procesador 111 puede estar configurado para ejecutar módulos, funciones y/o procesos para actualizar la instancia 114 de base de datos distribuida en respuesta a recibir un evento de sincronización asociado con una transacción desde otro dispositivo de cálculo, un registro asociado con un orden de eventos de sincronización, y/o similares. En otras realizaciones, el procesador 111 puede estar configurado para ejecutar módulos, funciones y/o procesos para actualizar la instancia 114 de base de datos distribuida en respuesta a recibir un valor para un parámetro almacenado en otra instancia de la base de datos distribuida (por ejemplo, instancia 124 de base de datos distribuida en el dispositivo 120 de cálculo), y/o hacer que un valor para un parámetro almacenado en la instancia 114 de base de datos distribuida en el dispositivo 110 de cálculo sea enviado al dispositivo 120 de cálculo. En algunas realizaciones, el procesador 111 puede ser un procesador de propósito general, un Arreglo de Puerta Programable en Campo (FPGA), un Circuito Integrado de Aplicación Específica (ASIC), un Procesador de Señal Digital (DSP), y/o similares.

La pantalla 113 puede ser cualquier pantalla adecuada, tal como, por ejemplo, una pantalla de cristal líquido (LCD), una pantalla de tubo de rayos catódicos (CRT) o similar. En otras realizaciones, cualquiera de los dispositivos 110, 120, 130, 140 de cálculo incluye otro dispositivo de salida en vez o además de las pantallas 113, 123, 133, 143. Por ejemplo, uno cualquiera de los dispositivos 110, 120, 130, 140 de cálculo puede incluir un dispositivo de salida de audio (por ejemplo, un altavoz), un dispositivo de salida táctil, y/o similares. En aún otras realizaciones, cualquiera de los dispositivos 110, 120, 130, 140 de cálculo incluye un dispositivo de entrada en vez o además de las pantallas 113, 123, 133, 143. Por ejemplo, uno cualquiera de los dispositivos 110, 120, 130, 140 de cálculo puede incluir un teclado, un ratón, y/o similares.

El dispositivo 120 de cálculo tiene un procesador 121, una memoria 122, y una pantalla 123, que pueden ser estructural y/o funcionalmente similares al procesador 111, la memoria 112, y la pantalla 113, respectivamente. También, la instancia 124 de base de datos distribuida puede ser estructural y/o funcionalmente similar a la instancia 114 de base de datos distribuida.

El dispositivo 130 de cálculo tiene un procesador 131, una memoria 132, y una pantalla 133, que pueden ser estructural y/o funcionalmente similares al procesador 111, la memoria 112, y la pantalla 113, respectivamente. También, la instancia 134 de base de datos distribuida puede ser estructural y/o funcionalmente similar a la instancia 114 de base de datos distribuida.

El dispositivo 140 de cálculo tiene un procesador 141, una memoria 142, y una pantalla 143, que pueden ser estructural y/o funcionalmente similares al procesador 111, la memoria 112, y la pantalla 113, respectivamente. También, la instancia 144 de base de datos distribuida puede ser estructural y/o funcionalmente similar a la instancia 114 de base de datos distribuida.

A pesar de que los dispositivos 110, 120, 130, 140 de cálculo se muestran como similares entre sí, cada dispositivo de cálculo del sistema 100 de base de datos distribuida puede ser diferente de los otros dispositivos de cálculo. Cada dispositivo 110, 120, 130, 140 de cálculo del sistema 100 de base de datos distribuida puede ser uno cualquiera de, por ejemplo, una entidad informática (por ejemplo, un dispositivo informático personal tal como un ordenador de escritorio, un ordenador portátil, etc.), un teléfono móvil, un asistente digital personal (PDA), y así sucesivamente. Por ejemplo, el dispositivo 110 de cálculo puede ser un ordenador de escritorio, el dispositivo 120 de cálculo puede ser un teléfono inteligente, y el dispositivo 130 de cálculo puede ser un servidor.

En algunas realizaciones, una o más porciones de los dispositivos 110, 120, 130, 140 de cálculo pueden incluir un módulo basado en hardware (por ejemplo, un procesador de señales digitales (DSP), un arreglo de puerta programable en campo (FPGA)) y/o un módulo basado en software (por ejemplo, un módulo de código de ordenador almacenado en memoria y/o ejecutado en un procesador). En algunas realizaciones, una o más de las funciones asociadas con los dispositivos 110, 120, 130, 140 de cálculo (por ejemplo, las funciones asociadas con los procesadores 111, 121, 131, 141) pueden estar incluidas en uno o más módulos (véase, por ejemplo, figura 2).

Las propiedades del sistema 100 de base de datos distribuida, incluyendo las propiedades de los dispositivos de cálculo (por ejemplo, los dispositivos 110, 120, 130, 140 de cálculo), el número de dispositivos de cálculo, y la red 105, pueden ser seleccionadas en cualquier número de formas. En algunas instancias, las propiedades del sistema 100 de base de datos distribuida pueden ser seleccionadas por un administrador del sistema 100 de base de datos distribuida. En otras instancias, las propiedades del sistema 100 de base de datos distribuida pueden ser seleccionadas colectivamente por los usuarios del sistema 100 de base de datos distribuida.

Debido a que es usado un sistema 100 de base de datos distribuida, no se designa ningún líder entre los dispositivos 110, 120, 130, y 140 de cálculo. Específicamente, ninguno de los dispositivos 110, 120, 130, o 140 de cálculo es identificado y/o seleccionado como un líder para resolver disputas entre valores almacenados en las instancias 111, 12, 131, 141 de base de datos distribuida de los dispositivos 110, 120, 130, 140 de cálculo. En vez de esto, usando

los procesos de sincronización de eventos, los procesos y/o métodos de votación descritos en este documento, los dispositivos 110, 120, 130, 140 de cálculo pueden converger colectivamente en un valor para un parámetro.

No tener un líder en un sistema de base de datos distribuida aumenta la seguridad del sistema de base de datos distribuida. Específicamente, con un líder hay un único punto de ataque y/o falla. Si el software malicioso infecta al líder y/o un valor para un parámetro en la instancia de base de datos distribuida del líder es alterado maliciosamente, la falla y/o valor incorrecto es propagado a través de las otras instancias de bases de datos distribuida. En un sistema sin líder, sin embargo, no hay un único punto de ataque y/o falla. Específicamente, si un parámetro en una instancia de base de datos distribuida de un sistema sin líder contiene un valor, el valor cambiará después de que esa instancia de base de datos distribuida intercambie valores con las otras instancias de bases de datos distribuida en el sistema, como se describe con más detalle en este documento. Adicionalmente, los sistemas de bases de datos distribuidas sin líder descritos en este documento aumentan la velocidad de convergencia mientras que reducen la cantidad de datos enviados entre dispositivos como se describe con más detalle en este documento.

La figura 2 ilustra un dispositivo 200 de cálculo de un sistema de base de datos distribuida (por ejemplo, sistema 100 de base de datos distribuida), de acuerdo con una realización. En algunas realizaciones, el dispositivo 200 de cálculo puede ser similar a los dispositivos 110, 120, 130, 140 de cálculo mostrados y descritos con respecto a la figura 1. El dispositivo 200 de cálculo incluye un procesador 210 y una memoria 220. El procesador 210 y memoria 220 están acoplados operativamente entre sí. En algunas realizaciones, el procesador 210 y memoria 220 pueden ser similares al procesador 111 y memoria 112, respectivamente, descritos en detalle con respecto a la figura 1. Como se muestra en la figura 2, el procesador 210 incluye un módulo 211 de convergencia de base de datos y módulo 210 de comunicación, y la memoria 220 incluye una instancia 221 de base de datos distribuida. El módulo 212 de comunicación habilita que el dispositivo 200 de cálculo se comunique con (por ejemplo, envíe datos a y/o reciba datos de) otros dispositivos de cálculo. En algunas realizaciones, el módulo 212 de comunicación (no se muestra en la figura 1) habilita que el dispositivo 110 de cálculo se comunique con dispositivos 120, 130, 140 de cálculo. El módulo 210 de comunicación puede incluir y/o habilitar, por ejemplo, un controlador de interfaz de red (NIC), conexión inalámbrica, un puerto con cable, y/o similares. Como tal, el módulo 210 de comunicación puede establecer y/o mantener una sesión de comunicación entre el dispositivo 200 de cálculo y otro dispositivo (por ejemplo, a través de una red tal como red 105 de la figura 1 o el Internet (no se muestra)). Indicado de manera similar, el módulo 210 de comunicación puede habilitar que el dispositivo 200 de cálculo envíe datos a y/o reciba datos de otro dispositivo.

En algunas instancias, el módulo 211 de convergencia de base de datos puede intercambiar eventos y/o transacciones con otros dispositivos informáticos, almacenar eventos y/o transacciones que recibe el módulo 211 de convergencia de base de datos, y calcular un orden de los eventos y/o transacciones con base en el orden parcial definido por el patrón de referencias entre los eventos. Cada evento puede ser un registro que contiene un direccionamiento criptográfico de dos eventos anteriores (vinculando ese evento con los dos eventos anteriores y sus eventos antepasados, y viceversa), datos de carga útil (tales como transacciones que van a ser registradas), otra información tal como la hora actual, una marca de tiempo (por ejemplo, fecha y hora UTC) que su creador afirma es la hora en que fue definido por primera vez el evento, y/o similar. En algunas instancias, el primer evento definido por un miembro solo incluye un direccionamiento de un único evento definido por otro miembro. En tales instancias, el miembro aún no tiene un direccionamiento propio anterior (por ejemplo, un direccionamiento de un evento definido previamente por ese miembro). En algunas instancias, el primer evento en una base de datos distribuida no incluye un direccionamiento de ningún evento anterior (dado que no hay un evento anterior para esa base de datos distribuida).

En algunas realizaciones, tal direccionamiento criptográfico de los dos eventos anteriores puede ser un valor de direccionamiento definido con base en una función de direccionamiento criptográfico usando un evento como una entrada. Específicamente, en tales realizaciones, el evento incluye una secuencia particular o cadena de bytes (que representan la información de ese evento). El direccionamiento de un evento puede ser un valor regresado de una función de direccionamiento usando la secuencia de bytes para ese evento como una entrada. En otras realizaciones, cualquier otro dato adecuado asociado con el evento (por ejemplo, un identificador, número de serie, los bytes que representan una porción específica del evento, etc.) puede ser usado como una entrada a la función de direccionamiento para calcular el direccionamiento de ese evento. Puede ser usada cualquier función de direccionamiento adecuada para definir el direccionamiento. En algunas realizaciones, cada miembro usa la misma función direccionamiento de tal manera que es generado el mismo direccionamiento en cada miembro para un evento dado. El evento puede luego ser firmado digitalmente por el miembro que define y/o que crea el evento.

En algunas instancias, el conjunto de eventos y sus interconexiones pueden formar un Gráfico Acíclico Dirigido (DAG). En algunas instancias, cada evento en un DAG hace referencia a dos eventos anteriores (vinculando ese evento a los dos eventos anteriores y sus eventos antepasados y viceversa), y cada referencia es estrictamente a los anteriores, de tal manera que no hay bucles. En algunas realizaciones, el DAG se basa en direccionamientos criptográficos, por lo que la estructura de datos puede denominarse un *hashDAG*. El *hashDAG* codifica directamente un orden parcial, lo que significa que se conoce que el evento X viene antes del evento Y si Y contiene un direccionamiento de X, o si Y contiene un direccionamiento de un evento que contiene un direccionamiento de X, o para tales trayectorias de longitud arbitraria. Sin embargo, si no hay una trayectoria de X a Y o de Y a X, entonces el orden parcial no define cual evento se produjo primero. Por lo tanto, el módulo de convergencia de base de datos puede calcular un orden total a partir del orden parcial. Esto puede hacerse mediante cualquier función determinista adecuada que es usada por los dispositivos de cálculo, de tal manera que los dispositivos de cálculo calculan el mismo orden. En algunas

realizaciones, cada miembro puede recalcular este orden después de cada sincronización, y finalmente estos órdenes pueden converger de tal manera que surja un consenso.

Puede ser usado un algoritmo de consenso para determinar el orden de eventos en un *hashDAG* y/o el orden de transacciones almacenadas dentro de los eventos. El orden de las transacciones a su vez puede definir un estado de una base de datos como resultado de realizar esas transacciones de acuerdo con el orden. El estado definido de la base de datos puede ser almacenado como una variable de estado de base de datos.

En algunas instancias, el módulo de convergencia de base de datos puede usar la siguiente función para calcular un orden total a partir del orden parcial en el *hashDAG*. Para cada uno de los otros dispositivos de cálculo (denominados "miembros"), el módulo de convergencia de base de datos puede examinar el *hashDAG* para descubrir un orden en el cual los eventos (y/o indicaciones de esos eventos) fueron recibidos por ese miembro. El módulo de convergencia de base de datos puede entonces calcular como si ese miembro hubiera asignado un "rango" numérico a cada evento, siendo el rango 1 para el primer evento que recibió el miembro, 2 para el segundo evento que recibió el miembro, y así sucesivamente. El módulo de convergencia de base de datos puede hacer esto para cada miembro en el *hashDAG*. Luego, para cada evento, el módulo de convergencia de base de datos puede calcular la mediana de los rangos asignados, y puede ordenar los eventos por sus medianas. El ordenamiento puede romper los empates de una manera determinista, tal como ordenar dos eventos empatados por un orden numérico de sus direccionamientos, o por algún otro método, en el cual el módulo de convergencia de base de datos de cada miembro usa el mismo método. El resultado de este tipo es el orden total.

La figura 6 ilustra un *hashDAG* 640 de un ejemplo para determinar un orden total. HashDAG 640 ilustra dos eventos (el círculo de rayas más bajo y círculo de puntos más bajo) y la primera vez que cada miembro recibe una indicación de esos eventos (los otros círculos de rayas y de puntos). Cada nombre del miembro en la parte superior está coloreado por cual evento es el primero en su orden lento. Hay más votos iniciales de rayas que de puntos, por lo tanto los votos por consenso para cada uno de los miembros son de rayas. En otras palabras, los miembros finalmente convergen en un acuerdo de que el evento de rayas se produjo antes que el evento de puntos.

En este ejemplo, los miembros (dispositivos de cálculo etiquetados Alice, Bob, Carol, Dave y Ed) trabajarán para definir un consenso de si el evento 642 o evento 644 se produjo primero. Cada círculo de rayas indica el evento en el cual un miembro recibió primero un evento 644 (y/o una indicación de ese evento 644). De manera similar, cada círculo de puntos indica el evento en el cual un miembro recibió primero un evento 642 (y/o una indicación de ese evento 642). Como se muestra en el *hashDAG* 640, Alice, Bob y Carol recibieron cada uno el evento 644 (y/o una indicación de evento 644) antes del evento 642. Dave y Ed ambos recibieron el evento 642 (y/o una indicación de evento 642) antes del evento 644 (y/o una indicación de evento 644). De este modo, debido a que un mayor número de miembros recibió el evento 644 antes del evento 642, el orden total puede ser determinado por cada miembro para indicar que ese evento 644 se produjo antes del evento 642.

En otras instancias, el módulo de convergencia de base de datos puede usar una función diferente para calcular el orden total a partir del orden parcial en el *hashDAG*. En tales realizaciones, por ejemplo, el módulo de convergencia de base de datos puede usar las siguientes funciones para calcular el orden total, donde un entero positivo Q es un parámetro compartido por los miembros.

creador(x)) = el miembro que creó el evento x

anc(x) = el conjunto de eventos que son antepasados de x, incluyendo el propio x

otro(x) = el evento creado por el miembro que se sincronizó justo antes de que fuera creado x

propio(x) = el último evento antes de x con el mismo creador

propio(x, 0) = propio(x)

propio(x, n) = propio(propio(x), n - 1)

orden(x, y) = k, donde y es el k-ésimo evento del que se enteró el creador(x)

último(x) = {y|y ∈ anc(x) Λ ¬∃z ∈ anc(x), (y ∈ anc(z) Λ creador(y) = creador(z))}

$$\text{lento}(x,y) = \begin{cases} \infty & \text{si } y \notin anc(x) \\ \text{orden}(x,y) & \text{si } y \in anc(x) \wedge y \notin anc(\text{propio}(x)) \\ \text{rápido}(x,y) & \text{si } \forall i \in \{1,\dots,Q\},\ \text{rápido}(x,y) = \text{rápido}(\text{propio}(x,i),y) \\ \text{lento}(\text{propio}(x),y) & \text{de otra manera} \end{cases}$$

rápido(x,y) = la posición de y en una lista ordenada, con elemento z ∈ anc(x) ordenados por la mediana lento(w,z) y con empates rotos por el direccionamiento de cada evento w∈último(x)

En esta realización, rápido(x,y) da la posición de y en el orden total de los eventos, en la opinión del creador(x), sustancialmente de manera inmediata después de que se crea y/o define x. Si Q es infinito, entonces lo anterior calcula el mismo orden total que en la realización descrita previamente. Si Q es finito, y todos los miembros están en línea, entonces lo anterior calcula el mismo orden total que en la realización descrita previamente. Si Q es finito y una minoría de los miembros está en línea en un momento dado, entonces esta función permite a los miembros en línea llegar a un consenso entre ellos mismos que permanecerá sin cambios a medida que los nuevos miembros se conecten en línea lentamente, uno por uno. Sin embargo, si hay una partición de la red, entonces los miembros de cada partición pueden llegar a su propio consenso. Luego, cuando la partición esté curada, los miembros de la partición más pequeña adoptarán el consenso de la partición más grande.

En aún otras instancias, como se describe con respecto a las figuras 14-17b, el módulo de convergencia de base de datos puede usar todavía una función diferente para calcular el orden total a partir del orden parcial en el *hashDAG*. Como se muestra en las figuras 14-15, cada miembro (Alice, Bob, Carol, Dave y Ed) crea y/o define eventos (1401-1413 como se muestra en la figura 14; 1501-1506 se muestra en la figura 15). Usando la función y subfunciones descritas con respecto a las figuras 14-17b, el orden total para los eventos puede ser calculado ordenando los eventos por su ronda recibida (también denominada en este documento como un valor de orden), rompiendo los empates por su marca de tiempo recibida, y rompiendo esos empates por sus firmas, como se describe en más detalles en este documento. En otras instancias, el orden total para los eventos puede ser calculado ordenando los eventos por su ronda recibida, rompiendo los empates por su generación recibida (en vez de su marca de tiempo recibida), y rompiendo esos empates por sus firmas. Los siguientes párrafos especifican funciones que se usan para calcular y/o definir una ronda recibida del evento y la generación recibida para determinar un orden para los eventos. Los siguientes términos son usados e ilustrados en relación con las figuras 14-17b.

"Padre": un evento X es un padre de evento Y si Y contiene un direccionamiento de X. Por ejemplo, en la figura 14, los padres de evento 1412 incluyen el evento 1406 y evento 1408.

"Antepasado": los antepasados de un evento X son X, sus padres, los padres de sus padres, y así sucesivamente. Por ejemplo, en la figura 14, los antepasados del evento 1412 son eventos 1401, 1402, 1403, 1406, 1408, y 1412. Se puede decir que los antepasados de un evento están vinculados a ese evento y viceversa.

"Descendiente": los descendientes de un evento X son X, sus hijos, los hijos de sus hijos, y así sucesivamente. Por ejemplo, en la figura 14, los descendientes del evento 1401 son todos los eventos que se muestran en la figura. Para otro ejemplo, los descendientes del evento 1403 son eventos 1403, 1404, 1406, 1407, 1409, 1410, 1411, 1412, y 1413. Se puede decir que los descendientes de un evento están vinculados a ese evento y viceversa.

"N": el número total de miembros de la población. Por ejemplo, en la figura 14, los miembros son dispositivos de cálculo etiquetados Alice, Bob, Carol, Dave y Ed, y N es igual a cinco.

"M": el menor entero que es más de un cierto porcentaje de N (por ejemplo, más de 2/3 de N). Por ejemplo, en la figura 14, si el porcentaje está definido como 2/3, entonces M es igual a cuatro. En otras instancias, M podría ser definido, por ejemplo, como un porcentaje diferente de N (por ejemplo, 1/3, 1/2, etc.), un número predefinido específico, y/o de cualquier otra manera adecuada.

"Padre propio": el padre propio de un evento X es su evento padre Y creado y/o definido por el mismo miembro. Por ejemplo, en la figura 14, el padre propio del evento 1405 es 1401.

"Antepasado propio": los antepasados propios de un evento X son X, su padre propio, el padre propio de su padre propio, y así sucesivamente.

"Número de Secuencia" (o "SN"): un atributo entero de un evento, definido como el Número de Secuencia del padre propio del evento, más uno. Por ejemplo, en la figura 14, el padre propio del evento 1405 es 1401. Dado que el Número de Secuencia del evento 1401 es uno, el Número de Secuencia del evento 1405 es dos (es decir, uno más uno).

"Número de Generación" (o "GN"): un atributo entero de un evento, definido como el máximo de los Números de Generación de los padres del evento, más uno. Por ejemplo, en la figura 14, el evento 1412 tiene dos padres, los eventos 1406 y 1408, que tienen Números de Generación cuatro y dos, respectivamente. De este modo, el Número de Generación del evento 1412 es cinco (es decir, cuatro más uno).

"Incremento de Ronda" (o "RI"): un atributo de un evento que puede ser ya sea cero o uno.

"Número de Ronda" (o "RN"): un atributo entero de un evento. En algunas instancias, el Número de Ronda puede ser definido como el máximo de los Números de Rondas de los padres del evento, más el Incremento de Ronda del evento. Por ejemplo, en la figura 14, el evento 1412 tiene dos padres, los eventos 1406 y 1408, ambos teniendo un Número de Ronda de uno. El evento 1412 también tiene un Incremento de Ronda de uno. De este modo, el Número de Ronda del evento 1412 es dos (es decir, uno más uno). En otras instancias, un evento puede tener un Número de Ronda R si R es el entero mínimo de tal manera que el evento puede ver fuertemente (como se describe en este documento) al menos M eventos definidos y/o creados por diferentes miembros, todos los cuales tienen un número de ronda R-1. Si no hay tal entero, el Número de Ronda para un evento puede ser un valor predeterminado (por ejemplo, 0, 1, etc.).

En tales instancias, el Número de Ronda para un evento puede ser calculado sin usar un Incremento de Ronda. Por ejemplo, en la figura 14, si M está definido como el menor entero mayor que 1/2 veces N, entonces M es tres. Luego el evento 1412 ve fuertemente los M eventos 1401, 1402, y 1408, cada uno de los cuales fue definido por un miembro diferente y tiene un Número de Ronda de 1. El evento 1412 no puede ver fuertemente al menos M eventos con Número de Ronda de 2 que fueron definidos por diferentes miembros. Por lo tanto, el Número de Ronda para el evento 1412 es 2. En algunas instancias, el primer evento en la base de datos distribuida incluye un Número de Ronda de 1. En otras instancias, el primer evento en la base de datos distribuida puede incluir un Número de Ronda de 0 o cualquier otro número adecuado.

"Bifurcación": un evento X es una bifurcación con el evento Y si están definidos y/o creados por el mismo miembro, y ninguno es un antepasado propio del otro. Por ejemplo, en la figura 15, el miembro Dave bifurca creando y/o definiendo eventos 1503 y 1504, ambos teniendo el mismo padre propio (es decir, evento 1501), de tal manera que el evento 1503 no es un antepasado propio del evento 1504, y el evento 1504 no es un antepasado propio del evento 1503.

"Identificación" de bifurcación: la bifurcación puede ser "identificada" por un tercer evento creado y/o definido después de los dos eventos que son bifurcaciones entre sí, si esos dos eventos son ambos antepasados del tercer evento. Por ejemplo, en la figura 15, el miembro Dave se bifurca creando eventos 1503 y 1504, ninguno de los cuales es un antepasado propio del otro. Esta bifurcación puede ser identificada por el evento 1506 posterior debido a que los eventos 1503 y 1504 son ambos antepasados del evento 1506. En algunas instancias, la identificación de bifurcación puede indicar que un miembro en particular (por ejemplo, Dave) ha hecho trampa.

"Identificación" de un evento: un evento X "identifica" o "ve" un evento antepasado Y si X no tiene un evento antepasado Z que sea una bifurcación con Y. Por ejemplo, en la figura 14, el evento 1412 identifica (también denominado como "ve") el evento 1403 debido a que el evento 1403 es un antepasado del evento 1412, y el evento 1412 no tiene eventos antepasados que sean bifurcaciones con el evento 1403. En algunas instancias, el evento X puede identificar el evento Y si X no identifica la bifurcación antes del evento Y. En tales instancias, incluso si el evento X identifica la bifurcación por el miembro que define el evento Y subsecuente al evento Y, el evento X puede ver el evento Y. El evento X no identifica eventos mediante ese miembro subsecuente a la bifurcación. Además, si un miembro define dos eventos diferentes que son ambos esos primeros eventos del miembro en la historia, el evento X puede identificar la bifurcación y no identificar ningún evento mediante ese miembro.

"Identificación fuerte" (también denominada en este documento como "ver fuertemente") de un evento: un evento X "identifica fuertemente" (o "ve fuertemente") un evento antepasado Y creado y/o definido por el mismo miembro que X, si X identifica Y. El evento X "identifica fuertemente" un evento antepasado Y que no es creado y/o definido por el mismo miembro que X, si existe un conjunto S de eventos que (1) incluye tanto X como Y y (2) son antepasados del evento X y (3) son descendientes del evento antepasado Y y (4) son identificados por X y (5) pueden identificar cada uno Y y (6) son creados y/o definidos por al menos M miembros diferentes. Por ejemplo, en la figura14, si M es definido como el menor entero que es más de 2/3 de N (es decir, M=1+suelo(2N/3), que sería cuatro en este ejemplo), entonces el evento 1412 identifica fuertemente al evento 1401 antepasado debido a que el conjunto de eventos 1401, 1402, 1406, y 1412 es un conjunto de al menos cuatro eventos que son antepasados del evento 1412 y descendientes del evento 1401, y son creados y/o definidos por los cuatro miembros Dave, Carol, Bob, y Ed, respectivamente, y el evento 1412 identifica cada uno de los eventos 1401, 1402, 1406, y 1412, y cada uno de los eventos 1401, 1402, 1406, y 1412 identifica el evento 1401. Indicado de manera similar, un evento X (por ejemplo, evento 1412) puede "ver fuertemente" el evento Y (por ejemplo, evento 1401) si X puede ver al menos M eventos (por ejemplo, eventos 1401, 1402, 1406, y 1412) creados o definidos por diferentes miembros, cada uno de los cuales puede ver Y.

Primer “evento de ronda R" (también denominado en este documento como "testigo"): un evento es un primer “evento de ronda R" (o un "testigo") si el evento (1) tiene Número de Ronda R, y (2) tiene un padre propio que tiene un Número de Ronda menor que R o no tiene padre propio. Por ejemplo, en la figura 14, el evento 1412 es un primer “evento de ronda 2" debido a que tiene un Número de Ronda de dos, y su padre propio es el evento 1408, que tiene un Número de Ronda de uno (es decir, menor que dos).

En algunas instancias, el incremento de Ronda para un evento X es definido como 1 si y solo si X "identifica fuertemente" al menos M primeros "eventos de rondas R", donde R es el Número de Ronda máximo de sus padres. Por ejemplo, en la figura 14, si M es definido como el menor entero mayor que 1/2 veces N, entonces M es tres. Entonces el evento 1412 identifica fuertemente los M eventos 1401, 1402, y 1408, todos los cuales son primeros eventos de rondas 1. Ambos padres de 1412 son la ronda 1, y 1412 identifica fuertemente al menos M primeras rondas 1, por lo tanto el incremento de ronda para 1412 es uno. Los eventos en el diagrama marcados con "RI=0" fallan cada uno en identificar fuertemente al menos M primeras rondas 1, por lo tanto sus incrementos de rondas son 0.

En algunas instancias, el siguiente método puede ser usado para determinar si el evento X puede identificar fuertemente el evento antepasado Y. Para cada primer evento Y antepasado de ronda R, mantener un arreglo A1 de enteros, uno por miembro, dando el número de secuencia más bajo del evento X, donde ese miembro creó y/o definió el evento X, y X puede identificar Y. Para cada evento Z, mantener un arreglo A2 de enteros, uno por miembro, dando el número de secuencia más alto de un evento W creado y/o definido por ese miembro, de tal manera que Z pueda identificar W. Para determinar si Z puede identificar fuertemente el evento antepasado Y, contar el número de posiciones de elementos E de tal manera que A1[E] <= A2[E]. El evento Z puede identificar fuertemente Y si y solo si

este conteo es mayor que M. Por ejemplo, en la figura 14, los miembros Alice, Bob, Carol, Dave y Ed pueden identificar cada uno el evento 1401, donde el evento más temprano que puede hacerlo son sus eventos {1404, 1403, 1402, 1401, 1408}, respectivamente. Estos eventos tienen números de secuencia A1={1,1,1,1,1}. De manera similar, el último evento para cada uno de ellos que es identificado por el evento 1412 es evento {NINGUNO, 1406, 1402, 1401, 1412}, donde Alice aparece como "NINGUNO" debido a que 1412 no puede identificar ningún evento de Alice. Estos eventos tienen números de secuencia de A2={0,2,1,1,2}, respectivamente, donde todos los eventos tienen números de secuencia positivos, por lo que el 0 significa que Alice no tiene eventos que son identificados por 1412. Comparando la lista A1 con la lista A2 da los resultados {1<=0, 1<=2, 1<=1, 1<=1, 1<=2} que es equivalente a {falso, verdadero, verdadero, verdadero, verdadero} que tiene cuatro valores que son verdaderos. Por lo tanto, existe un conjunto S de cuatro eventos que son antepasados de 1412 y descendientes de 1401. Cuatro es al menos M, por lo tanto 1412 identifica fuertemente a 1401.

Aún otra variación en la implementación del método para determinar, con A1 y A2, si el evento X puede identificar fuertemente el evento antepasado Y es como sigue. Si los elementos enteros en ambos arreglos son menores que 128, entonces es posible almacenar cada elemento en un único byte, y empaquetar 8 de tales elementos en una única palabra de 64 bits, y dejar que A1 y A2 sean arreglos de tales palabras. El bit más significativo de cada byte en A1 puede ser establecido en 0, y el bit más significativo de cada byte en A2 puede ser establecido en 1. Restar las dos palabras correspondientes, luego realizar un AND bit a bit con una máscara para poner a cero todo menos los bits más significativos, luego desplazar a la derecha en posiciones de 7 bits, para obtener un valor que se expresa en el lenguaje de programación C como: ((A2[i] - A1[i]) & 0x8080808080808080) >> 7). Esto se puede agregar a un acumulador en funcionamiento S que fue inicializado a cero. Después de hacer esto múltiples veces, convertir el acumulador a un conteo desplazando y agregando los bytes, para obtener ((S & 0xff) + ((S >> 8) & 0xff) + ((S >> 16) & 0xff) + ((S >> 24) & 0xff) + ((S >> 32) & 0xff) + ((S >> 40) & 0xff) + ((S >> 48) & 0xff) + ((S >> 56)) & 0xff)). En algunas instancias, estos cálculos pueden ser realizados en lenguajes de programación tales como C, Java, y/o similares. En otras instancias, los cálculos pueden ser realizados usando instrucciones específicas de procesador tales como las instrucciones de Extensiones Vectoriales Avanzadas (AVX) proporcionadas por Intel y AMD, o el equivalente en una unidad de procesamiento de gráficos (GPU) o unidad de procesamiento de gráficos de propósito general (GPGPU). En algunas arquitecturas, los cálculos pueden ser realizados más rápido usando palabras de más de 64 bits, tales como 128, 256, 512, o más bits.

Evento "famoso": un evento X de ronda R es "famoso" si (1) el evento X es un primer “evento de ronda R" (o "testigo") y (2) se llega a decisión de "SÍ" a través de la ejecución de un protocolo de acuerdo Bizantino, que se describe a continuación. En algunas realizaciones, el protocolo de acuerdo Bizantino puede ser ejecutado por una instancia de una base de datos distribuida (por ejemplo, instancia 114 de base de datos distribuida) y/o un módulo de convergencia de base de datos (por ejemplo, módulo 211 de convergencia de base de datos). Por ejemplo, en la figura14, se muestran cinco primeras rondas 1: 1401, 1402, 1403, 1404, y 1408. Si M es definido como el menor entero mayor que 1/2 veces N, que es tres, entonces 1412 es una primera ronda 2. Si el protocolo se ejecuta más tiempo, entonces el *hashDAG* crecerá hacia arriba, y finalmente los otros cuatro miembros también tendrán las primeras rondas 2 por encima de la parte superior de esta figura. Cada primera ronda 2 tendrá un "voto" sobre si cada una de las primeras rondas 1 es "famosa". El evento 1412 votaría SÍ para que 1401, 1402, y 1403 sean famosos, debido a que son las primeras rondas 1 que puede identificar. El evento 1412 votaría NO para que 1404 sea famoso, debido a que 1412 no puede identificar 1404. Para una primera ronda 1 dada, tal como 1402, su estado de ser "famosa" o no se decidirá calculando los votos de cada primera ronda 2 por si es famosa o no. Luego esos votos se propagarán a las primeras rondas 3, luego a primeras rondas 4 y así sucesivamente, hasta que finalmente se llegue a un acuerdo sobre si 1402 fue famoso. El mismo proceso es repetido para otros primeros.

Un protocolo de acuerdo Bizantino puede recolectar y usar los votos y/o decisiones de los primeros “eventos de ronda R" para identificar "eventos famosos. Por ejemplo, una primera Y de "ronda R+1" Y votará "SÍ" si Y puede "identificar" el evento X, de lo contrario vota "NO". Los votos luego son calculados para cada ronda G, para G = R+2, R+3, R+4, etc., hasta que se llega a una decisión por cualquier miembro. Hasta que se haya llegado una decisión, es calculado un voto para cada ronda G. Algunas de esas rondas pueden ser rondas de "mayoría", mientras que otras rondas pueden ser rondas de "monedas". En algunas instancias, por ejemplo, la Ronda R+2 es una ronda de mayoría, y las rondas futuras son designadas ya sea como una ronda de mayoría o una de moneda (por ejemplo, de acuerdo con una programación predefinida). Por ejemplo, en algunas instancias, si una ronda futura es una ronda de mayoría o una ronda de monedas puede ser determinada arbitrariamente, sujeta a la condición de que no pueda haber dos rondas de monedas consecutivas. Por ejemplo, podría estar predefinido que habrá cinco rondas de mayoría, luego una ronda de monedas, luego cinco rondas de mayoría, luego una ronda de monedas, repetidas mientras sea necesario para llegar a un acuerdo.

En algunas instancias, si la ronda G es una ronda de mayoría, los votos pueden ser calculados como sigue. Si existe un evento de ronda G que identifica fuertemente al menos M primeras rondas G-1 votando V (donde V es ya sea "SÍ" o "NO"), entonces la decisión de consenso es V, y el protocolo de acuerdo Bizantino finaliza. De lo contrario, cada primer evento de ronda G calcula un nuevo voto que es la mayoría de las primeras rondas G-1 que cada primer evento de ronda G puede identificar fuertemente. En las instancias donde haya un empate en vez de mayoría, el voto puede ser designado como "SÍ".

Indicado de manera similar, si X es un testigo de ronda R (o primera ronda R), entonces pueden ser calculados los resultados de votos en las rondas R+1, R+2, y así sucesivamente, donde los testigos en cada ronda están votando si X es famoso. En la ronda R+1, todos los testigos que pueden ver X votan SÍ, y los otros testigos votan NO. En la ronda R+2, cada testigo vota de acuerdo con la mayoría de votos de los testigos de ronda R+1 que puede ver fuertemente. De manera similar, en la ronda R+3, cada testigo vota de acuerdo con la mayoría de votos del testigo de ronda R+2 que puede ver fuertemente. Esto puede continuar durante múltiples rondas. En caso de un empate, el voto puede ser establecido en SÍ. En otras instancias, el empate puede ser establecido en NO o puede ser establecido aleatoriamente. Si cualquier ronda tiene al menos M de los testigos votando NO, entonces la elección finaliza, y X no es famoso. Si cualquier ronda tiene al menos M de los testigos votando SÍ, entonces la elección finaliza, y X es famoso. Si ni SÍ ni NO tiene al menos M votos, la elección continúa a la siguiente ronda.

Como ejemplo, en la figura 14, considerar algún primer evento X de ronda que está debajo de la figura mostrada. Luego, cada primera ronda 1 tendrá un voto sobre si X es famoso. El evento 1412 puede identificar fuertemente los primeros eventos 1401, 1402, y 1408 de ronda 1. Por lo que su voto se basará en sus votos. Si esto es una ronda de mayoría, entonces 1412 verificará si al menos M de {1401, 1402, 1408} tiene un voto de SÍ. Si lo hacen, entonces la decisión es SÍ, y ha sido logrado el acuerdo. Si al menos M de ellos votan NO, entonces la decisión es NO, y ha sido logrado el acuerdo. Si el voto no tiene al menos M en ninguna dirección, entonces 1412 es dado un voto que es una mayoría de los votos de los de 1401, 1402, y 1408 (y rompería empates votando SÍ, si hubiera un empate). Ese voto sería usado luego en la siguiente ronda, continuando hasta que se llegue a un acuerdo.

En algunas instancias, si la ronda G es una ronda de monedas, los votos pueden ser calculados como sigue. Si el evento X puede identificar al menos M primeras rondas G-1 votando V (donde V es ya sea "SÍ" o "NO"), entonces el evento X cambiará su voto a V. De lo contrario, si la ronda G es una ronda de monedas, entonces cada primer evento X de ronda G cambia su voto al resultado de una determinación pseudoaleatoria (similar a un lanzamiento de moneda en algunas instancias), que es definido como el bit menos significativo de la firma de evento X.

Indicado de manera similar, en tales instancias, si la elección llega a una ronda R+K (una ronda de monedas), donde K es un factor designado (por ejemplo, un múltiplo de un número tal como 3, 6, 7, 8, 16, 32 o cualquier otro número adecuado), entonces la elección no finaliza en esa ronda. Si la elección llega a esta ronda, puede continuar durante al menos una ronda más. En tal ronda, si el evento Y es un testigo de ronda R+K, entonces si puede ver fuertemente al menos M testigos de la ronda R+K-1 que están votando V, entonces Y votará V. De lo contrario, Y votará de acuerdo con a un valor aleatorio (por ejemplo, de acuerdo con un bit de la firma del evento Y (por ejemplo, bit menos significativo, bit más significativo, bit seleccionado aleatoriamente) donde 1=SÍ y 0=NO, o viceversa, de acuerdo con una marca de tiempo del evento Y, usando un protocolo de "moneda compartida" criptográfico y/o cualquier otra determinación aleatoria). Esta determinación aleatoria es impredecible antes de que se cree Y, y de este modo puede aumentar la seguridad de los eventos y el protocolo de consenso.

Por ejemplo, en la figura14, si la ronda 2 es una ronda de monedas, y la votación es sobre si algún evento antes de la ronda 1 fue famoso, entonces el evento 1412 primero verificará si al menos M de {1401, 1402, 1408} votaron SÍ, o al menos M de ellos votaron NO. Si ese es el caso, entonces 1412 votará de la misma forma. Si no hay al menos M votando en ninguna dirección, entonces 1412 tendrá un voto aleatorio o pseudoaleatorio (por ejemplo, con base en el bit menos significativo de la firma digital que Ed creó para el evento 1412 cuando él lo firmó, en el momento en que él lo creó y/o definió).

En algunas instancias, el resultado de la determinación pseudoaleatoria puede ser el resultado de un protocolo de moneda compartida criptográfico, que puede, por ejemplo, ser implementado como el bit menos significativo de una firma de umbral del número de ronda.

Puede ser construido un sistema a partir de uno cualquiera de los métodos para calcular el resultado de la determinación pseudoaleatoria descritos anteriormente. En algunas instancias, el sistema funciona en ciclos a través de los diferentes métodos en algún orden. En otras instancias, el sistema puede elegir entre los diferentes métodos de acuerdo con un patrón predefinido.

"Ronda recibida": Un evento X tiene una "ronda recibida" de R si R es el entero mínimo de tal manera que al menos la mitad de los primeros eventos de ronda famosa R (o testigos famosos) con el número de ronda R son descendientes de y/o pueden ver X. En otras instancias, puede ser usado cualquier otro porcentaje adecuado. Por ejemplo, en otra instancia, un evento X tiene una "ronda recibida" de R si R es el entero mínimo de tal manera que al menos un porcentaje predeterminado (por ejemplo, 40%, 60%, 80%, etc.) de los primeros eventos de ronda famosa R (o testigos famosos) con el número de ronda R son descendientes de y/o pueden ver X.

En algunas instancias, la "generación recibida" de evento X puede ser calculada como sigue. Encontrar cual miembro creó y/o definió cada primer evento de ronda R que puede identificar el evento X. Luego determinar el número de generación para el evento más temprano por ese miembro que puede identificar X. Luego definir la "generación recibida" de X como la mediana de esa lista.

En algunas instancias, una "marca de tiempo recibida" T de un evento X puede ser la mediana de las marcas de tiempo en los eventos que incluyen el primer evento por cada miembro que identifica y/o ve X. Por ejemplo, la marca

de tiempo recibida del evento 1401 puede ser la mediana del valor de las marcas de tiempo para los eventos 1402, 1403, 1403, y 1408. En algunas instancias, la marca de tiempo para el evento 1401 puede ser incluida en el cálculo de mediana. En otras instancias, la marca de tiempo recibida para X puede ser cualquier otro valor o combinación de los valores de las marcas de tiempo en los eventos que son los primeros eventos por cada miembro para identificar o ver X. Por ejemplo, la marca de tiempo recibida para X puede basarse en un promedio de las marcas de tiempo, una desviación estándar de las marcas de tiempo, un promedio modificado (por ejemplo, eliminando las marcas de tiempo más antiguas y más recientes del cálculo), y/o similares. En aún otras instancias, puede ser usada una mediana extendida.

En algunas instancias, el orden total y/o orden de consenso para los eventos es calculado ordenando los eventos por su ronda recibida (también denominada en este documento como un valor de orden), rompiendo los empates por su marca de tiempo recibida, y rompiendo esos empates por sus firmas. En otras instancias, el orden total para los eventos puede ser calculado ordenando los eventos por su ronda recibida, rompiendo los empates por su generación recibida, y rompiendo esos empates por sus firmas. Los párrafos anteriores especifican funciones usadas para calcular y/o definir una ronda recibida, marca de tiempo recibida, y/o la generación recibida del evento.

En otras instancias, en vez de usar la firma de cada evento, puede ser usada la firma de ese evento XORed con las firmas de los eventos famosos o testigos famosos con la misma ronda recibida y/o generación recibida en esa ronda. En otras instancias, puede ser usada cualquier otra combinación adecuada de firmas de eventos para romper empates y definir el orden de consenso de eventos.

En aún otras instancias, en vez de definir la "generación recibida" como la mediana de una lista, la "generación recibida" puede ser definida como la lista misma. Luego, cuando se ordena por generación recibida, dos generaciones recibidas pueden ser comparadas por los elementos medios de sus listas, rompiendo los empates por el elemento inmediatamente antes del medio, rompiendo esos empates por el elemento inmediatamente después del medio, y continuando alternando entre el elemento antes de los usados hasta ahora y el elemento después, hasta que se rompa el empate.

En algunas instancias, la marca de tiempo de mediana puede ser reemplazada con una "mediana extendida". En tales instancias, puede ser definida una lista de marcas de tiempo para cada evento en vez de una única marca de tiempo recibida. La lista de marcas de tiempo para un evento X puede incluir el primer evento por cada miembro que identifica y/o ve X. Por ejemplo, en la figura 14, la lista de marcas de tiempo para el evento 1401 puede incluir las marcas de tiempo para eventos 1402, 1403, 1403, y 1408. En algunas instancias, también puede ser incluida la marca de tiempo para el evento 1401. Cuando se rompe un empate con la lista de marcas de tiempo (es decir, dos eventos tienen la misma ronda recibida), las marcas de tiempo medias de cada lista del evento (o una predeterminada de la primera o segunda de las dos marcas de tiempo medias, si es de longitud uniforme) pueden ser comparadas. Si estas marcas de tiempo son las mismas, pueden ser comparadas las marcas de tiempo inmediatamente después de las marcas de tiempo medias. Si estas marcas de tiempo son las mismas, pueden ser comparadas las marcas de tiempo que preceden inmediatamente a las marcas de tiempo medias. Si estas marcas de tiempo también son las mismas, son comparadas las marcas de tiempo después de las tres marcas de tiempo ya comparadas. Esto puede continuar alternando hasta que se rompa el empate. Similar a la discusión anterior, si las dos listas son idénticas, el empate se puede romper mediante las firmas de los dos elementos.

En aún otras instancias, puede ser usada una "mediana extendida truncada" en vez de una "mediana extendida". En tal instancia, no es almacenada una lista completa de marcas de tiempo para cada evento. En vez de esto, solo unos pocos de los valores cercanos a la mitad de la lista son almacenados y usados para comparación.

La marca de tiempo mediana recibida puede potencialmente ser usada para otros propósitos además de calcular un orden total de eventos. Por ejemplo, Bob podría firmar un contrato que dice que acepta estar obligado por el contrato si y solo si hay un evento X que contiene una transacción donde Alice firma ese mismo contrato, con la marca de tiempo recibida para X que está en o antes de una cierta fecha límite. En ese caso, Bob no estaría obligado por el contrato si Alice lo firma después de la fecha límite, como se indica por la "marca de tiempo mediana recibida", como se describió anteriormente.

En algunas instancias, puede ser definido un estado de la base de datos distribuida después de que es logrado un consenso. Por ejemplo, si S(R) es el conjunto de eventos que pueden ser vistos por los testigos famosos en la ronda R, finalmente todos los eventos en S(R) tendrán una ronda recibida y marca de tiempo recibida conocidas. En ese momento, se conoce el orden de consenso para los eventos en S(R) y no cambiará. Una vez que se alcanza este punto, un miembro puede calcular y/o definir una representación de los eventos y su orden. Por ejemplo, un miembro puede calcular un valor de direccionamiento de los eventos en S(R) en su orden de consenso. El miembro puede entonces firmar digitalmente el valor de direccionamiento e incluir el valor de direccionamiento en el próximo evento que define el miembro. Esto puede ser usado para informar a los otros miembros que ese miembro ha determinado que los eventos en S(R) tienen el orden dado que no cambiará. Después de que al menos M de los miembros (o cualquier otro número o porcentaje adecuado de miembros) hayan firmado el valor de direccionamiento para S(R) (y de este modo estén de acuerdo con el orden representado por el valor de direccionamiento), esa lista de consenso de eventos junto con la lista de firmas de los miembros puede formar un único archivo (u otra estructura de datos) que puede ser usado para probar que el orden de consenso fue como se reivindicó para los eventos en S(R). En otras

instancias, si los eventos contienen transacciones que actualizan un estado del sistema de base de datos distribuida (como se describe en este documento), entonces el valor de direccionamiento puede ser del estado del sistema de base de datos distribuida después de aplicar las transacciones de los eventos en S(R) en el orden de consenso.

En algunas instancias, M (como se describió anteriormente) puede basarse en valores de peso (también denominados en este documento como valores de participación) asignados a cada miembro, en vez de solo una fracción, porcentaje y/o valor del número de miembros totales. En tal instancia, cada miembro tiene una participación asociada con su interés y/o influencia en el sistema de base de datos distribuida. Tal participación puede ser un valor de peso y/o un valor de participación. Se puede decir que cada evento definido por ese miembro tiene el valor de peso de su miembro definitorio. Entonces M puede ser una fracción de la participación total de todos los miembros y puede denominarse como un criterio y/o umbral de valor de participación. Los eventos descritos anteriormente como dependientes de M se producirán cuando un conjunto de miembros con una suma de participación de al menos M esté de acuerdo (es decir, cumplan con un criterio de valor de participación). De este modo, con base en su participación, ciertos miembros pueden tener una mayor influencia en el sistema y en cómo es derivado el orden de consenso. En algunas instancias, una transacción en un evento puede cambiar la participación de uno o más miembros, agregar nuevos miembros, y/o eliminar miembros. Si tal transacción tiene una ronda recibida de R, luego después de que ha sido calculada la ronda recibida, los eventos después de los testigos de ronda R recalcularán sus números de ronda y otra información usando las participaciones modificadas y lista modificada de miembros. En algunas instancias, los votos sobre si los eventos de ronda R son famosos pueden usar las participaciones antiguas y lista de miembros, pero los votos en las rondas después de R pueden usar las nuevas participaciones y lista de miembros.

En algunas instancias adicionales pueden ser asignados valores predeterminados de peso o participación a cada miembro del sistema de base de datos distribuida. Por consiguiente, un consenso alcanzado a través del protocolo de acuerdo Bizantino puede ser implementado con un nivel de seguridad asociado para proteger a un grupo de miembros o población de potenciales ataques de Sybil. En algunas instancias, tal nivel de seguridad puede ser garantizado matemáticamente. Por ejemplo, los atacantes pueden querer afectar el resultado de uno o más eventos reorganizando el orden parcial de eventos registrados en el *hashDAG*. Se puede hacer un ataque reorganizando uno o más ordenes parciales de *hashDAG* antes de que se alcance un consenso y/o un acuerdo final entre los miembros de una base de datos distribuida. En algunas instancias, pueden surgir controversias con respecto a la temporización en la cual se produjeron múltiples eventos en competencia. Como se discutió anteriormente, un resultado asociado con un evento puede depender del valor de M. Como tal, en algunas instancias, la determinación de si Alice o Bob actuaron primero con respecto a un evento (y/o una transacción dentro de un evento) se puede hacer cuando el número de votos o la suma de participación de los miembros votantes en acuerdo sea mayor o igual al valor de M.

Algunos tipos de ataques de reorganización de orden de eventos requieren que el atacante controle al menos una fracción o porcentaje (por ejemplo, 1/10, 1/4, 1/3, etc.) de N dependiendo del valor de M. En algunas instancias, el valor de M puede estar configurado para que sea, por ejemplo, 2/3 del grupo o población N. En tal caso, en tanto que más de 2/3 de los miembros del grupo o población no sean participantes de un ataque, puede ser llegado a un acuerdo por los miembros que no son parte del ataque y la base de datos distribuida continuará llegando a un consenso y operando según lo previsto. Además, en tal instancia, un atacante tendría que controlar al menos N menos M (N-M) miembros del grupo o población (1/3 de los miembros en este ejemplo) durante el período de ataque para detener que la base de datos converja, hacer que la base de datos distribuida converja a favor del atacante (por ejemplo, hacer que la base de datos converja en un orden injusto), converger en dos estados diferentes (por ejemplo, de tal manera que los miembros estén oficialmente de acuerdo en ambos de dos estados contradictorios) o falsificar moneda (cuando el sistema de base de datos distribuida opera con una criptomoneda).

En algunas implementaciones, pueden ser asignados pesos o participaciones a cada uno de los miembros de un grupo o población, y N será la suma total de todos sus pesos o participaciones. Por consiguiente, puede ser asignado un mayor valor de peso o participación a un subconjunto del grupo de miembros o población con base en la confianza o fiabilidad. Por ejemplo, puede ser asignado un mayor valor de peso o participación a los miembros que son menos probables de participar en un ataque o que tienen algún indicador que muestra su falta de propensión a participar en comportamientos deshonestos.

En algunas instancias, el nivel de seguridad de sistema de base de datos distribuida puede ser aumentado eligiendo M para que sea una fracción mayor de N. Por ejemplo, cuando M corresponde al menor número entero que es más de 2/3 del número de miembros de un grupo o población, N, y todos los miembros tienen igual poder de voto, un atacante necesitará tener control o influencia sobre al menos 1/3 de N para evitar que se llegue a un acuerdo entre los miembros no atacantes y hacer que la base de datos distribuida falle en llegar a un consenso. De manera similar, en tal instancia, el atacante necesitará tener control o influencia sobre al menos 1/3 de N para hacer que el sistema de base de datos distribuida converja y/o llegue a un acuerdo a favor del atacante (por ejemplo, hacer que la base de datos converja en un orden injusto), converger en dos estados diferentes (por ejemplo, de tal manera que los miembros estén de acuerdo oficialmente en dos estados contradictorios) o falsificar moneda (cuando el sistema de base de datos distribuida opera con una criptomoneda).

En algunas instancias, por ejemplo, un miembro deshonesto puede votar de dos formas diferentes para hacer que la base de datos distribuida converja en dos estados diferentes. Si, por ejemplo, N=300 y 100 miembros son deshonestos, hay 200 miembros honestos con, por ejemplo, 100 votando "sí" y 100 votando "no" por una transacción.

Si los 100 miembros deshonestos envían un mensaje (o evento) a los 100 votantes honestos "sí" de que los 100 miembros deshonestos votan "sí", los 100 votantes honestos "sí" creerán que el consenso final es "sí" dado que creerán que 2/3 de los miembros votan "sí". De manera similar, si los 100 miembros deshonestos envían un mensaje (o evento) a los 100 votantes honestos "no" de que los 100 miembros deshonestos votan "no", los 100 votantes honestos "no" creerán que el consenso final es "no" dado que creerán que 2/3 de los miembros votan "no". De este modo, en esta situación, algunos miembros honestos creerán que el consenso es "sí" mientras que otros miembros honestos creerán que el consenso es "no", haciendo que la base de datos distribuida converja en dos estados diferentes. Sin embargo, si el número de miembros deshonestos es menor que 100, los miembros honestos finalmente convergerán en un único valor (ya sea "sí" o "no") dado que los miembros deshonestos no podrán presionar tanto los votos "sí" como "no" por encima de 200 (es decir, 2/3 de N). Pueden ser usados otros valores adecuados de M de acuerdo con las especificaciones y/o requisitos específicos de una aplicación del sistema de base de datos distribuida.

En algunas instancias adicionales, cuando los miembros tienen un poder de voto desigual, por ejemplo, cuando los votantes más fiables o confiables tienen un poder de voto (por ejemplo, un valor de peso o valor de participación) de una unidad mientras que el resto de los miembros tienen una fracción de una unidad, puede llegarse a un acuerdo cuando la suma de participación o peso alcanza el valor de M. De este modo, en algunas instancias, a veces se puede llegar a un acuerdo incluso cuando una mayoría de miembros están en desacuerdo con una decisión final pero una mayoría de miembros fiables o confiables están de acuerdo. En otras palabras, el poder de voto de miembros que no son de confianza puede ser diluido para prevenir o mitigar posibles ataques. Por consiguiente, en algunas instancias, el nivel de seguridad puede ser aumentado requiriendo el consenso de miembros con una participación total de M, en vez de solo un conteo de M miembros. Un valor más alto para M significa que una porción mayor de la participación (por ejemplo, un mayor número de miembros en un sistema no ponderado) tiene que estar de acuerdo para hacer que el sistema de base de datos distribuida converja.

En algunas instancias, el sistema de base de datos distribuida puede soportar múltiples protocolos de seguridad de participación incluyendo pero no limitado a los protocolos que se muestran en la Tabla 1 y cualquier combinación de los mismos. Los protocolos que se muestran en la Tabla 1 describen múltiples técnicas para asignar una participación o peso a los miembros de un grupo o una población. Los protocolos en la Tabla 1 pueden ser implementados con criptomonedas tales como, por ejemplo, Bitcoin, un derivado de una criptomoneda nativa, una criptomoneda definida dentro del sistema de base de datos distribuida o cualquier otro tipo adecuado de criptomoneda. Aunque se describe con respecto a la criptomoneda, en otras instancias los protocolos que se muestran en la Tabla 1 pueden ser usados en cualquier otro sistema de base de datos distribuida adecuado, y con cualquier otro método para asignar participación.

TABLA 1

| Ejemplo de protocolos de seguridad de participación | |
|---|---|
| *Nombre de protocolo* | *Descripción* |
| Prueba de participación | Cada miembro puede asociarse en sí mismo con una o más carteras de criptomonedas de su propiedad, y su participación de voto se establece en un valor asociado al saldo total de esas una o más carteras de criptomonedas. |
| Prueba de quema | Similar a la prueba de participación, pero los miembros demuestran que destruyeron o consumieron la criptomoneda en cuestión. En otras palabras, se paga una tarifa para unirse a la población votante, y la participación de voto es proporcional a la cantidad pagada. |
| Prueba de trabajo | Miembros pueden ganar participaciones de voto ejecutando tareas computacionales o resolviendo un rompecabezas. A diferencia de criptomonedas convencionales, se puede incurrir en el coste computacional para ganar poder de participación de voto, en vez de extraer un bloque. |
| | En este protocolo puede ser requerido que los miembros de una población sigan realizando tareas computacionales o resolviendo acertijos para mantenerse al día con las demandas del sistema, y no perder su capacidad para mantener el sistema seguro. Este protocolo también puede estar configurado para hacer que las participaciones de voto decaigan con el tiempo, para fomentar el trabajo continuo de los miembros. |
| Autorizado | Cada miembro obtiene una participación de voto de exactamente una unidad. Solo se puede permitir que los miembros se conviertan en un miembro si tienen autorización. La autorización para unirse a la población puede estar |

| | |
|---|---|
| | sujeta a un voto de los miembros existentes, una prueba de pertenencia en alguna organización existente, o cualquier otra condición adecuada. |
| Híbrido | Los miembros fundadores de una población pueden iniciar con una participación de voto equitativa. Los miembros fundadores pueden invitar a otros miembros a unirse a la población, por lo que la pertenencia se puede propagar de manera viral. Cada miembro dividirá su propia participación de voto con aquellos a los que invite. De esta forma, si un miembro invitara a 1000 impostores a ser miembros, entonces todos los 1001 de ellos juntos seguirían teniendo la misma participación de voto total que el miembro tenía originalmente. Así que los impostores no ayudarán a lanzar un ataque de Sybil. |
| Trivial | A cada miembro se le proporciona una participación de voto que corresponde a una unidad. Cualquier miembro puede invitar a nuevos miembros a unirse a la población. A los nuevos miembros se les proporciona una participación de voto que corresponde a una unidad. |

La selección de un protocolo de seguridad de participación sobre otro puede depender de aplicaciones específicas. Por ejemplo, el protocolo híbrido puede ser adecuado para la implementación de transacciones casuales de bajo valor, una aplicación de colaboración empresarial, un juego de ordenador y otro tipo similar de aplicaciones en donde la compensación entre la seguridad y el gasto computacional mínimo se inclina hacia lo último. El protocolo híbrido puede ser efectivo para evitar que un único miembro disgustado interrumpa o ataque a un grupo de miembros o población.

Para otro ejemplo, el protocolo autorizado puede ser deseable cuando los requisitos de seguridad son la máxima prioridad y los miembros de población no son completos extraños o desconocidos. El protocolo autorizado puede ser usado para implementar aplicaciones dirigidas a, por ejemplo, bancos y tipos similares de entidades financieras o entidades vinculadas en un consorcio. En tal caso, los bancos en un consorcio pueden convertirse en miembros de la población y cada banco puede estar restringido a participar como un único miembro. Por consiguiente, M puede ser establecido en el menor entero que sea más de dos tercios de la población. Los bancos pueden no confiar mutuamente entre sí como una entidad individual pero pueden depender del nivel de seguridad proporcionado por el sistema de base de datos distribuida, que en este ejemplo restringe el número de miembros deshonestos para que no sean más de un tercio de los bancos en la población.

Para aún ejemplo más, el protocolo de prueba quema puede ser implementado cuando la población incluye un gran número de miembros extraños o desconocidos. En tal caso, un atacante puede obtener el control sobre una fracción de la participación total que exceda el valor dado a M. Sin embargo, la tarifa de entrada puede establecerse lo suficientemente alta de tal manera que el coste de un ataque exceda cualquier beneficio o ganancia esperados.

Para otro ejemplo, el protocolo de prueba de participación puede ser adecuado para grupos más grandes. El protocolo de prueba de participación puede ser una solución óptima o deseable cuando hay un gran grupo de miembros que poseen cantidades sustanciales de la criptomoneda en partes aproximadamente iguales, y no se prevé ni es probable que un miembro disruptivo adquiera una mayor cantidad de criptomonedas que la cantidad de propiedad colectiva por el gran grupo de miembros.

En otras instancias, otros protocolos adicionales o más complejos pueden ser derivados de un protocolo o una combinación de protocolos que se muestran en la Tabla 1. Por ejemplo, un sistema de base de datos distribuida puede estar configurado para implementar un protocolo híbrido, que sigue un protocolo autorizado para un período predeterminado de tiempo y finalmente permite a los miembros vender participaciones de voto entre sí. Para otro ejemplo, puede estar configurado un sistema de base de datos distribuida para implementar un protocolo de prueba de quema, y finalmente hacer transición a un protocolo de prueba de participación una vez que el valor de los eventos o transacciones involucradas alcance un umbral o un valor de criptomoneda predeterminado.

Los términos, definiciones, y algoritmos anteriores son usados para ilustrar las realizaciones y conceptos descritos en las figuras 14-17b. Las figuras 16a y 16b ilustran una primera aplicación de ejemplo de un método y/o proceso de consenso mostrado en forma matemática. Las figuras 17a y 17b ilustran una segunda aplicación de ejemplo de un método y/o proceso de consenso mostrado en forma matemática.

En otras instancias y como se describe con más detalle en este documento, el módulo 211 de convergencia de base de datos puede definir inicialmente un vector de valores para un parámetro, y puede actualizar el vector de valores a medida que recibe valores adicionales para el parámetro de otros dispositivos de cálculo. Por ejemplo, el módulo 211 de convergencia de base de datos puede recibir valores adicionales para el parámetro de otros dispositivos de cálculo a través del módulo 212 de comunicación. En algunas instancias, el módulo de convergencia de base de datos puede seleccionar un valor para el parámetro con base en el vector definido y/o actualizado de valores para el parámetro, como se describe con más detalle en este documento. En algunas realizaciones, el módulo 211 de convergencia de

base de datos también puede enviar un valor para el parámetro a otros dispositivos de cálculo a través del módulo 212 de comunicación, como se describe con más detalle en este documento.

En algunas realizaciones, el módulo 211 de convergencia de base de datos puede enviar una señal a la memoria 220 para hacer que sea almacenado en la memoria 220 (1) el vector definido y/o actualizado de valores para un parámetro, y/o (2) el valor seleccionado para el parámetro con base en el vector definido y/o actualizado de valores para el parámetro. Por ejemplo, (1) el vector definido y/o actualizado de valores para el parámetro y/o (2) el valor seleccionado para el parámetro con base en el vector definido y/o actualizado de valores para el parámetro, puede ser almacenado en una instancia 221 de base de datos distribuida implementada en la memoria 220. En algunas realizaciones, la instancia 221 de base de datos distribuida puede ser similar a las instancias 114, 124, 134, 144 de base de datos distribuida del sistema 100 de base de datos distribuida mostrado en la figura 1.

En la figura 2, el módulo 211 de convergencia de base de datos y el módulo 212 de comunicación se muestran en la figura 2 como se implementa en el procesador 210. En otras realizaciones, el módulo 211 de convergencia de base de datos y/o el módulo 212 de comunicación pueden ser implementados en la memoria 220. En aún otras realizaciones, el módulo 211 de convergencia de base de datos y/o el módulo 212 de comunicación pueden ser basados en hardware (por ejemplo, ASIC, FPGA, etc.).

La figura 7 ilustra un diagrama de flujo de señales de dos dispositivos de cálculo que sincronizan eventos, de acuerdo con una realización. Específicamente, en algunas realizaciones, las instancias 703 y 803 de bases de datos distribuidas pueden intercambiar eventos para obtener convergencia. El dispositivo 700 de cálculo puede seleccionar sincronizarse con el dispositivo 800 de cálculo de manera aleatoria, con base en una relación con el dispositivo 700 de cálculo, con base en la proximidad al dispositivo 700 de cálculo, con base en una lista ordenada asociada con el dispositivo 700 de cálculo, y/o similar. En algunas realizaciones, debido a que el dispositivo 800 de cálculo puede ser elegido por el dispositivo 700 de cálculo del conjunto de dispositivos de cálculo que pertenecen al sistema de base de datos distribuida, el dispositivo 700 de cálculo puede seleccionar el dispositivo 800 de cálculo múltiples veces seguidas o puede no seleccionar el dispositivo 800 de cálculo por algún tiempo. En otras realizaciones, una indicación de los dispositivos de cálculo seleccionados previamente puede ser almacenada en el dispositivo 700 de cálculo. En tales realizaciones, el dispositivo 700 de cálculo puede esperar un número predeterminado de selecciones antes de poder seleccionar de nuevo el dispositivo 800 de cálculo. Como se explicó anteriormente, las instancias 703 y 803 de base de datos distribuida pueden ser implementadas en una memoria de dispositivo 700 de cálculo y una memoria de dispositivo 800 de cálculo, respectivamente.

Las figuras 3-6 ilustran ejemplos de un *hashDAG*, de acuerdo con una realización. Hay cinco miembros, cada uno de los cuales está representado por una línea vertical oscura. Cada círculo representa un evento. Las dos líneas hacia abajo de un evento representan los direccionamientos de dos eventos previos. Cada evento en este ejemplo tiene dos líneas hacia abajo (una línea oscura al mismo miembro y una línea clara a otro miembro), excepto para cada primer evento del miembro. El tiempo avanza hacia arriba. En las figuras 3-6, los dispositivos de cálculo de una base de datos distribuida son indicados como Alice, Bob, Carol, Dave y Ed. Debe entenderse que tales indicaciones se refieren a dispositivos de cálculo estructural y funcionalmente similares a los dispositivos 110, 120, 130, y 140 de cálculo mostrados y descritos con respecto a la figura 1.

Sistema de Ejemplo 1: si el dispositivo 700 de cálculo se denomina Alice, y el dispositivo 800 de cálculo se denomina Bob, entonces una sincronización entre ellos puede ser como se ilustra en la figura 7. Una sincronización entre Alice y Bob puede ser como sigue:

- Alice envía a Bob los eventos almacenados en la base de datos 703 distribuida.

- Bob crea y/o define un nuevo evento que contiene:

-- un direccionamiento del último evento que Bob creó y/o definió

-- un direccionamiento del último evento que Alice creó y/o definió

-- una firma digital de Bob de los anteriores

- Bob envía a Alice los eventos almacenados en la base de datos 803 distribuida.

- Alice crea y/o define un nuevo evento.

- Alice envía a Bob ese evento.

- Alice calcula un orden total para los eventos, como una función de un *hashDAG*

- Bob calcula un orden total para los eventos, como una función de un *hashDAG*

En cualquier momento dado, un miembro puede almacenar los eventos recibidos hasta el momento, junto con un identificador asociado con el dispositivo de cálculo y/o instancia de base de datos distribuida que creó y/o definió cada evento. Cada evento contiene los direccionamientos de dos eventos anteriores, excepto un evento inicial (que no tiene

direccionamiento de padres), y el primer evento para cada miembro nuevo (que tiene un único direccionamiento de evento de padre, que representa el evento del miembro existente que los invitó a unirse). Se puede dibujar un diagrama que representa este conjunto de eventos. Puede mostrar una línea vertical para cada miembro, y un punto en esa línea para cada evento creado y/o definido por ese miembro. Se dibuja una línea diagonal entre dos puntos siempre que un evento (el punto más alto) incluya el direccionamiento de un evento anterior (el punto más bajo). Se puede decir que un evento está vinculado a otro evento si ese evento puede hacer referencia al otro evento a través de un direccionamiento de ese evento (ya sea directamente o a través de eventos intermediarios).

Por ejemplo, la figura 3 ilustra un ejemplo de *hashDAG* 600. El evento 602 es creado y/o definido por Bob como resultado de y después de la sincronización con Carol. El evento 602 incluye un direccionamiento del evento 604 (el evento previo creado y/o definido por Bob) y un direccionamiento de evento 606 (el evento previo creado y/o definido por Carol). En algunas realizaciones, por ejemplo, el direccionamiento de evento 604 incluido dentro del evento 602 incluye un puntero a sus eventos antepasados inmediatos, eventos 608 y 610. Como tal, Bob puede usar el evento 602 para hacer referencia a eventos 608 y 610 y reconstruir el *hashDAG* usando los punteros de los eventos anteriores. En algunas instancias, se puede decir que el evento 602 está vinculado a los otros eventos en el *hashDAG* 600 dado que el evento 602 puede hacer referencia a cada uno de los eventos en el *hashDAG* 600 a través de eventos antepasados anteriores. Por ejemplo, evento 602 está vinculado al evento 608 a través del evento 604. Para otro ejemplo, evento 602 está vinculado al evento 616 a través de eventos 606 y evento 612.

Sistema de Ejemplo 2: El sistema del Sistema de Ejemplo 1, donde el evento también incluye una "carga útil" de transacciones u otra información para registrar. Tal carga útil puede ser usada para actualizar los eventos con cualquier transacción y/o información que se produjo y/o fue definida desde el evento anterior inmediato del dispositivo de cálculo. Por ejemplo, el evento 602 puede incluir cualquier transacción realizada por Bob desde que fue creado y/o definido el evento 604. De este modo, cuando se sincroniza el evento 602 con otros dispositivos de cálculo, Bob puede compartir esta información. Por consiguiente, las transacciones realizadas por Bob pueden ser asociadas con un evento y compartidas con los otros miembros usando eventos.

Sistema de Ejemplo 3: El sistema del Sistema de Ejemplo 1, donde el evento también incluye la hora y/o fecha actual, útil para depuración, diagnóstico, y/u otros propósitos. La hora y/o fecha pueden ser la hora y/o fecha local cuando el dispositivo de cálculo (por ejemplo, Bob) crea y/o define el evento. En tales realizaciones, tal hora y/o fecha local no está sincronizada con los dispositivos restantes. En otras realizaciones, la hora y/o fecha pueden ser sincronizadas a través de los dispositivos (por ejemplo, cuando se intercambian eventos). En aún otras realizaciones, puede ser usado un temporizador global para determinar la hora y/o fecha.

Sistema de Ejemplo 4: El sistema del Sistema de Ejemplo 1, donde Alice no envía a Bob eventos creados y/o definidos por Bob, ni eventos antepasados de tal evento. Un evento x es un antepasado de un evento y si y contiene el direccionamiento de x, o y contiene el direccionamiento de un evento que es un antepasado de x. Indicado de manera similar, en tales realizaciones Bob envía a Alice los eventos aún no almacenados por Alice y no envía eventos ya almacenados por Alice.

Por ejemplo, la figura 4 ilustra un *hashDAG* 620 de ejemplo que ilustra los eventos antepasados (círculos de puntos) y eventos descendientes (círculos de rayas) del evento 622 (el círculo negro). Las líneas establecen un orden parcial en los eventos, donde los antepasados vienen antes del evento negro, y los descendientes vienen después del evento negro. El orden parcial no indica si los eventos blancos son antes o después del evento negro, por lo que es usado un orden total para decidir su secuencia. Para otro ejemplo, la figura 5 ilustra un *hashDAG* de ejemplo que ilustra un evento particular (círculo sólido) y la primera vez que cada miembro recibe una indicación de ese evento (círculos de rayas). Cuando Carol se sincroniza con Dave para crear y/o definir el evento 624, Dave no envía a Carol los eventos antepasados del evento 622 dado que Carol ya conoce y ha recibido tales eventos. En vez de esto, Dave envía a Carol los eventos que Carol aún tiene que recibir y/o almacenar en la instancia de base de datos distribuida de Carol. En algunas realizaciones, Dave puede identificar qué eventos enviar a Carol con base en lo que revela el *hashDAG* de Dave sobre qué eventos que Carol ha recibido previamente. El evento 622 es un antepasado del evento 626. Por lo tanto, en el momento del evento 626, Dave ya ha recibido el evento 622. La figura 4 muestra que Dave recibió el evento 622 de Ed quien recibió el evento 622 de Bob quien recibió el evento 622 de Carol. Adicionalmente, en el momento del evento 624, el evento 622 es el último evento que Dave ha recibido que fue creado y/o definido por Carol. Por lo tanto, Dave puede enviar a Carol los eventos que Dave ha almacenado aparte del evento 622 y sus antepasados. Adicionalmente, tras recibir el evento 626 de Dave, Carol puede reconstruir el *hashDAG* con base en los punteros en los eventos almacenados en la instancia de base de datos distribuida de Carol. En otras realizaciones, Dave puede identificar qué eventos enviar a Carol con base en que Carol envía el evento 622 a Dave (no se muestra en la figura 4) y Dave identifica usando el evento 622 (y las referencias en el mismo) para identificar los eventos que Carol ya ha recibido.

Sistema de Ejemplo 5: El sistema del Sistema de Ejemplo 1 donde ambos miembros envían eventos al otro en un orden de tal manera que un evento no es enviado hasta después de que el receptor haya recibido y/o almacenado los antepasados de ese evento. Por consiguiente, el remitente envía eventos del más antiguo al más nuevo, de tal manera que el receptor puede verificar los dos direccionamientos en cada evento a medida que es recibido el evento, comparando los dos direccionamientos con los dos eventos antepasados que ya fueron recibidos. El remitente puede identificar qué eventos enviar al receptor con base en el estado actual del *hashDAG* del remitente (por ejemplo, una

variable de estado de base de datos definida por el remitente) y lo que ese *hashDAG* indica que el receptor ya ha recibido. Con referencia a la figura 3, por ejemplo, cuando Bob está sincronizándose con Carol para definir el evento 602, Carol puede identificar que el evento 619 es el último evento creado y/o definido por Bob que Carol ha recibido. Por lo tanto Carol puede determinar que Bob conoce de ese evento, y sus antepasados. De este modo Carol puede enviar a Bob el evento 618 y evento 616 primero (es decir, los eventos más antiguos que Bob aún tiene que recibir y que Carol ha recibido). Carol puede luego enviar a Bob el evento 612 y luego evento 606. Esto le permite a Bob vincular fácilmente los eventos y reconstruir el *hashDAG* de Bob. El uso del *hashDAG* de Carol para identificar qué eventos aún tiene que recibir Bob puede aumentar la eficiencia de la sincronización y puede reducir el tráfico de red dado que Bob no solicita eventos de Carol.

En otras realizaciones, el evento más reciente puede ser enviado primero. Si el receptor determina (con base en el direccionamiento de los dos eventos previos en el evento más reciente y/o punteros a eventos previos en el evento más reciente) que aún no han recibido uno de los dos eventos previos, el receptor puede solicitar el remitente para enviar tales eventos. Esto puede producirse hasta que el receptor haya recibido y/o almacenado los antepasados del evento más reciente. Con referencia a la figura 3, en tales realizaciones, por ejemplo, cuando Bob recibe el evento 606 de Carol, Bob puede identificar el direccionamiento del evento 612 y evento 614 en evento 606. Bob puede determinar que el evento 614 fue recibido previamente de Alice cuando se creó y/o definió el evento 604. Por consiguiente, Bob no necesita solicitar el evento 614 de Carol. Bob también puede determinar que aún no ha sido recibido el evento 612. Bob puede luego solicitar el evento 612 de Carol. Luego Bob puede, con base en los direccionamientos dentro del evento 612, determinar que Bob no ha recibido los eventos 616 o 618 y por consiguiente puede solicitar estos eventos de Carol. Con base en los eventos 616 y 618, Bob podrá luego determinar que él ha recibido los antepasados del evento 606.

Sistema de Ejemplo 6: El sistema del Sistema de Ejemplo 5 con la restricción adicional de que cuando un miembro tiene una elección entre varios eventos para enviar a continuación, el evento se elige para minimizar el número total de bytes enviados hasta el momento creados y/o definidos por ese miembro. Por ejemplo, si a Alice solo le quedan dos eventos para enviar a Bob, y uno es 100 bytes y fue creado y/o definido por Carol, y uno es 10 bytes y fue creado y/o definido por Dave, y hasta el momento en esta sincronización Alice ya ha enviado 200 bytes de eventos de Carol y 210 de Dave, luego Alice debería enviar primero el evento de Dave, luego enviar subsecuentemente el evento de Carol. Debido a que 210 + 10 < 100 + 200. Esto puede ser usado para abordar ataques en los cuales un único miembro ya sea envía un único evento gigantesco, o una inundación de eventos pequeños. En el caso en el cual el tráfico exceda un límite de bytes de la mayoría de miembros (como se discutió con respecto al Sistema de Ejemplo 7), el método del Sistema de Ejemplo 6 puede asegurar que los eventos del atacante sean ignorados en vez de los eventos de usuarios legítimos. Indicado de manera similar, los ataques pueden ser reducidos enviando los eventos más pequeños antes que los más grandes (para defenderse contra un evento gigante que une una conexión). Además, si un miembro no puede enviar cada uno de los eventos en una única sincronización (por ejemplo, debido a la limitación de red, límites de bytes de miembros, etc.), entonces ese miembro puede enviar algunos eventos de cada miembro, en vez de simplemente enviar los eventos definidos y/o creados por el atacante y ninguno (de pocos) eventos creados y/o definidos por otros miembros.

Sistema de Ejemplo 7: El sistema del Sistema de Ejemplo 1 con una primera etapa adicional en la cual Bob envía a Alice un número que indica un número máximo de bytes que él está dispuesto a recibir durante esta sincronización, y Alice responde con su límite. Alice luego deja de enviar cuando el siguiente evento excedería este límite. Bob hace lo mismo. En tal realización, esto limita el número de bytes transferidos. Esto puede aumentar el tiempo de convergencia, pero reducirá la cantidad de tráfico de red por sincronización.

Sistema de Ejemplo 8: El sistema del Sistema de Ejemplo 1, en el cual se agregaron las siguientes etapas al inicio del proceso de sincronización:

- Alice identifica S, el conjunto de eventos que ella ha recibido y/o almacenado, omitiendo eventos que fueron creados y/o definidos por Bob o que son antepasados de eventos creados y/o definidos por Bob.

- Alice identifica a los miembros que crearon y/o definieron cada evento en S, y envía a Bob la lista de los números de ID de los miembros. Alice también envía un número de eventos que fueron creados y/o definidos por cada miembro que ella ya ha recibido y/o almacenado.

- Bob responde con una lista de cuántos eventos él ha recibido que fueron creados y/o definidos por los otros miembros.

- Alice luego envía a Bob solo los eventos que él aún tiene que recibir. Por ejemplo, si Alice indica a Bob que ella ha recibido 100 eventos creados y/o definidos por Carol, y Bob responde que él ha recibido 95 eventos creados y/o definidos por Carol, entonces Alice enviará solo los 5 eventos más recientes creados y/o definidos por Carol.

Sistema de Ejemplo 9: El sistema del Sistema de Ejemplo 1, con un mecanismo adicional para identificar y/o manejar a los tramposos. Cada evento contiene dos direccionamientos, uno del último evento creado y/o definido por ese miembro (el "direccionamiento propio"), y uno del último evento creado y/o definido por otro miembro (el "direccionamiento foráneo"). Si un miembro crea y/o define dos eventos diferentes con el mismo direccionamiento

propio, entonces ese miembro es un "tramposo". Si Alice descubre que Dave es un tramposo, al recibir dos eventos diferentes creados y/o definidos por él con el mismo direccionamiento propio, entonces ella almacena un indicador de que él es un tramposo, y se abstiene de sincronizarse con él en el futuro. Si ella descubre que él es un tramposo y aun así se sincroniza con él de nuevo y crea y/o define un nuevo evento que registra ese hecho, entonces Alice también se convierte en una tramposa, y los otros miembros que aprenden de que Alice se está sincronizando además con Dave dejan de sincronizarse con Alicia. En algunas realizaciones, esto solo afecta a las sincronizaciones de una forma. Por ejemplo, cuando Alice envía una lista de identificadores y el número de eventos que ella ha recibido para cada miembro, ella no envía una ID ni cuenta para el tramposo, por lo que Bob no responderá con ningún número correspondiente. Alice luego envía a Bob los eventos del tramposo que ella ha recibido y por los cuales ella no ha recibido una indicación de que Bob haya recibido tales eventos. Después de que es finalizada la sincronización, Bob también podrá determinar que Dave es un tramposo (si él aún no ha identificado a Dave como un tramposo), y Bob también se negará a sincronizar con el tramposo.

Sistema de Ejemplo 10: El sistema en Sistema de Ejemplo 9, con la adición de que Alice inicia un proceso de sincronización enviando a Bob una lista de tramposos que ella ha identificado y de cuyos eventos ella todavía está almacenando, y Bob responde con cualquier tramposo que él haya identificado además de los tramposos que Alice identificó. Luego continúan con normalidad, pero sin dar cuentas para los tramposos cuando se sincronizan entre sí.

Sistema de Ejemplo 11: El sistema en Sistema de Ejemplo 1, con un proceso que actualiza repetidamente un estado actual (por ejemplo, como capturado por una variable de estado de base de datos definida por un miembro del sistema) con base en transacciones dentro de cualquier evento nuevo que es recibido durante sincronización. Esto también puede incluir un segundo proceso que reconstruye repetidamente ese estado (por ejemplo, el orden de eventos), cada vez que cambia la secuencia de eventos, volviendo a una copia de un estado anterior, y recalculando el estado presente procesando los eventos en el nuevo orden. En algunas realizaciones, el estado actual es un estado, saldo, condición, y/o similar asociado con un resultado de las transacciones. Indicado de manera similar, el estado puede incluir la estructura de datos y/o variables modificadas por las transacciones. Por ejemplo, si las transacciones son transferencias de dinero entre cuentas bancarias, entonces el estado actual puede ser el saldo actual de las cuentas. Para otro ejemplo, si las transacciones están asociadas con un juego de multijugador, el estado actual puede ser la posición, número de vidas, ítems obtenidos, estado del juego, y/o similares asociados con el juego.

Sistema de Ejemplo 12: El sistema en Sistema de Ejemplo 11, hecho más rápido mediante el uso de *arrayList* de "clon rápido" para mantener el estado (por ejemplo, saldos de cuentas bancarias, estado de juego, etc.). Una *arrayList* de clon rápido es una estructura de datos que actúa similar a un arreglo con una característica adicional: soporta una operación de "clon" que parece crear y/o definir un nuevo objeto que es una copia del original. El clon actúa como si fuera una copia verdadera, debido a que los cambios en el clon no afectan al original. Sin embargo, la operación de clonación es más rápida que la creación de una copia verdadera, debido a que la creación de un clon no involucra en realidad copiar y/o actualizar todo el contenido de una *arrayList* a otra. En vez de tener dos clones y/o copias de la lista original, pueden ser usados dos objetos pequeños, cada uno con una tabla de direccionamiento y un puntero a la lista original. Cuando se hace una escritura en el clon, la tabla de direccionamiento recuerda cual elemento es modificado, y el nuevo valor. Cuando se realiza una lectura en una ubicación, primero es verificada la tabla de direccionamiento, y si ese elemento fue modificado, es regresado el nuevo valor de la tabla de direccionamiento. De lo contrario, es regresado ese elemento de la *arrayList* original. De esta forma, los dos "clones" son inicialmente solo punteros a la *arrayList* original. Pero a medida que cada uno es modificado repetidamente, crece para tener una gran tabla de direccionamiento que almacena diferencias entre el mismo y la lista original. Los clones pueden en sí mismos ser clonados, lo que hace que la estructura de datos se expanda a un árbol de objetos, cada uno con su propia tabla de direccionamiento y un puntero a su padre. Por lo tanto una lectura provoca un recorrido por el árbol hasta que se encuentra un vértice que tiene los datos solicitados, o se alcanza la raíz. Si el vértice se vuelve demasiado grande o complejo, entonces puede ser reemplazado con una copia verdadera del padre, los cambios en la tabla de direccionamiento pueden ser hechos en la copia, y la tabla de direccionamiento es descartada. Además, si ya no es necesario un clon, entonces durante la recolección de basura puede ser eliminado del árbol, y el árbol puede ser colapsado.

Sistema de Ejemplo 13: El sistema en Sistema de Ejemplo 11, hecho más rápido mediante el uso de una tabla de direccionamiento de "clon rápido" para mantener el estado (por ejemplo, saldos de cuentas bancarias, estado de juego, etc.). Es lo mismo que el Sistema 12, excepto que la raíz del árbol es una tabla de direccionamiento en vez de una *arrayList*.

Sistema de Ejemplo 14: El sistema en Sistema de Ejemplo 11, hecho más rápido mediante el uso de una base de datos relacional de "clon rápido" para mantener el estado (por ejemplo, saldos de cuentas bancarias, estado de juego, etc.). Este es un objeto que actúa como un envoltorio alrededor de un Sistema de Gestión de Bases de datos Relacionales (RDBMS) existente. Cada aparente "clon" es en realidad un objeto con un número de ID y un puntero a un objeto que contiene la base de datos. Cuando el código del usuario intenta realizar una consulta de Lenguaje de Consulta de Estructura (SQL) en la base de datos, esa consulta es modificada primero, luego enviada a la base de datos real. La base de datos real es idéntica a la base de datos como se ve por el código de cliente, excepto que cada tabla tiene un campo adicional para el ID de clon. Por ejemplo, suponer que hay una base de datos original con ID de clon 1, y luego se hacen dos clones de la base de datos, con IDs 2 y 3. Cada fila en cada tabla tendrá un 1, 2, o 3 en el campo de ID de clon. Cuando una consulta proviene del código de usuario en el clon 2, la consulta es modificada

de tal manera que la consulta solo leerá de las filas que tienen un 2 o 1 en ese campo. De manera similar, lee hasta 3 búsqueda de filas con un ID de 3 o 1. Si el comando de Lenguaje de Consulta Estructurado (SQL) va al clon 2 y dice que elimine una fila, y esa fila tiene un 1, entonces el comando debería solo cambiar el 1 por un 3, lo cual marca la fila como si ya no es compartida por los clones 2 y 3, y ahora solo son visibles para 3. Si hay varios clones en operación, luego pueden ser insertadas varias copias de la fila, y cada una puede ser cambiada a la ID de un clon diferente, de tal manera que las nuevas filas sean visibles a los clones excepto por el clon que acaba de "eliminar" la fila. De manera similar, si se agrega una fila al clon 2, entonces la fila es agregada a la tabla con una ID de 2. Una modificación de una fila es equivalente a una eliminación y luego a una inserción. Como antes, si son recolectados de basura varios clones, entonces el árbol puede ser simplificado. La estructura de ese árbol será almacenada en una tabla adicional que no es accesible para los clones, pero que es usada puramente de manera interna.

Sistema de Ejemplo 15: El sistema en Sistema de Ejemplo 11, hecho más rápido mediante el uso de un sistema de archivos de "clon rápido" para mantener el estado. Este es un objeto que actúa como envoltorio alrededor de un sistema de archivos. El sistema de archivos está construido en la parte superior del sistema de archivos existente, usando una base de datos relacional de clon rápido para gestionar las diferentes versiones del sistema de archivos. El sistema de archivos subyacente almacena un gran número de archivos, ya sea en un directorio, o divididos de acuerdo con el nombre del archivo (para mantener directorios pequeños). El árbol de directorios puede ser almacenado en la base de datos, y no se puede proporcionar al sistema de archivos de anfitrión. Cuando es clonado un archivo o directorio, el "clon" es solo un objeto con un número de ID, y la base de datos es modificada para reflejar que este clon ahora existe. Si es clonado un sistema de archivos de clon rápido, al usuario le parece como si hubiera sido creado y/o definido un nuevo disco duro completo, inicializado con una copia del disco duro existente. Los cambios en una copia pueden no tener ningún efecto en las otras copias. En realidad, solo hay una copia de cada archivo o directorio, y cuando un archivo es modificado a través de un clon se produce la copia.

Sistema de Ejemplo 16: El sistema en Sistema de Ejemplo 15 en el cual es creado y/o definido un archivo separado en el sistema operativo de anfitrión para cada porción de N bytes de un archivo en el sistema de archivos de clon rápido. N puede ser algún tamaño adecuado, tal como por ejemplo 4096 o 1024. De esta forma, si es cambiado un byte en un archivo grande, solo es copiada y modificada una porción del archivo grande. Esto también aumenta la eficiencia cuando se almacenan muchos archivos en la unidad que difieren en solo unos pocos bytes.

Sistema de Ejemplo 17: El sistema en Sistema de Ejemplo 11 donde cada miembro incluye en algunos o todos los eventos que crea y/o define un direccionamiento del estado en algún momento previo, junto con el número de eventos que se produjeron hasta ese punto, indicando que el miembro reconoce y/o identifica que ahora hay un consenso sobre el orden de eventos. Después que un miembro ha recolectado eventos firmados que contienen tal direccionamiento de la mayoría de los usuarios para un estado dado, el miembro puede luego almacenar eso como prueba del estado de consenso en ese punto, y eliminar de la memoria los eventos y transacciones antes de ese punto.

Sistema de Ejemplo 18: El sistema en Sistema de Ejemplo 1 donde las operaciones que calculan una mediana o una mayoría son reemplazadas con una mediana ponderada o mayoría ponderada, donde los miembros son ponderados por su "participación". La participación es un número que indica cuánto cuenta el voto de ese miembro. La participación podría ser tenencias en una criptomoneda, o simplemente un número arbitrario asignado cuando se invita al miembro a unirse por primera vez, y luego se divide entre nuevos miembros a los que el miembro invita a unirse. Los eventos antiguos pueden ser descartados cuando suficientes miembros han acordado el estado de consenso de tal manera que su participación total sea una mayoría de la participación en existencia. Si el orden total es calculado usando una mediana de rangos contribuidos por los miembros, entonces el resultado es un número donde la mitad de los miembros tienen un rango más alto y la mitad tienen un rango más bajo. Por otro lado, si el orden total es calculado usando la mediana ponderada, entonces el resultado es un número donde aproximadamente la mitad de la participación total está asociada con rangos más bajos a ese, y la mitad por encima. La votación ponderada y medianas pueden ser útiles para prevenir un ataque de Sybil, donde un miembro invita a un gran número de usuarios de "impostores" a unirse, cada uno de los cuales son simplemente seudónimos controlados por el miembro que invita. Si el miembro que invita es forzado a dividir su participación con los invitados, los impostores no serán útiles para el atacante en un intento de controlar los resultados de consenso. Por consiguiente, la prueba de participación puede ser útil en algunas circunstancias.

Sistema de Ejemplo 19: El sistema en Sistema de Ejemplo 1 en el cual en vez de una única base de datos distribuida, hay múltiples bases de datos en una jerarquía. Por ejemplo, podría haber una única base de datos de la que los usuarios son miembros, y luego varias bases de datos más pequeñas, o "porciones", cada una de las cuales tiene un subconjunto de los miembros. Cuando los eventos suceden en una porción, son sincronizados entre los miembros de esa porción y no entre miembros fuera de esa porción. Luego, de vez en cuando, después de que haya sido decidido un orden de consenso dentro de la porción, el estado resultante (o eventos con su orden total de consenso) puede ser compartido con toda la pertenencia de la gran base de datos.

Sistema de Ejemplo 20: El sistema en Sistema de Ejemplo 11, con la capacidad de tener un evento que actualice el software para actualizar el estado (por ejemplo, como capturado por una variable de estado de base de datos definida por un miembro del sistema). Por ejemplo, eventos X y Y pueden contener transacciones que modifican el estado, de acuerdo con el código de software que lee las transacciones dentro de esos eventos, y luego actualiza el estado de

manera apropiada. Luego, el evento Z puede contener un aviso de que ahora está disponible una nueva versión del software. Si un orden total dice que los eventos suceden en el orden X, Z, Y, entonces el estado puede ser actualizado procesando las transacciones en X con el software antiguo, luego las transacciones en Y con el nuevo software. Pero si el orden de consenso era X, Y, Z, entonces tanto X como Y pueden ser actualizados con el software antiguo, lo cual podría dar un estado final diferente. Por lo tanto, en tales realizaciones, el aviso para actualizar el código puede producirse dentro de un evento, de tal manera que la comunidad pueda lograr el consenso sobre cuándo conmutar de la versión antigua a la nueva versión. Esto asegura que los miembros mantendrán estados sincronizados. También asegura que el sistema pueda seguir funcionando, incluso durante las actualizaciones, sin necesidad de reinicializar o reiniciar el proceso.

Sistema de Ejemplo 21: El sistema del Sistema de Ejemplo 1, donde es implementado un protocolo de prueba de participación para llegar a un consenso y el poder de voto de cada miembro es proporcional a la cantidad de criptomoneda de propiedad del miembro. La criptomoneda de este ejemplo se denominará de aquí en adelante como *StakeCoin*. La pertenencia al grupo o población es abierta, no autorizada, de este modo, puede no haber confianza entre miembros.

El protocolo de prueba de participación puede ser computacionalmente menos costoso que otros protocolos por ejemplo, el protocolo de prueba de trabajo. En este ejemplo, M (como se describe anteriormente) puede ser 2/3 de la cantidad de *StakeCoins* de propiedad colectiva por los miembros. De este modo, el sistema de base de datos distribuida puede ser seguro (y puede converger según lo previsto) si ningún atacante puede obtener 1/3 del total de *StakeCoins* de propiedad por los miembros participantes juntos. El sistema de base de datos distribuida puede seguir funcionando a un nivel de seguridad matemáticamente garantizado en tanto que más de 2/3 de las *StakeCoins* sean propiedad de miembros activos honestos. Esto permite que la base de datos converja correctamente.

Puede ser lograda una forma para que un atacante obtenga el control sobre el sistema de base de datos distribuida negociando individualmente con los propietarios de *StakeCoin* dentro del sistema de base de datos distribuida para comprar sus *StakeCoins*. Por ejemplo, Alice puede obtener la mayoría de las *StakeCoins* comprando las *StakeCoins* propiedad de Bob, Carol y Dave dejando a Ed en una posición vulnerable. Esto es similar a monopolizar el mercado de un producto, o intentar comprar suficientes acciones en una empresa para una adquisición hostil. El escenario descrito representa no solo un ataque al sistema de base de datos distribuida que usa las *StakeCoins* sino que también es un ataque a *StakeCoin* en sí misma. Si un miembro obtiene un cuasimonopolio sobre una criptomoneda, tal miembro puede manipular el valor de mercado de criptomoneda, y disponer para vender alto y comprar bajo repetidamente. Esto puede ser rentable a corto plazo, pero en última instancia puede socavar la confianza en la criptomoneda, y tal vez llevar a que sea abandonada universalmente. Un valor de mercado de moneda puede ser independiente de la tecnología usada para transmitir la moneda. Por ejemplo, si un individuo o entidad obtiene la propiedad de la mayoría de los dólares estadounidenses en el mundo, o de la mayoría de los futuros de moneda en el mundo, entonces tal individuo o entidad puede socavar el mercado de manera rentable.

Un ataque hecho mediante la adquisición de un cuasimonopolio de una criptomoneda es más difícil si la criptomoneda es tanto valiosa como extendida. Si la criptomoneda es valiosa, entonces costará mucho comprar una gran fracción del suministro de moneda de *StakeCoin*. Si la criptomoneda está extendida, con muchas personas diferentes que poseen *StakeCoins*, entonces los intentos de monopolizar el mercado de *StakeCoin* se harán visibles desde el principio, lo cual naturalmente elevará el precio de una *StakeCoin*, haciendo aún más difícil obtener el suministro de moneda restante.

Se puede hacer un segundo tipo de ataque obteniendo una cantidad de *StakeCoins* que puede ser pequeña en comparación con la cantidad colectiva de *StakeCoins* a través de múltiples sistemas de bases de datos distribuidas, pero grande en comparación con la cantidad de *StakeCoins* de propiedad de los miembros que participan en un sistema de base de datos distribuida particular. Este tipo de ataque puede ser evitado cuando la criptomoneda es definida específicamente para uso en una aplicación del sistema de base de datos distribuida. En otras palabras, las *StakeCoins* y una implementación del sistema de base de datos distribuida pueden ser definidas simultáneamente para estar vinculadas entre sí, y cada una contribuye al valor creciente de la otra. Indicado de manera similar, no hay implementaciones adicionales de la base de datos distribuida que comercializan *StakeCoins*.

Puede ser deseable tener una criptomoneda valiosa desde el principio cuando es definida recientemente una implementación de un sistema de base de datos distribuida. Aunque la criptomoneda puede aumentar su valor con el tiempo, una criptomoneda valiosa puede ser beneficiosa en las primeras etapas del sistema. En algunas instancias, un consorcio de entidades participantes puede iniciar la criptomoneda y su sistema de base de datos distribuida asociado. Por ejemplo, diez grandes corporaciones u organizaciones respetadas que son los fundadores pueden recibir una cantidad significativa de *StakeCoins* para iniciar la criptomoneda de *StakeCoin* y el sistema de base de datos distribuida. El sistema puede estar configurado de tal manera que el suministro de criptomonedas no crecerá rápidamente, y tendrá algún límite de tamaño máximo. Cada entidad fundadora puede tener un incentivo para participar como un miembro en el sistema de base de datos distribuida y la implementación de la *StakeCoin* (por ejemplo, implementado como un sistema de base de datos distribuida que puede ser estructurado como un *hashDAG* con un algoritmo de consenso). Debido a que no hay una prueba de trabajo, puede ser económico ser un miembro participante que ejecuta un nodo. Las entidades fundadoras pueden ser lo suficientemente confiables como para que

sea poco probable que cualquier gran fracción de ellas se confabulara para socavar el sistema, especialmente debido a que eso puede destruir el valor de sus *StakeCoins* y el sistema de base de datos distribuida implementado.

En algunas implementaciones, otros miembros pueden unirse al sistema de base de datos distribuida y otras personas o entidades pueden comprar *StakeCoins*, ya sea directamente de las entidades fundadoras, o en un intercambio. El sistema de base de datos distribuida puede estar configurado para incentivar a los miembros a participar pagando pequeñas cantidades de *StakeCoins* por participar. Con el tiempo, el sistema puede volverse mucho más distribuido, con la participación finalmente extendiéndose, de tal manera que sea difícil para cualquier persona o entidad monopolizar el mercado, incluso si las entidades fundadoras se confabulan para hacer un ataque. En ese momento, la criptomoneda puede alcanzar un valor independiente; el sistema de base de datos distribuida puede tener un nivel independiente de seguridad; y el sistema se puede abrir sin requisitos de autorización (por ejemplo, sin tener que ser invitado a unirse por un miembro fundador). De este modo, ahorrando costes computacionales recurrentes exigidos por sistemas implementados con protocolos alternativos, por ejemplo, sistemas implementados con protocolos de prueba de trabajo.

Aunque se describió anteriormente con respecto al uso de una base de datos distribuida de *hashDAG*, puede ser usado cualquier otro protocolo de base de datos distribuida adecuado para implementar el Sistema de Ejemplo 21. Por ejemplo, mientras que los números de ejemplo específicos y la participación pueden cambiar, el Sistema de Ejemplo 21 puede ser usado para aumentar la seguridad de cualquier sistema de base de datos distribuida adecuado.

Se espera que los sistemas descritos anteriormente creen y/o logren un mecanismo de convergencia eficiente para consenso distribuido, con eventual consenso. Se pueden demostrar diversos teoremas sobre esto, como se muestra en lo siguiente.

Teorema de Ejemplo 1: Si el evento x precede al evento y en el orden parcial, entonces en el conocimiento de un miembro dado de los otros miembros en un momento dado, cada uno de los otros miembros habrá ya sea recibido una indicación de x antes de y, o aún no hará recibido una indicación de y.

Prueba: Si el evento x precede al evento y en el orden parcial, entonces x es un antepasado de y. Cuando un miembro recibe una indicación de y por primera vez, ese miembro ya sea ya ha recibido una indicación de x antes (en cuyo caso escucharon de x antes de y), o será el caso de que la sincronización proporcione a ese miembro tanto x como y (en cuyo caso escucharán de x antes de y durante esa sincronización, debido a que los eventos recibidos durante una única sincronización son considerados que han sido recibidos en un orden consistente con las relaciones de ascendencia descritas con respecto al Sistema de Ejemplo 5). QED

Teorema de Ejemplo 2: Para cualquier *hashDAG* dado, si x precede a y en el orden parcial, entonces x precederá a y en el orden total calculado para ese *hashDAG*.

Prueba: Si x precede a y en el orden parcial, entonces mediante el teorema 1:

para todo i, rango(i,x) < rango(i,y)

donde rango(i,x) es el rango asignado por el miembro i al evento x, que es 1 si x es el primer evento recibido por el miembro i, 2 si es el segundo, y así sucesivamente. Dejar que med(x) sea la mediana del rango(i,x) sobre todo i, y de manera similar para med(y).

Para un k dado, elegir un i1 e i2 de tal manera que el rango(i1,x) sea el k-ésimo rango x más pequeño, y el rango(i2,y) sea el k-ésimo rango más pequeño y. Entonces:

rango(i1,x) < rango(i2,y)

Esto se debe a que el rango(i2,y) es mayor que o igual a k de los rangos y, cada uno de los cuales es estrictamente mayor que el rango x correspondiente. Por lo tanto, el rango(i2,y) es estrictamente mayor que al menos k de los rangos x, y por lo que es estrictamente mayor que el k-ésimo rango x más pequeño. Este argumento es válido para cualquier k.

Dejar que n sea el número de miembros (que es el número de valores i). Entonces n debe ser ya sea impar o par. Si n es impar, entonces dejar que k=(n+1)/2, y el k-ésimo rango más pequeño será la mediana. Por lo tanto med(x) < med(y). Si n es par, entonces cuando k=n/2, el k-ésimo rango x más pequeño será estrictamente menor que el k-ésimo rango y más pequeño, y también el (k+1)ésimo rango x más pequeño será estrictamente menor que el (k+1)ésimo rango y más pequeño. Por lo que el promedio de los dos rangos x será menor que el promedio de los dos rangos y. Por lo tanto, med(x) < med(y). Por lo que en ambos casos, la mediana de rangos x es estrictamente menor que la mediana de rangos y. Por lo que si el orden total es definido ordenando las acciones por rango de mediana, entonces x precederá a y en el orden total. QED

Teorema de Ejemplo 3: Si un "período de chismes" es la cantidad de tiempo para que los eventos existentes se propaguen a través de la sincronización a todos los miembros, entonces:

después de 1 período de chismes: todos los miembros han recibido los eventos

después de 2 períodos de chismes: todos los miembros están de acuerdo en el orden de esos eventos

después de 3 períodos de chismes: todos los miembros conocen que ha sido llegado a un acuerdo

después de 4 períodos de chismes: todos los miembros obtienen firmas digitales de todos los otros miembros, respaldando este orden de consenso.

Prueba: Dejar que S0 sea el conjunto de los eventos que han sido creados y/o definidos en un tiempo T0 dado. Si cada miembro finalmente se sincronizará con cada otro miembro infinitamente a menudo, entonces con probabilidad 1 finalmente habrá un tiempo T1 en el cual los eventos en S0 se han extendido a cada miembro, de tal manera que cada miembro conoce todos los eventos. Ese es el final del primer período de chismes. Dejar que S1 sea el conjunto de eventos que existen en el tiempo T1 y que aún no existían en T0. Entonces con probabilidad 1 finalmente habrá un tiempo T2 en el cual cada miembro ha recibido cada evento en el conjunto S1, que son los que existían en el tiempo T1. Ese es el final del segundo período de chismes. De manera similar, T3 es cuando todos los eventos en S2, los existentes en T2 pero no antes de T1, se han extendido a todos los miembros. Nótese que cada período de chismes finalmente finaliza con probabilidad 1. En promedio, cada uno durará tanto como sea necesario para realizar sincronizaciones log2(n), si hay n miembros.

Para el tiempo T1, cada miembro habrá recibido cada evento en S0.

Para el tiempo T2, un miembro dado Alice habrá recibido un registro de cada uno de los otros miembros recibiendo cada evento en S0. Por lo tanto Alice puede calcular el rango para cada acción en S0 para cada miembro (que es el orden en el cual ese miembro recibió esa acción), y luego ordenar los eventos por la mediana de los rangos. El orden total resultante no cambia, para los eventos en S0. Esto se debe a que el orden resultante es una función del orden en el cual cada miembro recibió por primera vez una indicación de cada uno de esos eventos, que no cambia. Es posible, que el orden calculado de Alice tendrá algunos eventos S1 intercalados entre los eventos de S0. Esos eventos S1 aún pueden cambiar donde caen dentro de la secuencia de eventos S0. Pero el orden relativo de eventos en S0 no cambiará.

Para el tiempo T3, Alice habrá aprendido un orden total sobre la unión de S0 y S1, y el orden relativo de los eventos en esa unión no cambiará. Adicionalmente, ella puede encontrar dentro de esta secuencia el evento más temprano de S1, y puede concluir que la secuencia de los eventos anteriores a S1 no cambiará, ni siquiera por la inserción de nuevos eventos fuera de S0. Por lo tanto, para el tiempo T3, Alice puede determinar que ha sido logrado un consenso para el orden de los eventos en la historia antes del primer evento S1. Ella puede firmar digitalmente un direccionamiento del estado (por ejemplo, como capturado por una variable de estado de base de datos definida por Alice) que resulta de estos eventos que se producen en este orden, y enviar la firma como parte del próximo evento que ella crea y/o define.

Para el tiempo T4, Alice habrá recibido firmas similares de los otros miembros. En ese momento ella simplemente puede mantener esa lista de firmas junto con el estado del que atestiguan, y ella puede descartar los eventos que ella ha almacenado antes del primer evento S1. QED

Los sistemas descritos en este documento describen una base de datos distribuida que logra un consenso de manera rápida y segura. Este puede ser un bloque de construcción útil para muchas aplicaciones. Por ejemplo, si las transacciones describen una transferencia de criptomoneda de una cartera de criptomonedas a otra, y si el estado es simplemente una declaración de la cantidad actual en cada cartera, entonces este sistema constituirá un sistema de criptomoneda que evita la costosa prueba de trabajo en sistemas existentes. La aplicación automática de reglas permite que esto agregue características que no son comunes en las criptomonedas actuales. Por ejemplo, las monedas perdidas pueden ser recuperadas, para evitar la deflación, haciendo cumplir una regla de que si una cartera no envía ni recibe criptomoneda durante un cierto período de tiempo, entonces esa cartera es eliminada, y su valor es distribuido a las otras carteras existentes, proporcional a la cantidad que contienen actualmente. De esa forma, el suministro de moneda no crecerá ni se encogerá, incluso si se pierde la clave privada para una cartera.

Otro ejemplo es un juego distribuido, que actúa similar a un juego de Multijugador Masivo en Línea (MMO) que es jugado en un servidor, pero logra eso sin usar un servidor central. El consenso puede ser logrado sin que ningún servidor central esté en control.

Otro ejemplo es un sistema para redes sociales que se construye en la parte superior de tal base de datos. Debido a que las transacciones son firmadas digitalmente, y los miembros reciben información sobre los otros miembros, esto proporciona ventajas de seguridad y conveniencia sobre sistemas actuales. Por ejemplo, puede ser implementado un sistema de correo electrónico con fuertes políticas anti-correo basura, debido a que los correos electrónicos no pueden tener direcciones de retorno falsificadas. Tal sistema también podría convertirse en un sistema social unificado, combinando en una única base de datos distribuida las funciones que se hacen actualmente por correo electrónico, tuits, textos, foros, wikis, y/u otras redes sociales.

Otras aplicaciones pueden incluir funciones criptográficas más sofisticadas, tales como firmas digitales grupales, en las cuales el grupo en conjunto coopera para firmar un contrato o documento. Ésta, y otras formas de cálculo multiparte, pueden implementarse de manera útil usando tal sistema de consenso distribuido.

Otro ejemplo es un sistema de libro mayor público. Cualquiera puede pagar para almacenar alguna información en el sistema, pagando una pequeña cantidad de criptomoneda (o moneda del mundo real) por byte por año para almacenar información en el sistema. Estos fondos entonces pueden ser distribuidos automáticamente a los miembros que almacenan esos datos, y a los miembros que se sincronizan repetidamente para trabajar para lograr el consenso. Puede transferir automáticamente a los miembros una pequeña cantidad de la criptomoneda por cada vez que se sincronizan.

Estos ejemplos muestran que la base de datos de consenso distribuida es útil como un componente de muchas aplicaciones. Debido a que la base de datos no usa una prueba de trabajo costosa, posiblemente usando una prueba de participación más barata en su lugar, la base de datos puede ejecutarse con un nodo completo que se ejecuta en ordenadores más pequeños o incluso en dispositivos móviles e incorporados.

Aunque se describió anteriormente como un evento que contiene un direccionamiento de dos eventos anteriores (un direccionamiento propio y un direccionamiento foráneo), en otras realizaciones, un miembro puede sincronizarse con otros dos miembros para crear y/o definir un evento que contiene direccionamientos de tres eventos anteriores (un direccionamiento propio y dos direccionamientos foráneos). En aún otras realizaciones, puede ser incluido cualquier número de direccionamientos de eventos de eventos anteriores de cualquier número de miembros dentro de un evento. En algunas realizaciones, diferentes eventos pueden incluir diferentes números de direccionamientos de eventos anteriores. Por ejemplo, un primer evento puede incluir dos direccionamientos de eventos y un segundo evento puede incluir tres direccionamientos de eventos.

Aunque los eventos se describen anteriormente como que incluyen direccionamientos (o valores de direccionamiento criptográficos) de eventos anteriores, en otras realizaciones, puede ser creado y/o definido un evento para incluir un puntero, un identificador, y/o cualquier otra referencia adecuada a los eventos anteriores. Por ejemplo, un evento puede ser creado y/o definido para incluir un número de serie asociado con y usado para identificar un evento anterior, vinculando de este modo los eventos. En algunas realizaciones, tal número de serie puede incluir, por ejemplo, un identificador (por ejemplo, dirección de control de acceso al medio (MAC), dirección de Protocolo de Internet (IP), una dirección asignada, y/o similares) asociado con el miembro que creó y/o definido el evento y un orden del evento definido por ese miembro. Por ejemplo, un miembro que tiene un identificador de 10 y el evento es el 15a evento creado y/o definido por ese miembro puede asignar un identificador de 1015 a ese evento. En otras realizaciones, puede ser usado cualquier otro formato adecuado para asignar identificadores para eventos.

En otras realizaciones, los eventos pueden contener direccionamientos criptográficos completos, pero solo son transmitidas porciones de esos direccionamientos durante la sincronización. Por ejemplo, si Alice envía a Bob un evento que contiene un direccionamiento H, y J son los primeros 3 bytes de H, y Alice determina que de los eventos y direccionamientos que ella ha almacenado, H es el único direccionamiento que inicia con J, entonces ella puede enviar J en vez de H durante la sincronización. Si Bob luego determina que él tiene otro direccionamiento que inicia con J, entonces él puede responder a Alice para solicitar el H completo. De esa forma, los direccionamientos pueden ser comprimidos durante la transmisión.

Aunque los sistemas de ejemplo mostrados y descritos anteriormente se describen con referencia a otros sistemas, en otras realizaciones puede ser implementada cualquier combinación de los sistemas de ejemplo y sus funcionalidades asociadas para crear y/o definir una base de datos distribuida. Por ejemplo, el Sistema de Ejemplo 1, Sistema de Ejemplo 2, y Sistema de Ejemplo 3 pueden ser combinados para crear y/o definir una base de datos distribuida. Para otro ejemplo, en algunas realizaciones, el Sistema de Ejemplo 10 puede ser implementado con el Sistema de Ejemplo 1 pero sin el Sistema de Ejemplo 9. Para aún otro ejemplo, el Sistema de Ejemplo 7 puede ser combinado e implementado con el Sistema de Ejemplo 6. En otras realizaciones, puede ser implementada cualquier otra combinación adecuada de los sistemas de ejemplo.

Aunque se describió anteriormente como intercambio de eventos para obtener convergencia, en otras realizaciones, las instancias de bases de datos distribuidas pueden intercambiar valores y/o vectores de valores para obtener convergencia como se describe con respecto a las figuras 8-13. Específicamente, por ejemplo, la figura 8 ilustra un flujo de comunicación entre un primer dispositivo 400 de cálculo de un sistema de base de datos distribuida (por ejemplo, sistema 100 de base de datos distribuida) y un segundo dispositivo 500 de cálculo del sistema de base de datos distribuida (por ejemplo, sistema 100 de base de datos distribuida), de acuerdo con una realización. En algunas realizaciones, los dispositivos 400, 500 de cálculo pueden ser estructural y/o funcionalmente similares al dispositivo 200 de cálculo mostrado en la figura 2. En algunas realizaciones, el dispositivo 400 de cálculo y dispositivo 500 de cálculo se comunican entre sí de una manera similar a cómo los dispositivos 110, 120, 130, 140 de cálculo se comunican entre sí dentro del sistema 100 de base de datos distribuida, mostrado y descrito con respecto a la figura 1.

Similar al dispositivo 200 de cálculo, descrito con respecto a la figura 2, los dispositivos 400, 500 de cálculo pueden cada uno definir inicialmente un vector de valores para un parámetro, actualizar el vector de valores, seleccionar un valor para el parámetro con base en el vector definido y/o actualizado de valores para el parámetro, y almacenar (1) el vector definido y/o actualizado de valores para el parámetro y/o (2) el valor seleccionado para el parámetro con base en el vector definido y/o actualizado de valores para el parámetro. Cada uno de los dispositivos 400, 500 de cálculo puede definir inicialmente un vector de valores para un parámetro de cualquier número de formas. Por ejemplo, cada

uno de los dispositivos 400, 500 de cálculo puede definir inicialmente un vector de valores para un parámetro estableciendo cada valor del vector de valores para que sea igual al valor almacenado inicialmente en las instancias 403, 503 de bases de datos distribuidas, respectivamente. Para otro ejemplo, cada uno de los dispositivos 400, 500 de cálculo puede definir inicialmente un vector de valores para un parámetro estableciendo cada valor del vector de valores para que sea igual a un valor aleatorio. Cómo va a ser definido inicialmente el vector de valores para un parámetro puede ser seleccionado, por ejemplo, por un administrador de un sistema de base de datos distribuida al cual pertenecen los dispositivos 400, 500 de cálculo, o individual o colectivamente por los usuarios de los dispositivos de cálculo (por ejemplo, los dispositivos 400, 500 de cálculo) del sistema de base de datos distribuida.

Los dispositivos 400, 500 de cálculo también pueden almacenar cada uno el vector de valores para el parámetro y/o el valor seleccionado para el parámetro en instancias 403, 503 de bases de datos distribuidas, respectivamente. Cada una de las instancias 403, 503 de bases de datos distribuidas puede ser implementada en una memoria (no se muestra en la figura 8) similar a la memoria 220, mostrada en la figura 2.

En la etapa 1, el dispositivo 400 de cálculo solicita del dispositivo 500 de cálculo un valor para un parámetro almacenado en la instancia 503 de base de datos distribuida del dispositivo 500 de cálculo (por ejemplo, un valor almacenado en un campo específico de la instancia 503 de base de datos distribuida). En algunas realizaciones, el dispositivo 500 de cálculo puede ser elegido por el dispositivo 400 de cálculo de un conjunto de dispositivos de cálculo que pertenecen a un sistema de base de datos distribuida. El dispositivo 500 de cálculo puede ser elegido de manera aleatoria, elegido con base en una relación con el dispositivo 400 de cálculo, con base en la proximidad al dispositivo 400 de cálculo, elegido con base en una lista ordenada asociada con el dispositivo 400 de cálculo, y/o similar. En algunas realizaciones, debido a que el dispositivo 500 de cálculo puede ser elegido por el dispositivo 400 de cálculo del conjunto de dispositivos de cálculo que pertenecen al sistema de base de datos distribuida, el dispositivo 400 de cálculo puede seleccionar el dispositivo 500 de cálculo múltiples veces seguidas o puede no seleccionar el dispositivo 500 de cálculo por algún tiempo. En otras realizaciones, puede ser almacenada una indicación de los dispositivos de cálculo seleccionados previamente en el dispositivo 400 de cálculo. En tales realizaciones, el dispositivo 400 de cálculo puede esperar un número predeterminado de selecciones antes de poder seleccionar de nuevo el dispositivo 500 de cálculo. Como se explicó anteriormente, la instancia 503 de base de datos distribuida puede ser implementada en una memoria de dispositivo 500 de cálculo.

En algunas realizaciones, la solicitud del dispositivo 400 de cálculo puede ser una señal enviada por un módulo de comunicación del dispositivo 400 de cálculo (no se muestra en la figura 8). Esta señal puede ser portada por una red, tal como red 105 (se muestra en la figura 1), y recibida por un módulo de comunicación del dispositivo 500 de cálculo. En algunas realizaciones, cada uno de los módulos de comunicación de dispositivos 400, 500 de cálculo puede ser implementado dentro de un procesador o memoria. Por ejemplo, los módulos de comunicación de dispositivos 400, 500 de cálculo pueden ser similares al módulo 212 de comunicación mostrado en la figura 2.

Después de recibir, desde el dispositivo 400 de cálculo, la solicitud para el valor del parámetro almacenado en la instancia 503 de base de datos distribuida, el dispositivo 500 de cálculo envía el valor del parámetro almacenado en la instancia 503 de base de datos distribuida al dispositivo 400 de cálculo en la etapa 2. En algunas realizaciones, el dispositivo 500 de cálculo puede recuperar el valor del parámetro de la memoria, y enviar el valor como una señal a través de un módulo de comunicación del dispositivo 500 de cálculo (no se muestra en la figura 8). En alguna instancia si la instancia 503 de base de datos distribuida no incluye ya un valor para el parámetro (por ejemplo, la transacción aún no ha sido definida en la instancia 503 de base de datos distribuida), la instancia 503 de base de datos distribuida puede solicitar un valor para el parámetro del dispositivo 403 de cálculo (si no se ha proporcionado ya en la etapa 1) y almacenar ese valor para el parámetro en la instancia 503 de base de datos distribuida. En algunas realizaciones, el dispositivo 400 de cálculo usará entonces este valor como el valor para el parámetro en la instancia 503 de base de datos distribuida.

En la etapa 3, el dispositivo 400 de cálculo envía al dispositivo 500 de cálculo un valor para un parámetro almacenado en la instancia 403 de base de datos distribuida. En otras realizaciones, el valor para el parámetro almacenado en la instancia 403 de base de datos distribuida (etapa 1) y la solicitud para el valor para el mismo parámetro almacenado en la instancia 503 de base de datos distribuida (etapa 3) pueden ser enviados como una única señal. En otras realizaciones, el valor para el parámetro almacenado en la instancia 403 de base de datos distribuida puede ser enviado en una señal diferente de la señal para la solicitud para el valor para el parámetro almacenado en la instancia 503 de base de datos distribuida. En realizaciones donde el valor para el parámetro almacenado en la instancia 403 de base de datos distribuida es enviado en una señal diferente de la señal para la solicitud para el valor para el parámetro almacenado en la instancia 503 de base de datos distribuida, el valor para el parámetro almacenado en la instancia 403 de base de datos distribuida, las dos señales pueden ser enviadas en cualquier orden. En otras palabras, cualquier señal puede ser la enviada antes que la otra.

Después de que el dispositivo 400 de cálculo recibe el valor del parámetro enviado desde el dispositivo 500 de cálculo y/o el dispositivo 500 de cálculo recibe el valor para el parámetro enviado desde el dispositivo 400 de cálculo, en algunas realizaciones, el dispositivo 400 de cálculo y/o el dispositivo 500 de cálculo puede actualizar el vector de valores almacenados en la instancia 403 de base de datos distribuida y/o el vector de valores almacenados en la instancia 503 de base de datos distribuida, respectivamente. Por ejemplo, los dispositivos 400, 500 de cálculo pueden actualizar el vector de valores almacenados en instancias 403, 503 de bases de datos distribuidas para incluir el valor

del parámetro recibido por los dispositivos 400, 500 de cálculo, respectivamente. Los dispositivos 400, 500 de cálculo también pueden actualizar el valor del parámetro almacenado en la instancia 403 de base de datos distribuida y/o el valor del parámetro almacenado en la instancia 503 de base de datos distribuida, respectivamente, con base en el vector actualizado de valores almacenados en la instancia 403 de base de datos distribuida y/o el vector actualizado de valores almacenados en la instancia 503 de base de datos distribuida, respectivamente.

Aunque las etapas están etiquetadas 1, 2, y 3 en la figura 8 y en la discusión anterior, debe entenderse que las etapas 1, 2, y 3 pueden ser realizadas en cualquier orden. Por ejemplo, la etapa 3 puede ser realizada antes que las etapas 1 y 2. Adicionalmente, la comunicación entre el dispositivo 400 y 500 de cálculo no se limita a las etapas 1, 2, y 3 mostradas en la figura 3, como se describe en detalle en este documento. Además, después de que se completan las etapas 1, 2 y 3, el dispositivo 400 de cálculo puede seleccionar otro dispositivo de cálculo del conjunto de dispositivos de cálculo dentro del sistema de base de datos distribuida con el cual intercambiar valores (similar a las etapas 1, 2 y 3).

En algunas realizaciones, los datos comunicados entre los dispositivos 400, 500 de cálculo pueden incluir datos comprimidos, datos encriptados, firmas digitales, sumas de verificación criptográficas, y/o similares. Adicionalmente, cada uno de los dispositivos 400, 500 de cálculo puede enviar datos al otro dispositivo de cálculo para acusar recibo de los datos enviados previamente por el otro dispositivo. Cada uno de los dispositivos 400, 500 de cálculo también puede ignorar los datos que han sido enviados repetidamente por el otro dispositivo.

Cada uno de los dispositivos 400, 500 de cálculo puede definir inicialmente un vector de valores para un parámetro, y almacenar este vector de valores para un parámetro en instancias 403, 503 de bases de datos distribuidas, respectivamente. Las figuras 9a-9c ilustran ejemplos de vectores de valores para un parámetro. Un vector puede ser cualquier conjunto de valores para un parámetro (por ejemplo, un arreglo unidimensional de valores para un parámetro, un arreglo de valores cada uno teniendo múltiples partes, etc.). En las figuras 9a-9c se proporcionan tres ejemplos de vectores con propósitos de ilustración. Como se muestra, cada uno de los vectores 410, 420, 430 tiene cinco valores para un parámetro particular. Sin embargo, debe entenderse que un vector de valores puede tener cualquier número de valores. En algunas instancias, el número de valores incluidos en un vector de valores puede ser establecido por usuario, situación, aleatoriamente, etc.

Un parámetro puede ser cualquier objeto de datos capaz de tomar diferentes valores. Por ejemplo, un parámetro puede ser un voto binario, en el cual el valor de voto puede ser ya sea "SÍ" o "NO" (o un "1" o "0" binario). Como se muestra en la figura 9a, el vector de valores 410 es un vector que tiene cinco votos binarios, siendo los valores 411, 412, 413, 414, 415 "SÍ", "NO", "NO", "SÍ", y "SÍ", respectivamente. Para otro ejemplo, un parámetro puede ser un conjunto de elementos de datos. La figura 9b muestra un ejemplo donde el parámetro es un conjunto de letras de alfabeto. Como se muestra, el vector de valores 420 tiene cinco conjuntos de cuatro letras de alfabeto, siendo los valores 421, 422, 423, 424, 425 {A, B, C, D}, {A, B, C, E}, {A, B, C, F}, {A, B, F, G}, y {A, B, G, H}, respectivamente. Para aún otro ejemplo, un parámetro puede ser un conjunto clasificado y/o ordenado de elementos de datos. La figura 9c muestra un ejemplo donde el parámetro es un conjunto clasificado de personas. Como se muestra, el vector de valores 430 tiene cinco conjuntos clasificados de seis personas, siendo los valores 431, 432, 433, 434, 435

(1. Alice, 2. Bob, 3. Carol, 4. Dave, 5. Ed, 6. Frank),

(1. Bob, 2. Alice, 3. Carol, 4. Dave, 5. Ed, 6. Frank),

(1. Bob, 2. Alice, 3. Carol, 4. Dave, 5. Frank, 6. Ed),

(1. Alice, 2. Bob, 3. Carol, 4. Ed, 5. Dave, 6. Frank), y

(1. Alice, 2. Bob, 3. Ed, 4. Carol, 5. Dave, 6. Frank),

respectivamente.

Después de definir un vector de valores para un parámetro, cada uno de los dispositivos 400, 500 de cálculo puede seleccionar un valor para el parámetro con base en el vector de valores para el parámetro. Esta selección puede ser realizada de acuerdo con cualquier método y/o proceso (por ejemplo, una regla o un conjunto de reglas). Por ejemplo, la selección puede ser realizada de acuerdo con "reglas de mayoría", donde el valor para el parámetro es seleccionado para que sea el valor que aparece en más del 50% de los valores incluidos en el vector. Para ilustrar, el vector de valores 410 (se muestra en la figura 9a) incluye tres valores "SÍ" y dos valores "NO". Bajo las "reglas de mayoría", el valor seleccionado para el parámetro con base en el vector de valores sería "SÍ", debido a que "SÍ" aparece en más del 50% de valores 411, 412, 413, 414, 415 (de vector de valores 410).

Para otro ejemplo, la selección puede ser realizada de acuerdo con la "apariencia de mayoría", donde el valor para el parámetro es seleccionado para que sea un conjunto de elementos de datos, apareciendo cada elemento de datos en más del 50% de los valores incluidos en el vector. Para ilustrar el uso de la figura 9b, los elementos de datos "A", "B", y "C" aparecen en más del 50% de los valores 421, 422, 423, 424, 425 del vector de valores 420. Bajo "apariencia de mayoría", el valor seleccionado para el parámetro con base en el vector de valores sería {A, B, C} debido a que solo estos elementos de datos (es decir, "A", "B", y "C") aparecen en tres de los cinco valores del vector de valores 420.

Para aún otro ejemplo, la selección puede ser realizada de acuerdo con "rango por mediana", donde el valor para el parámetro es seleccionado para ser un conjunto clasificado de elementos de datos (por ejemplo, valores de datos distintos dentro de un valor de un vector de valores), el rango de cada elemento de datos es igual al rango de mediana de ese elemento de datos a través de todos los valores incluidos en el vector. Para ilustrar, el rango de mediana de cada elemento de datos en la figura 9c es calculado a continuación:

| | | |
|---|---|---|
| Alice: | (1, 2, 2, 1, 1); | rango de mediana = 1; |
| Bob: | (2, 1, 1, 2, 2); | rango de mediana = 2; |
| Carol: | (3, 3, 3, 3, 4); | rango de mediana = 3; |
| Dave: | (4, 4, 4, 5, 5); | rango de mediana = 4; |
| Ed: | (5, 5, 6, 4, 3); | rango de mediana = 5; |
| Frank: | (6, 6, 5, 6, 6); | rango de mediana = 6; |

De este modo, bajo "rango por mediana", el valor para el conjunto clasificado de elementos de datos calculado con base en el vector de valores 430 sería (1. Alice, 2. Bob, 3. Carol, 4. Dave, 5. Ed, 6. Frank). En algunas realizaciones, si dos o más elementos de datos tienen una misma mediana (por ejemplo, un empate), el orden puede ser determinado mediante cualquier método adecuado (por ejemplo, de manera aleatoria, primera indicación de rango, última indicación de rango, alfabéticamente y/o numéricamente, etc.).

Para un ejemplo adicional, la selección puede ser realizada de acuerdo con la "votación de Kemeny Young", donde el valor del parámetro es seleccionado para que sea un conjunto clasificado de elementos de datos, siendo el rango calculado para minimizar un valor de coste. Por ejemplo, Alice se clasifica antes que Bob en vectores de valores 431, 434, 435, para un total de tres de los cinco vectores de valores. Bob se clasifica antes que Alice en vectores de valores 432 y 433, para un total de dos de los cinco vectores de valores. El valor de coste para clasificar a Alice antes que Bob es 2/5 y el valor de coste para clasificar a Bob antes que Alice es 3/5. De este modo, el valor de coste para Alice antes que Bob es menor, y Alice será clasificada antes que Bob en "votación de Kemeny Young".

Debe entenderse que "reglas de mayoría", "apariencia de mayoría", "rango por mediana" y "votación de Kemeny Young" se discuten como ejemplos de métodos y/o procesos que pueden ser usados para seleccionar un valor para el parámetro con base en el vector de valores para el parámetro. También puede ser usado cualquier otro método y/o proceso. Por ejemplo, el valor para el parámetro puede ser seleccionado para que sea el valor que aparece en más del x% de los valores incluidos en el vector, donde x% puede ser cualquier porcentaje (es decir, no limitado al 50% como se usa en "reglas de mayoría"). El porcentaje (es decir, x%) también puede variar a través de las selecciones realizadas en diferentes momentos, por ejemplo, en relación con un valor de confianza (discutido en detalle en este documento).

En algunas realizaciones, debido a que un dispositivo de cálculo puede seleccionar aleatoriamente otros dispositivos de cálculo con los cuales intercambiar valores, un vector de valores de un dispositivo de cálculo puede incluir en cualquier momento múltiples valores de otro dispositivo de cálculo único. Por ejemplo, si un tamaño de vector es cinco, un dispositivo de cálculo puede haber seleccionado aleatoriamente otro dispositivo de cálculo dos veces dentro de las últimas cinco iteraciones de intercambio de valor. Por consiguiente, el valor almacenado en la otra instancia de base de datos distribuida del dispositivo de cálculo sería incluido dos veces en el vector de cinco valores para el dispositivo de cálculo solicitante.

Las figuras 10a-10d juntas ilustran, como un ejemplo, cómo un vector de valores puede ser actualizado cuando un dispositivo de cálculo se comunica con otro dispositivo de cálculo. Por ejemplo, el dispositivo 400 de cálculo puede definir inicialmente un vector de valores 510. En algunas realizaciones, el vector de valores 510 puede ser definido con base en un valor para un parámetro almacenado en la instancia 403 de base de datos distribuida en el dispositivo 400 de cálculo. Por ejemplo, cuando el vector de valores 510 es definido primero, cada valor del vector de valores 510 (es decir, cada uno de los valores 511, 512, 513, 514, 515) puede estar establecido para que sea igual al valor para el parámetro almacenado en la instancia 403 de base de datos distribuida. Para ilustrar, si el valor para el parámetro almacenado en la instancia 403 de base de datos distribuida, en el momento en que es definido el vector de valores 510, es "SÍ", entonces cada valor del vector de valores 510 (es decir, cada uno de los valores 511, 512, 513, 514, 515) sería establecido en "SÍ", como se muestra en la figura 10a. Cuando el dispositivo 400 de cálculo recibe un valor para el parámetro almacenado en una instancia de la base de datos distribuida de otro dispositivo de cálculo (por ejemplo, instancia 504 de base de datos distribuida de dispositivo 500 de cálculo), el dispositivo 400 de cálculo puede actualizar el vector de valores 510 para incluir el valor para el parámetro almacenado en la instancia 504 de base de datos distribuida. En algunas instancias, el vector de valores 510 puede ser actualizado de acuerdo con Primeros en Entrar, Primeros en Salir (FIFO). Por ejemplo, si el dispositivo 400 de cálculo recibe el valor 516 ("SÍ"), el dispositivo 400 de cálculo puede agregar el valor 516 al vector de valores 510 y eliminar el valor 511 del vector de valores 510, para definir el vector de valores 520, como se muestra en la figura 10b. Por ejemplo, si en un momento posterior el

dispositivo de cálculo recibe los valores 517, 518, el dispositivo 400 de cálculo puede agregar los valores 517, 518 al vector de valores 510 y eliminar el valor 512, 513, respectivamente, del vector de valores 510, para definir el vector de valores 530, 540, respectivamente. En otras instancias, el vector de valores 510 puede ser actualizado de acuerdo con esquemas aparte de Primeros en Entrar, Primeros en Salir, tales como, Últimos en Entrar, Primeros en Salir (LIFO).

Después de que el dispositivo 400 de cálculo actualiza el vector de valores 510 para definir vectores de valores 520, 530, y/o 540, el dispositivo 400 de cálculo puede seleccionar un valor para el parámetro con base en el vector de valores 520, 530, y/o 540. Esta selección puede ser realizada de acuerdo con cualquier método y/o proceso (por ejemplo, una regla o un conjunto de reglas), como se discutió anteriormente con respecto a las figuras 9a-9c.

En algunas instancias, los dispositivos 400, 500 de cálculo pueden pertenecer a un sistema de base de datos distribuida que almacena información relacionada con transacciones que involucran instrumentos financieros. Por ejemplo, cada uno de los dispositivos 400, 500 de cálculo puede almacenar un voto binario (un ejemplo de un "valor") sobre si una acción particular está disponible para compra (un ejemplo de un "parámetro"). Por ejemplo, la instancia 403 de base de datos distribuida del dispositivo 400 de cálculo puede almacenar un valor de "SÍ", lo cual indica que la acción particular está verdaderamente disponible para compra. La instancia 503 de base de datos distribuida del dispositivo 500 de cálculo, por otro lado, puede almacenar un valor de "NO", lo cual indica que la acción particular no está disponible para compra. En algunas instancias, el dispositivo 400 de cálculo puede definir inicialmente un vector de votos binarios con base en el voto binario almacenado en la instancia 403 de base de datos distribuida. Por ejemplo, el dispositivo 400 de cálculo puede establecer cada voto binario dentro del vector de votos binarios para que sea igual al voto binario almacenado en la instancia 403 de base de datos distribuida. En este caso, el dispositivo 400 de cálculo puede definir un vector de votos binarios similar al vector de valores 510. En algún momento posterior, el dispositivo 400 de cálculo puede comunicarse con el dispositivo 500 de cálculo, solicitando al dispositivo 500 de cálculo que envíe su voto binario sobre si la acción particular está disponible para compra. Una vez que el dispositivo 400 de cálculo recibe el voto binario del dispositivo 500 de cálculo (en este ejemplo, "NO", que indica que la acción particular no está disponible para compra), el dispositivo 400 de cálculo puede actualizar su vector de votos binarios. Por ejemplo, el vector actualizado de votos binarios puede ser similar al vector de valores 520. Esto puede producirse indefinidamente, hasta que un valor de confianza cumpla con un criterio predeterminado (descrito con más detalle en este documento), periódicamente, y/o similar.

La figura 11 muestra un diagrama 10 de flujo que ilustra las etapas realizadas por el dispositivo 110 de cálculo dentro del sistema 100 de base de datos distribuida, de acuerdo con una realización. En la etapa 11, el dispositivo 110 de cálculo define un vector de valores para un parámetro con base en un valor del parámetro almacenado en la instancia 113 de base de datos distribuida. En algunas realizaciones, el dispositivo 110 de cálculo puede definir un vector de valores para el parámetro con base en un valor para un parámetro almacenado en la instancia 113 de base de datos distribuida. En la etapa 12, el dispositivo 110 de cálculo elige otro dispositivo de cálculo dentro del sistema 110 de base de datos distribuida y solicita del dispositivo de cálculo elegido un valor para el parámetro almacenado en la instancia de base de datos distribuida del dispositivo de cálculo elegido. Por ejemplo, el dispositivo 110 de cálculo puede elegir aleatoriamente el dispositivo 120 de cálculo de entre los dispositivos 120, 130, 140 de cálculo, y solicitar del dispositivo 120 de cálculo un valor para el parámetro almacenado en la instancia 123 de base de datos distribuida. En la etapa 13, el dispositivo 110 de cálculo (1) recibe, desde el dispositivo de cálculo elegido (por ejemplo, el dispositivo 120 de cálculo), el valor para el parámetro almacenado en la instancia de base de datos distribuida del dispositivo de cálculo elegido (por ejemplo, la instancia 123 de base de datos distribuida) y (2) envía, al dispositivo de cálculo elegido (por ejemplo, el dispositivo 120 de cálculo), un valor para el parámetro almacenado en la instancia 113 de base de datos distribuida. En la etapa 14, el dispositivo 110 de cálculo almacena el valor para el parámetro recibido desde el dispositivo de cálculo elegido (por ejemplo, el dispositivo 120 de cálculo) en el vector de valores para el parámetro. En la etapa 15, el dispositivo 110 de cálculo selecciona un valor para el parámetro con base en el vector de valores para el parámetro. Esta selección puede ser realizada de acuerdo con cualquier método y/o proceso (por ejemplo, una regla o un conjunto de reglas), como se discutió anteriormente con respecto a las figuras 9a-9c. En algunas realizaciones, el dispositivo 110 de cálculo puede repetir la selección de un valor para el parámetro en momentos diferentes. El dispositivo 110 de cálculo también puede funcionar en ciclos repetidamente a través de las etapas 12 hasta 14 entre cada selección de un valor para el parámetro.

En algunas instancias, el sistema 100 de base de datos distribuida puede almacenar información relacionada con las transacciones dentro de un Juego Multijugador Masivo (MMG). Por ejemplo, cada dispositivo de cálculo que pertenece al sistema 100 de base de datos distribuida puede almacenar un conjunto clasificado de jugadores (un ejemplo de un "valor") en el orden en el cual fue poseído un ítem particular (un ejemplo de un "parámetro"). Por ejemplo, la instancia 114 de base de datos distribuida del dispositivo 110 de cálculo puede almacenar un conjunto clasificado de jugadores (1. Alice, 2. Bob, 3. Carol, 4. Dave, 5. Ed, 6. Frank), similar al valor 431, lo que indica que la posesión del ítem en particular comenzó con Alice, luego fue pasado a Bob, luego fue pasado a Carol, luego fue pasado a Dave, luego fue pasado a Ed, y finalmente fue pasado a Frank. La instancia 124 de base de datos distribuida del dispositivo 120 de cálculo puede almacenar un valor de un conjunto clasificado de jugadores similar al valor 432: (1. Bob, 2. Alice, 3. Carol, 4. Dave, 5. Ed, 6. Frank); la instancia 134 de base de datos distribuida del dispositivo 130 de cálculo puede almacenar un valor de un conjunto clasificado de jugadores similar al valor 433: (1. Bob, 2. Alice, 3. Carol, 4. Dave, 5. Frank, 6. Ed); la instancia 144 de base de datos distribuida del dispositivo 140 de cálculo puede almacenar un valor de un conjunto clasificado de jugadores similar al valor 434: (1. Alice, 2. Bob, 3. Carol, 4. Ed, 5. Dave, 6. Frank); la

instancia de base de datos distribuida de un quinto dispositivo de cálculo (no se muestra en la figura 1) puede almacenar un valor de un conjunto clasificado de jugadores similar al valor 435: (1. Alice, 2. Bob, 3. Ed, 4. Carol, 5. Dave, 6. Frank).

Después de que el dispositivo 110 de cálculo define un vector de conjunto clasificado de jugadores, el dispositivo de cálculo puede recibir valores de conjuntos clasificados de jugadores de los otros dispositivos de cálculo del sistema 100 de base de datos distribuida. Por ejemplo, el dispositivo 110 de cálculo puede recibir (1. Bob, 2. Alice, 3. Carol, 4. Dave, 5. Ed, 6. Frank) del dispositivo 120 de cálculo; (1. Bob, 2. Alice, 3. Carol, 4. Dave, 5. Frank, 6. Ed) del dispositivo 130 de cálculo; (1. Alice, 2. Bob, 3. Carol, 4. Ed, 5. Dave, 6. Frank) del dispositivo 140 de cálculo; y (1. Alice, 2. Bob, 3. Ed, 4. Carol, 5. Dave, 6. Frank) del quinto dispositivo de cálculo (no se muestra en la figura 1). Cuando el dispositivo 110 de cálculo recibe valores de conjuntos clasificados de jugadores de los otros dispositivos de cálculo, el dispositivo 110 de cálculo puede actualizar su vector de conjuntos clasificados de jugadores para incluir los valores de conjuntos clasificados de jugadores recibidos de los otros dispositivos de cálculo. Por ejemplo, el vector de conjuntos clasificados de jugadores almacenados en la instancia 114 de base de datos distribuida del dispositivo 110 de cálculo, después de recibir los valores de los conjuntos clasificados enumerados anteriormente, puede ser actualizado para ser similar al vector de valores 430. Después de que el vector de conjuntos clasificados de jugadores ha sido actualizado para ser similar al vector de valores 430, el dispositivo 110 de cálculo puede seleccionar un conjunto clasificado de jugadores con base en el vector de conjuntos clasificados de jugadores. Por ejemplo, la selección puede ser realizada de acuerdo con "rango por mediana", como se discutió anteriormente con respecto a las figuras 9a-9c. Bajo "rango por mediana", el dispositivo 110 de cálculo seleccionaría (1. Alice, 2. Bob, 3. Carol, 4. Dave, 5. Ed, 6. Frank) con base en el vector de conjuntos clasificados de jugadores similar al vector de valores 430.

En algunas instancias, el dispositivo 110 de cálculo no recibe el valor total de otro dispositivo de cálculo. En algunas instancias, el dispositivo 110 de cálculo puede recibir un identificador asociado con porciones del valor total (también denominado como el valor compuesto), tal como un valor de direccionamiento criptográfico, en vez de las porciones mismas. Para ilustrar, el dispositivo 110 de cálculo, en algunas instancias, no recibe (1. Alice, 2. Bob, 3. Carol, 4. Ed, 5. Dave, 6. Frank), el valor 434 completo, del dispositivo 140 de cálculo, pero sólo recibe (4. Ed, 5. Dave, 6. Frank) del dispositivo 140 de cálculo. En otras palabras, el dispositivo 110 de cálculo no recibe del dispositivo 140 de cálculo (1. Alice, 2. Bob, 3 Carol), ciertas porciones del valor 434. En vez de esto, el dispositivo 110 de cálculo puede recibir del dispositivo 140 de cálculo un valor de direccionamiento criptográfico asociado con estas porciones del valor 434, es decir, (1. Alice, 2. Bob, 3. Carol).

Un valor de direccionamiento criptográfico representa de manera única las porciones del valor con las que está asociado. Por ejemplo, un direccionamiento criptográfico que representa (1. Alice, 2. Bob, 3. Carol) será diferente de direccionamientos criptográficos que representan:

(1. Alice);

(2. Bob);

(3. Carol);

(1. Alice, 2. Bob);

(2. Bob, 3. Carol);

(1. Bob, 2. Alice, 3. Carol);

(1. Carol, 2. Bob, 3. Alice);

etc.

Después de que el dispositivo 110 de cálculo recibe del dispositivo 140 de cálculo un valor de direccionamiento criptográfico asociado con ciertas porciones del valor 434, el dispositivo 110 de cálculo puede (1) generar un valor de direccionamiento criptográfico usando las mismas porciones del valor 431 almacenado en la instancia 113 de base de datos distribuida y (2) comparar el valor de direccionamiento criptográfico generado con el valor de direccionamiento criptográfico recibido.

Por ejemplo, el dispositivo 110 de cálculo puede recibir del dispositivo 140 de cálculo un valor de direccionamiento criptográfico asociado con las ciertas porciones del valor 434, indicado en cursiva: (1. Alice, 2. Bob, 3. Carol, 4. Ed, 5. Dave, 6. Frank). El dispositivo de cálculo puede entonces generar un valor de direccionamiento criptográfico usando las mismas porciones del valor 431 (almacenado en la instancia 113 de base de datos distribuida), indicado en cursiva: (1. Alice, 2. Bob, 3. Carol, 4. Dave, 5. Ed, 6. Frank). Debido a que las porciones en cursiva de valor 434 y las porciones en cursiva de valor 431 son idénticas, el valor de direccionamiento criptográfico recibido (asociado con las porciones en cursiva de valor 434) también será idéntico al valor de direccionamiento criptográfico generado (asociado con porciones en cursiva de valor 431).

Comparando el valor de direccionamiento criptográfico generado con el valor de direccionamiento criptográfico recibido, el dispositivo 110 de cálculo puede determinar si solicitar del dispositivo 140 de cálculo las porciones reales asociadas con el valor de direccionamiento criptográfico recibido. Si el valor de direccionamiento criptográfico generado es idéntico al valor de direccionamiento criptográfico recibido, el dispositivo 110 de cálculo puede determinar que una copia idéntica a las porciones reales asociadas con el valor de direccionamiento criptográfico recibido ya está almacenada en la instancia 113 de base de datos distribuida, y por lo tanto las porciones reales asociadas con el valor de direccionamiento criptográfico recibido no son necesarias desde el dispositivo 140 de cálculo. Por otro lado, si el valor de direccionamiento criptográfico generado no es idéntico al valor de direccionamiento criptográfico recibido, el dispositivo 110 de cálculo puede solicitar las porciones reales asociadas con el valor de direccionamiento criptográfico recibido del dispositivo 140 de cálculo.

Aunque los valores de direccionamiento criptográficos discutidos anteriormente están asociados con porciones de valores únicos, debe entenderse que un valor de direccionamiento criptográfico puede ser asociado con un valor único completo y/o valores múltiples. Por ejemplo, en algunas realizaciones, un dispositivo de cálculo (por ejemplo, el dispositivo 140 de cálculo) puede almacenar un conjunto de valores en su instancia de base de datos distribuida (por ejemplo, la instancia 144 de base de datos distribuida). En tales realizaciones, después de un período de tiempo predeterminado desde que ha sido actualizado un valor en la instancia de base de datos, después de que un valor de confianza (discutido con respecto a la figura 13) para el valor cumple con un criterio predeterminado (por ejemplo, alcance un umbral predeterminado), después de una cantidad especificada de tiempo desde que se originó la transacción y/o con base en cualquier otro factor adecuado, ese valor puede ser incluido en un valor de direccionamiento criptográfico con otros valores cuando son solicitados datos de y enviados a otra instancia de base de datos. Esto reduce el número de valores específicos que son enviados entre instancias de bases de datos.

En algunas instancias, por ejemplo, el conjunto de valores en la base de datos puede incluir un primer conjunto de valores, incluyendo las transacciones entre el año 2000 y el año 2010; un segundo conjunto de valores, incluyendo transacciones entre el año 2010 y el año 2013; un tercer conjunto de valores, incluyendo transacciones entre el año 2013 y el año 2014; y un cuarto conjunto de valores, incluyendo transacciones entre 2014 y el presente. Usando este ejemplo, si el dispositivo 110 de cálculo solicita del dispositivo 140 de cálculo datos almacenados en la instancia 144 de base de datos distribuida del dispositivo 140 de cálculo, en algunas realizaciones, el dispositivo 140 de cálculo puede enviar al dispositivo 110 de cálculo (1) un primer valor de direccionamiento criptográfico asociado con el primer conjunto de valores, (2) un segundo valor de direccionamiento criptográfico asociado con el segundo conjunto de valores, (3) un tercer valor de direccionamiento criptográfico asociado con el tercer conjunto de valores; y (4) cada valor del cuarto conjunto de valores. Los criterios para cuando es agregado un valor a un direccionamiento criptográfico pueden ser establecidos por un administrador, usuarios individuales, con base en un número de valores que ya están en la instancia de base de datos, y/o similares. El envío de valores de direccionamiento criptográficos en vez de cada valor individual reduce el número de valores individuales proporcionados cuando se intercambian valores entre instancias de bases de datos.

Cuando un dispositivo de cálculo receptor (por ejemplo, dispositivo 400 de cálculo en la etapa 2 de la figura8) recibe un valor de direccionamiento criptográfico (por ejemplo, generado por el dispositivo 500 de cálculo con base en valores en instancia 503 de base de datos distribuida), ese dispositivo de cálculo genera un valor de direccionamiento criptográfico usando el mismo método y/o proceso y los valores en su instancia de base de datos (por ejemplo, instancia 403 de base de datos distribuida) para los parámetros (por ejemplo, transacciones durante un período de tiempo especificado) usados para generar el valor de direccionamiento criptográfico recibido. El dispositivo de cálculo receptor puede luego comparar el valor de direccionamiento criptográfico recibido con el valor de direccionamiento criptográfico generado. Si los valores no coinciden, el dispositivo de cálculo receptor puede solicitar los valores individuales usados para generar el direccionamiento criptográfico recibido del dispositivo de cálculo emisor (por ejemplo, dispositivo 500 de cálculo en la figura 8) y comparar los valores individuales de la instancia de base de datos emisor (por ejemplo, instancia 503 de base de datos distribuida) con los valores individuales para esas transacciones en la instancia de base de datos recibida (por ejemplo, instancia 403 de base de datos distribuida).

Por ejemplo, si el dispositivo de cálculo receptor recibe el valor de direccionamiento criptográfico asociado con las transacciones entre el año 2000 y el año 2010, el dispositivo de cálculo receptor puede generar un direccionamiento criptográfico usando los valores para las transacciones entre el año 2000 y el año 2010 almacenado en su instancia de base de datos. Si el valor de direccionamiento criptográfico recibido coincide con el valor de direccionamiento criptográfico generado localmente, el dispositivo de cálculo receptor puede asumir que los valores para las transacciones entre el año 2000 y el año 2010 son los mismos en ambas bases de datos y no es solicitada información adicional. Sin embargo, si el valor de direccionamiento criptográfico recibido no coincide con el valor de direccionamiento criptográfico generado localmente, el dispositivo de cálculo receptor puede solicitar los valores individuales que el dispositivo de cálculo emisor usó para generar el valor de direccionamiento criptográfico recibido. El dispositivo de cálculo receptor puede entonces identificar la discrepancia y actualizar un vector de valores para ese valor individual.

Los valores de direccionamiento criptográficos pueden depender de cualquier proceso y/o función de direccionamiento adecuados para combinar múltiples valores y/o porciones de un valor en un único identificador. Por ejemplo, puede ser usado cualquier número adecuado de valores (por ejemplo, transacciones dentro de un período de tiempo) como

entradas para una función de direccionamiento y puede ser generado un valor de direccionamiento con base en la función de direccionamiento.

Aunque la discusión anterior usa valores de direccionamiento criptográficos como el identificador asociado con valores y/o porciones de valores, debe entenderse que pueden ser usados otros identificadores usados para representar múltiples valores y/o porciones de valores. Ejemplos de otros identificadores incluyen huellas digitales, sumas de verificación, valores de direccionamiento regulares, y/o similares.

La figura 12 muestra un diagrama de flujo (diagrama 20 de flujo) que ilustra las etapas realizadas por el dispositivo 110 de cálculo dentro del sistema 100 de base de datos distribuida, de acuerdo con una realización. En la realización ilustrada por la figura 12, el vector de valores se restablece con base en una probabilidad predefinida. Indicado de manera similar, cada valor en el vector de valores se puede restablecer a un valor cada cierto tiempo y con base en una probabilidad. En la etapa 21, el dispositivo 110 de cálculo selecciona un valor para el parámetro con base en el vector de valores para el parámetro, similar a la etapa 15 ilustrada en la figura 11 y discutida anteriormente. En la etapa 22, el dispositivo 110 de cálculo recibe valores para el parámetro de otros dispositivos de cálculo (por ejemplo, dispositivos 120, 130, 140 de cálculo) y envía un valor para el parámetro almacenado en la instancia 113 de base de datos distribuida a los otros dispositivos de cálculo (por ejemplo, dispositivos 120, 130, 140 de cálculo). Por ejemplo, la etapa 22 puede incluir realizar las etapas 12 y 13, ilustradas en la figura 11 y discutidas anteriormente, para cada uno de los otros dispositivos de cálculo. En la etapa 23, el dispositivo 110 de cálculo almacena los valores para el parámetro recibido de los otros dispositivos de cálculo (por ejemplo, dispositivos 120, 130, 140 de cálculo) en el vector de valores para el parámetro, similar a la etapa 14 ilustrada en la figura 11 y discutida anteriormente. En la etapa 24, el dispositivo 110 de cálculo determina si restablecer el vector de valores con base en una probabilidad predefinida de restablecer el vector de valores. En algunas instancias, por ejemplo, hay un 10% de probabilidad de que el dispositivo 110 de cálculo restablezca el vector de valores para el parámetro después de cada vez que el dispositivo 110 de cálculo actualice el vector de valores para el parámetro almacenado en la instancia 114 de base de datos distribuida. En tal escenario, el dispositivo 110 de cálculo, en la etapa 24, determinaría si se restablece o no, con base en el 10% de probabilidad. La determinación puede ser realizada, en algunas instancias, por el procesador 111 del dispositivo 110 de cálculo.

Si el dispositivo 110 de cálculo determina restablecer el vector de valores con base en la probabilidad predefinida, el dispositivo 110 de cálculo, en la etapa 25, restablece el vector de valores. En algunas realizaciones, el dispositivo 110 de cálculo puede restablecer cada valor en el vector de valores para el parámetro para que sea igual al valor para el parámetro almacenado en la instancia 113 de base de datos distribuida en el momento del restablecimiento. Por ejemplo, si, justo antes del restablecimiento, el vector de valores es el vector de valores 430, y el valor para el parámetro almacenado en la instancia 113 de base de datos distribuida es (1. Alice, 2. Bob, 3. Carol, 4. Dave, 5. Ed, 6. Frank) (por ejemplo, bajo "rango por mediana"), entonces cada valor en el vector de valores sería restablecido para que se igual (1. Alice, 2. Bob, 3. Carol, 4. Dave, 5. Ed, 6. Frank). En otras palabras, cada uno de los valores 431, 432, 433, 434, 435 del vector de valores 430 sería restablecido para que sea igual al valor 431. Restableciendo cada valor en el vector de valores para el parámetro para que sea igual al valor para el parámetro almacenado en la instancia de base de datos distribuida en el momento del restablecimiento, cada cierto tiempo y con base en una probabilidad, ayuda a un sistema de base de datos distribuida (al cual pertenece un dispositivo de cálculo) a llegar a un consenso. Indicado de manera similar, el restablecimiento facilita el acuerdo sobre el valor para un parámetro entre los dispositivos de cálculo de un sistema de base de datos distribuida.

Por ejemplo, la instancia 114 de base de datos distribuida del dispositivo 110 de cálculo puede almacenar un conjunto clasificado de jugadores (1. Alice, 2. Bob, 3. Carol, 4. Dave, 5. Ed, 6. Frank), similar al valor 431, lo que indica que la posesión del ítem particular comenzó con Alice, luego fue pasado a Bob, luego fue pasado a Carol, luego fue pasado a Dave, luego fue pasado a Ed, y finalmente fue pasado a Frank.

La figura 13 muestra un diagrama de flujo (diagrama 30 de flujo) que ilustra las etapas realizadas por el dispositivo 110 de cálculo dentro del sistema 100 de base de datos distribuida, de acuerdo con una realización. En la realización ilustrada por la figura 13, la selección para un valor del parámetro con base en un vector de valores para el parámetro se produce cuando un valor de confianza asociado con una instancia de la base de datos distribuida es cero. El valor de confianza puede indicar el nivel de "consenso", o acuerdo, entre el valor del parámetro almacenado en el dispositivo 110 de cálculo y los valores del parámetro almacenado en los otros dispositivos de cálculo (por ejemplo, dispositivos 120, 130, 140 de cálculo) del sistema 100 de base de datos distribuida. En algunas realizaciones, como se describe en detalle en este documento, el valor de confianza es incrementado (por ejemplo, incrementado en uno) cada vez que un valor para el parámetro recibido desde otro dispositivo de cálculo por el dispositivo 110 de cálculo es igual al valor para el parámetro almacenado en el dispositivo 110 de cálculo, y el valor de confianza es reducido (es decir, reducido en uno) cada vez que un valor para el parámetro recibido desde otro dispositivo de cálculo por el dispositivo 110 de cálculo no es igual al valor para el parámetro almacenado en el dispositivo 110 de cálculo, si el valor de confianza es superior a cero.

En la etapa 31, el dispositivo 110 de cálculo recibe un valor para el parámetro desde otro dispositivo de cálculo (por ejemplo, dispositivo 120 de cálculo) y envía un valor para el parámetro almacenado en la instancia 113 de base de datos distribuida al otro dispositivo de cálculo (por ejemplo, dispositivo 120 de cálculo). Por ejemplo, la etapa 31 puede incluir realizar las etapas 12 y 13, ilustradas en la figura 11 y discutidas anteriormente. En la etapa 32, el dispositivo

110 de cálculo almacena el valor para el parámetro recibido del otro dispositivo de cálculo (por ejemplo, dispositivo 120 de cálculo) en el vector de valores para el parámetro, similar a la etapa 14 ilustrada en la figura 11 y discutida anteriormente. En la etapa 33, el dispositivo 110 de cálculo determina si el valor para el parámetro recibido del otro dispositivo de cálculo (por ejemplo, dispositivo 120 de cálculo) es igual al valor para el parámetro almacenado en la instancia 113 de base de datos distribuida. Si el valor para el parámetro recibido del otro dispositivo de cálculo (por ejemplo, dispositivo 120 de cálculo) es igual al valor para el parámetro almacenado en la instancia 113 de base de datos distribuida, entonces el dispositivo 110 de cálculo, en la etapa 34, incrementa un valor de confianza asociado con la instancia 113 de base de datos distribuida en uno, y el proceso ilustrado por el diagrama 30 de flujo vuelve a la etapa 31. Si el valor para el parámetro recibido del otro dispositivo de cálculo (por ejemplo, dispositivo 120 de cálculo) no es igual al valor para el parámetro almacenado en la instancia 113 de base de datos distribuida, entonces el dispositivo 110 de cálculo, en la etapa 35, reduce el valor de confianza asociado con la instancia 113 de base de datos distribuida en uno, si el valor de confianza es mayor que cero.

En la etapa 36, el dispositivo 110 de cálculo determina si el valor de confianza asociado con la instancia 113 de base de datos distribuida es igual a cero. Si el valor de confianza es igual a cero, entonces el dispositivo de cálculo, en la etapa 37, selecciona un valor para el parámetro con base en el vector de valores para el parámetro. Esta selección puede ser realizada de acuerdo con cualquier método y/o proceso (por ejemplo, una regla o un conjunto de reglas), como se discutió anteriormente. Si el valor de confianza no es igual a cero, entonces el proceso ilustrado por el diagrama 30 de flujo vuelve a la etapa 31.

Como se discutió anteriormente, los valores de confianza están asociados con instancias de bases de datos distribuidas. Sin embargo, debe entenderse que un valor de confianza también puede ser asociado con un valor de un vector almacenado en una instancia de base de datos distribuida y/o el dispositivo de cálculo que almacena el valor de un vector (por ejemplo, dentro de su instancia de base de datos distribuida) en vez de, o además de, la instancia de base de datos distribuida.

Los valores relacionados con los valores de confianza (por ejemplo, umbrales, valores de incremento, y valores de reducción) usados con respecto a la figura 13 son solo con propósitos ilustrativos. Debe entenderse que pueden ser usados otros valores relacionados con los valores de confianza (por ejemplo, umbrales, valores de incremento, y valores de reducción). Por ejemplo, los aumentos y/o disminuciones del valor de confianza, usados en las etapas 34 y 35, respectivamente, pueden ser cualquier valor. Para otro ejemplo, el umbral de confianza de cero, usado en las etapas 35 y 36, también puede ser cualquier valor. Adicionalmente, los valores relacionados con los valores de confianza (por ejemplo, umbrales, valores de incremento, y valores de reducción) pueden cambiar durante el curso de operación, es decir, como el proceso ilustrado por los bucles del diagrama 30 de flujo.

En algunas realizaciones, el valor de confianza puede afectar el flujo de comunicación entre un primer dispositivo de cálculo de un sistema de base de datos distribuida y un segundo dispositivo de cálculo del sistema de base de datos distribuida, descrito anteriormente con respecto a la figura 8. Por ejemplo, si el primer dispositivo de cálculo (por ejemplo, dispositivo 110 de cálculo) tiene un alto valor de confianza asociado con su instancia de base de datos distribuida (por ejemplo, instancia 114 de base de datos distribuida), entonces el primer dispositivo de cálculo puede solicitar del segundo dispositivo de cálculo una porción más pequeña de un valor para un parámetro (y un valor de direccionamiento criptográfico asociado con una porción mayor del valor para el parámetro) de lo que el primer dispositivo de cálculo solicitaría de otro modo del segundo dispositivo de cálculo (por ejemplo, si el primer dispositivo de cálculo tiene un bajo valor de confianza asociado con su instancia de base de datos distribuida). El alto valor de confianza puede indicar que es probable que el valor para el parámetro almacenado en el primer dispositivo de cálculo esté de acuerdo con valores para el parámetro almacenado en otros dispositivos de cálculo del sistema de base de datos distribuida y como tal, es usado un valor de direccionamiento criptográfico para verificar el acuerdo.

En algunas instancias, el valor de confianza del primer dispositivo de cálculo puede aumentar para alcanzar un umbral en el cual el primer dispositivo de cálculo determina que ya no debe solicitar valores particulares, porciones particulares de valores, y/o valores de direccionamiento criptográficos asociados con valores particulares y/o porciones particulares de valores de otros dispositivos de cálculo del sistema de base de datos distribuida. Por ejemplo, si un valor de confianza del valor cumple con un criterio específico (por ejemplo, alcanza un umbral), el primer dispositivo de cálculo puede determinar que el valor ha convergido y no solicitar además intercambiar este valor con otros dispositivos. Para otro ejemplo, el valor puede ser agregado a un valor de direccionamiento criptográfico con base en que su valor de confianza cumpla con un criterio. En tales instancias, el valor de direccionamiento criptográfico para el conjunto de valores puede ser enviado en vez del valor individual después de que el valor de confianza cumpla con el criterio, como se discutió en detalle anteriormente. El intercambio de menos valores, y/o porciones reales más pequeñas (de valores) con valores de direccionamiento criptográficos asociados con las porciones restantes (de valores) puede facilitar la comunicación eficiente entre dispositivos de cálculo de un sistema de base de datos distribuida.

En algunas instancias, a medida que aumenta el valor de confianza para el valor específico de un parámetro de una instancia de base de datos distribuida, el dispositivo de cálculo asociado con esa instancia de base de datos distribuida puede solicitar intercambiar valores para ese parámetro con otros dispositivos de cálculo con menos frecuencia. De manera similar, en algunas instancias, a medida que disminuye el valor de confianza para un valor específico de un parámetro de una instancia de base de datos distribuida, el dispositivo de cálculo asociado con esa instancia de base de datos distribuida puede solicitar intercambiar valores para ese parámetro con otros dispositivos de cálculo con más

frecuencia. De este modo, el valor de confianza puede ser usado para disminuir un número de valores intercambiados entre dispositivos de cálculo.

Aunque han sido descritas anteriormente diversas realizaciones, debe entenderse que han sido presentadas solamente a modo de ejemplo, y no de limitación. Cuando los métodos descritos anteriormente indican que ciertos eventos se producen en cierto orden, el orden de ciertos eventos puede ser modificado. Adicionalmente, ciertos de los eventos pueden ser realizados simultáneamente en un proceso paralelo cuando sea posible, así como realizados secuencialmente como se describió anteriormente.

Algunas realizaciones descritas en este documento se relacionan con un producto de almacenamiento de ordenador con un medio legible por ordenador no transitorio (también puede denominarse como un medio legible por procesador no transitorio) que tiene instrucciones o código de ordenador en el mismo para realizar diversas operaciones implementadas por ordenador. El medio legible por ordenador (o medio legible por procesador) no es transitorio en el sentido de que no incluye señales de propagación transitorias per se (por ejemplo, una onda electromagnética de propagación que porta información en un medio de transmisión tal como espacio o un cable). Los medios y código de ordenador (también pueden denominarse como código) pueden ser aquellos diseñados y construidos para el propósito o propósitos específicos. Ejemplos de medios legibles por ordenador no transitorios incluyen, pero no se limitan a: medios de almacenamiento magnético tales como discos duros, discos flexibles, y cinta magnética; medios de almacenamiento óptico tales como Discos Compactos/Discos de Vídeo Digitales (CD/DVDs), Memorias de Sólo Lectura en Disco Compacto (CD-ROMs), y dispositivos holográficos; medios de almacenamiento magnetoópticos tales como discos ópticos; módulos de procesamiento de señales de onda portadora; y dispositivos de hardware que están especialmente configurados para almacenar y ejecutar código de programa, tales como Circuitos Integrados de Aplicación Específica (ASICs), Dispositivos Lógicos Programables (PLDs), Memoria de Solo Lectura (ROM) y dispositivos de Memoria de Acceso Aleatorio (RAM). Otras realizaciones descritas en este documento se relacionan con un producto de programa de ordenador, el cual puede incluir, por ejemplo, las instrucciones y/o código de ordenador discutido en este documento.

Ejemplos de código de ordenador incluyen, pero no se limitan a, microcódigo o microinstrucciones, instrucciones de máquina, tales como producidas por un compilador, código usado para producir un servicio web, y archivos que contienen instrucciones de nivel superior que son ejecutadas por un ordenador usando un intérprete. Por ejemplo, realizaciones pueden ser implementadas usando lenguajes de programación imperativos (por ejemplo, C, Fortran, etc.), lenguajes de programación funcionales (Haskell, Erlang, etc.), lenguajes de programación lógica (por ejemplo, Prolog), lenguajes de programación orientados a objetos (por ejemplo, Java, C++, etc.) u otros lenguajes de programación y/o herramientas de desarrollo adecuados. Ejemplos adicionales de código de ordenador incluyen, pero no se limitan a, señales de control, código encriptado, y código comprimido.

Aunque han sido descritas anteriormente diversas realizaciones, debe entenderse que han sido presentadas solamente a modo de ejemplo, no de limitación, y pueden ser hechos diversos cambios en forma y detalles. Cualquier porción del aparato y/o métodos descritos en este documento puede combinarse en cualquier combinación, excepto combinaciones mutuamente excluyentes. Las realizaciones descritas en este documento pueden incluir diversas combinaciones y/o subcombinaciones de las funciones, componentes y/o características de las diferentes realizaciones descritas.

**REIVINDICACIONES**

1. Un método que comprende:

recibir, en un primer dispositivo (110) de cálculo, un primer evento de una instancia (124) de una base de datos (100) distribuida, estando la instancia (124) de la base de datos (100) distribuida en un segundo dispositivo (120) de cálculo de una pluralidad de dispositivos (110, 120, 130, 140) de cálculo que implementan la base de datos distribuida a través de una red (105) acoplada operativamente a la pluralidad de dispositivos (110, 120, 130, 140) de cálculo;

definir, a través del primer dispositivo (110) de cálculo y con base en el primer evento y un segundo evento, un tercer evento;

determinar, a través del primer dispositivo de (110) ordenador, un primer conjunto de eventos basados al menos en parte en el tercer evento,

cada evento del primer conjunto de eventos es:

a) identificado por un segundo conjunto de eventos, un valor de peso colectivo asociado con el segundo conjunto de eventos que cumple con un primer criterio de valor de peso, cada evento del segundo conjunto de eventos (1) siendo definido por una instancia diferente de la base de datos distribuida y (2) siendo identificado por el tercer evento, y

b) asociado con un número de primera ronda; calcular, a través del primer dispositivo de (110) ordenador, un número redondo para el tercer evento con base en una determinación de que una suma de valores de peso asociados con cada evento del primer conjunto de eventos cumple con un segundo criterio de valor de peso, correspondiendo un número redondo para el primer evento a un número de segunda ronda mayor que el número de primera ronda;

determinando, a través del primer dispositivo de (110) ordenador, un tercer conjunto de eventos con base en el tercer evento, cada evento del tercer conjunto de eventos es:

a) identificado por un cuarto conjunto de eventos que incluye el tercer evento, siendo definido cada evento del cuarto conjunto de eventos por una instancia diferente de la base de datos distribuida, un valor de peso colectivo asociado con el cuarto conjunto de eventos que cumple un tercer criterio de valor de peso y

b) del primer conjunto de eventos;

definir, a través del primer dispositivo de cálculo, un valor de orden para un cuarto evento con base en un valor de peso colectivo asociado con el tercer conjunto de eventos que cumplen un cuarto criterio de valor de peso;

y almacenar el valor de orden en una instancia (114) de la base de datos (100) distribuida asociada con el primer dispositivo (110) de cálculo de la pluralidad de dispositivos (110, 120, 130, 140) de cálculo.

2. El método de la reivindicación 1, en donde un conjunto de valores de peso incluye un valor de peso asociado con cada instancia de la base de datos distribuida que define un evento del segundo conjunto de eventos, el valor de peso colectivo asociado con el segundo conjunto de eventos se basa en una suma de valores de peso del conjunto de valores de peso.

3. El método de la reivindicación 1, en el que un conjunto de valores de peso incluye un valor de peso (1) asociado con cada instancia de la base de datos distribuida que define un evento del segundo conjunto de eventos y (2) proporcional a una cantidad de criptomoneda asociada con esa instancia de la base de datos distribuida, basándose el valor de peso colectivo asociado con el segundo conjunto de eventos en una suma de valores de peso del conjunto de valores de peso.

4. El método de la reivindicación 1, en donde el segundo evento es del segundo dispositivo de cálculo.

5. El método de la reivindicación 1, que comprende además:

calcular, mediante el primer dispositivo (110) de cálculo, un valor de direccionamiento basado en el valor del pedido, estando asociado el valor de direccionamiento únicamente con un estado de consenso de la base de datos distribuida.

6. El método de la reivindicación 1, en donde:

cada evento del primer conjunto de eventos está definido por una instancia diferente de la base de datos distribuida, y

cada evento del primer conjunto de eventos es un evento más antiguo que tiene el primer número de ronda de un conjunto de eventos definido por la instancia de la base de datos distribuida que define ese evento.

7. El método de la reivindicación 1, en donde al menos uno del primer criterio de valor de peso, el segundo criterio de valor de peso, el tercer criterio de valor de peso o el cuarto criterio de valor de peso se define con base en un valor de peso colectivo de la base de datos distribuida.

8. El método de la reivindicación 1, en donde la pluralidad de dispositivos de ordenandor que implementan la base de datos distribuida por primera vez están asociados con un primer conjunto de organizaciones, la pluralidad de dispositivos de cálculo que implementan la base de datos distribuida en una segunda vez después de la primera vez. están asociados con un segundo conjunto de organizaciones que incluye organizaciones que no pertenecen al primer conjunto de organizaciones.

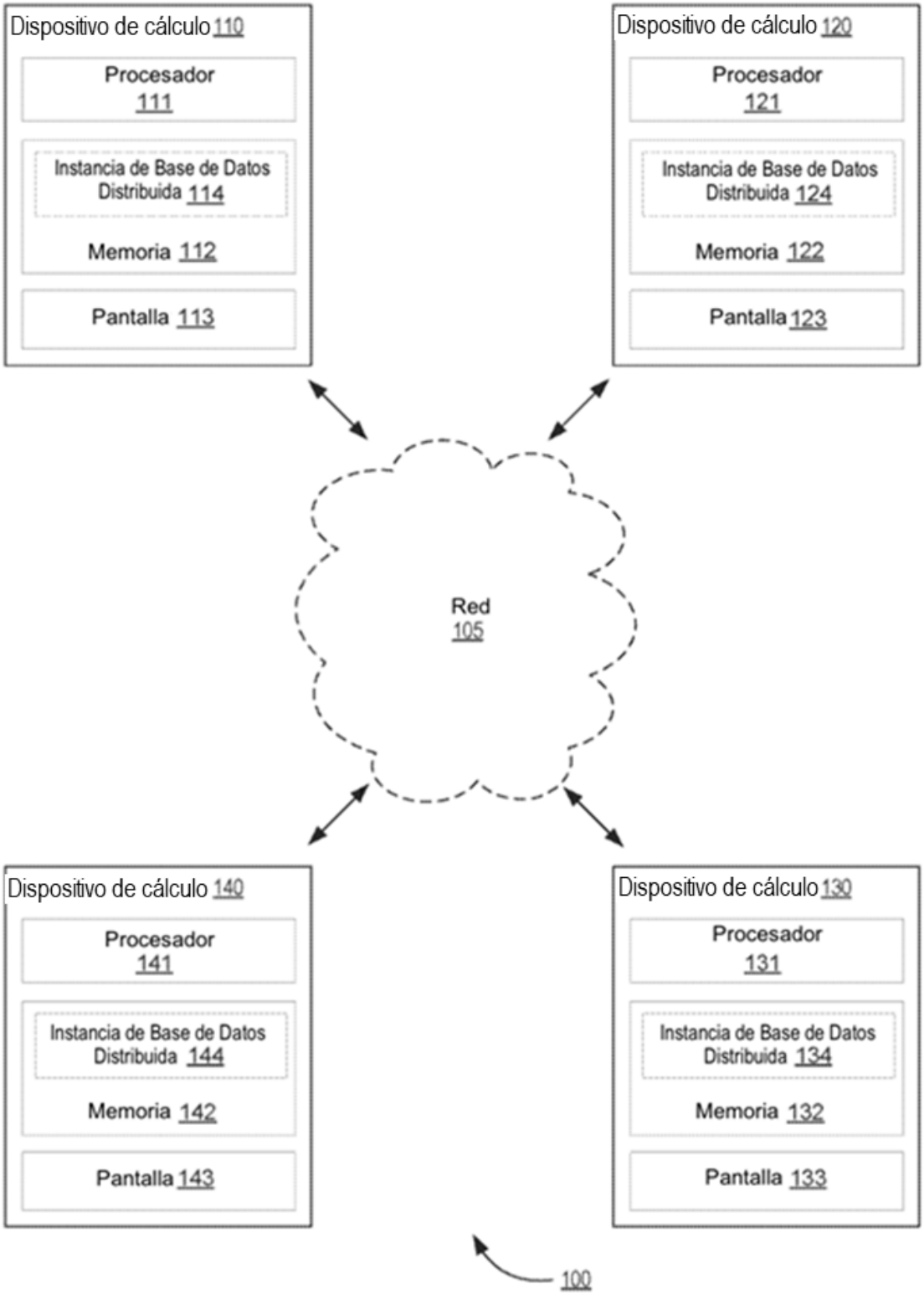
Dispositivo de cálculo 110
Procesador 111
Instancia de Base de Datos Distribuida 114
Memoria 112
Pantalla 113
Dispositivo de cálculo 120
Procesador 121
Instancia de Base de Datos Distribuida 124
Memoria 122
Pantalla 123
Red 105
Dispositivo de cálculo 140
Procesador 141
Instancia de Base de Datos Distribuida 144
Memoria 142
Pantalla 143
Dispositivo de cálculo 130
Procesador 131
Instancia de Base de Datos Distribuida 134
Memoria 132
Pantalla 133
100

FIG. 1


Dispositivo de cálculo 200
Módulo de Convergencia de Base de Datos 211
Módulo de Comunicación 212
Procesador 210
Instancia de Base de Datos Distribuida 221
Memoria 220

FIG. 2

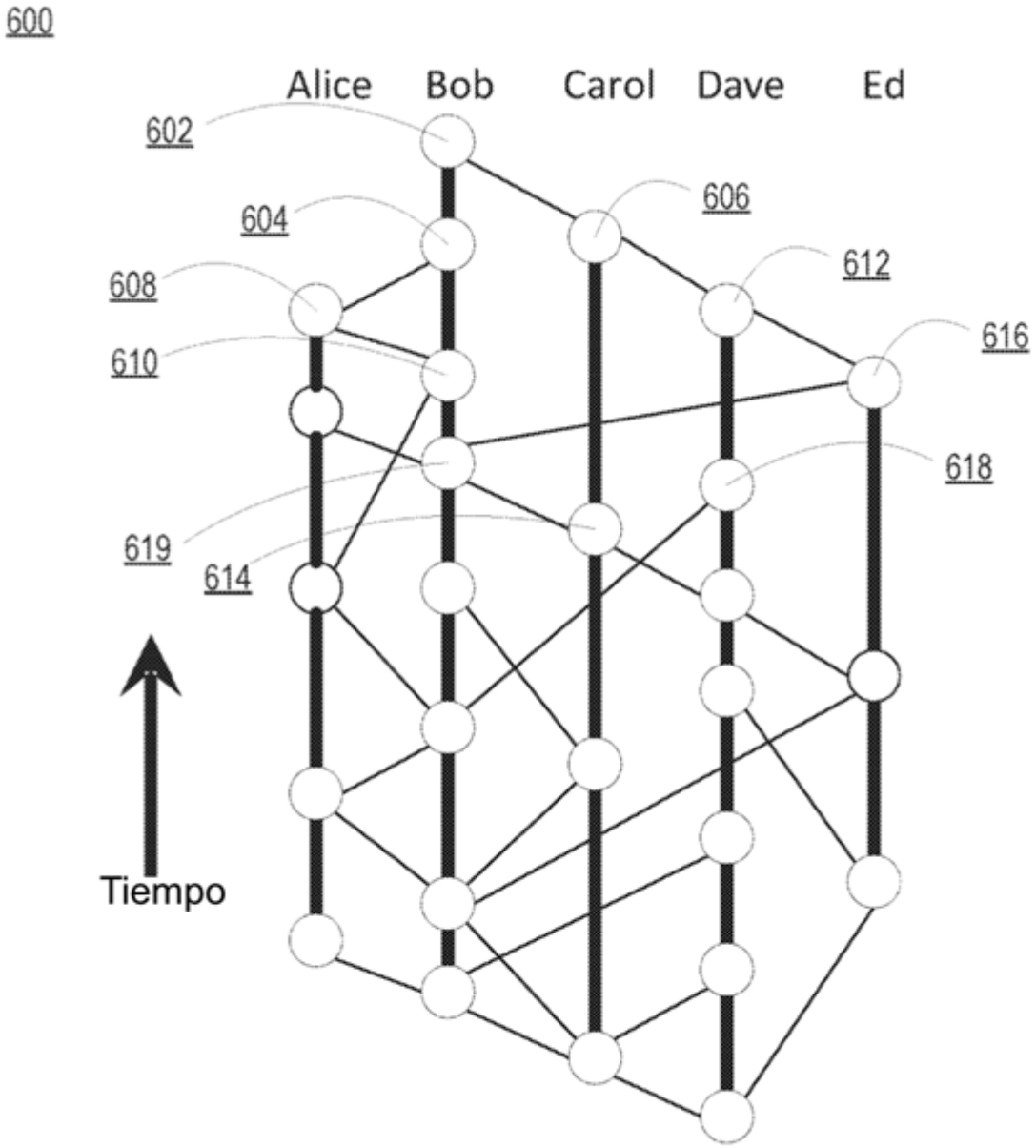
600
Alice
Bob
Carol
Dave
Ed
602
604
606
608
610
612
616
618
619
614
Tiempo

FIG. 3

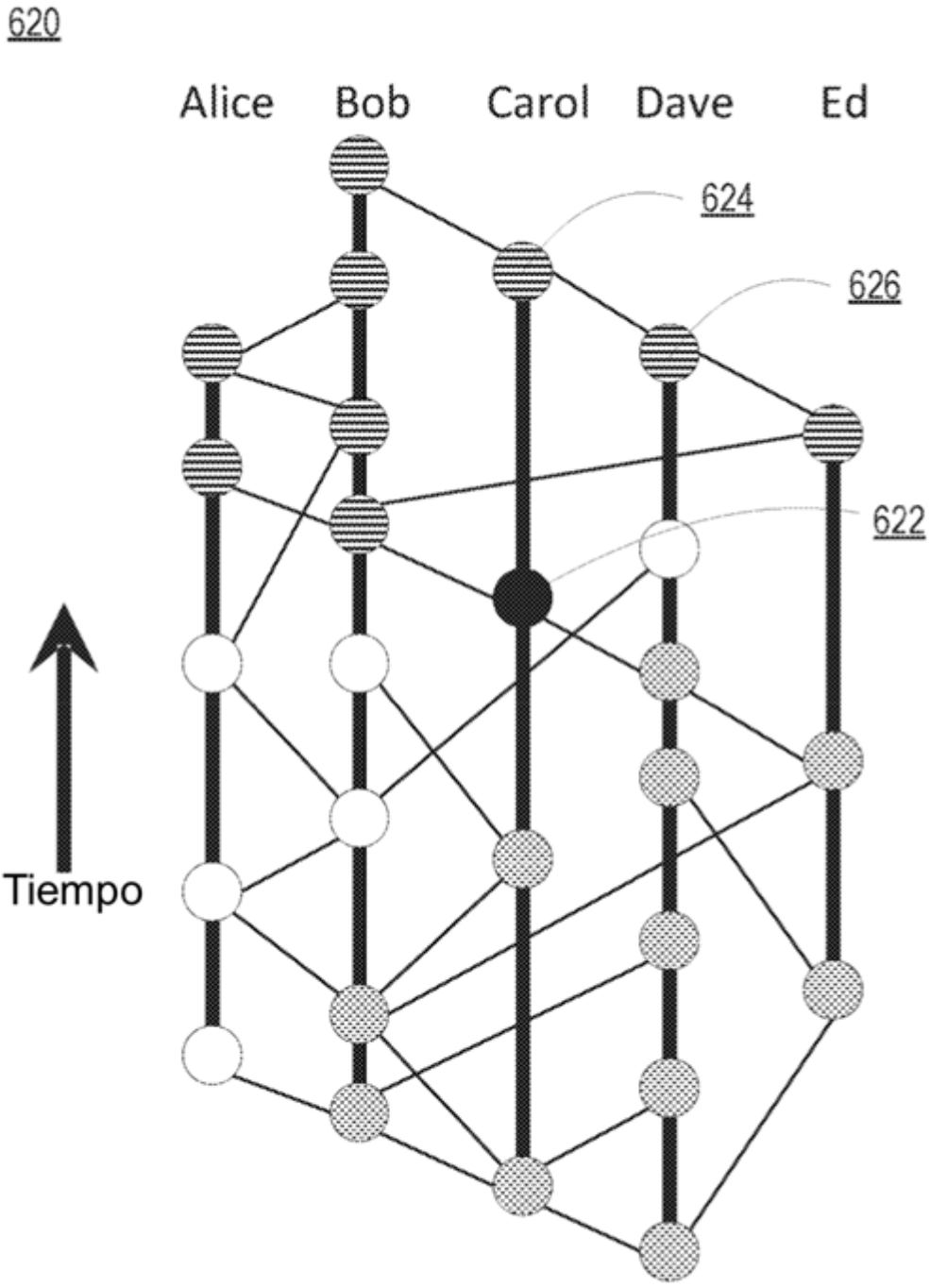


FIG. 4

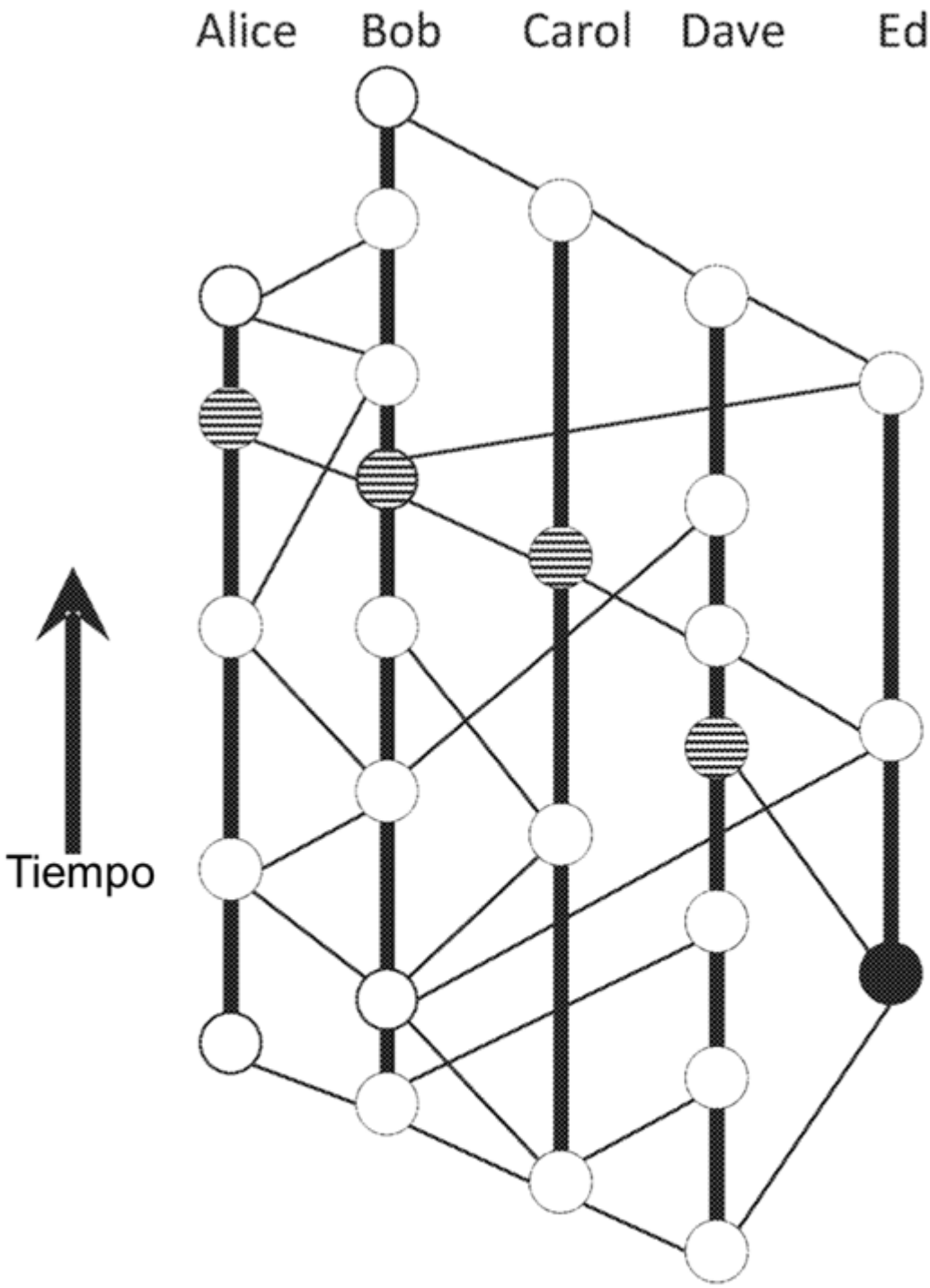


FIG. 5

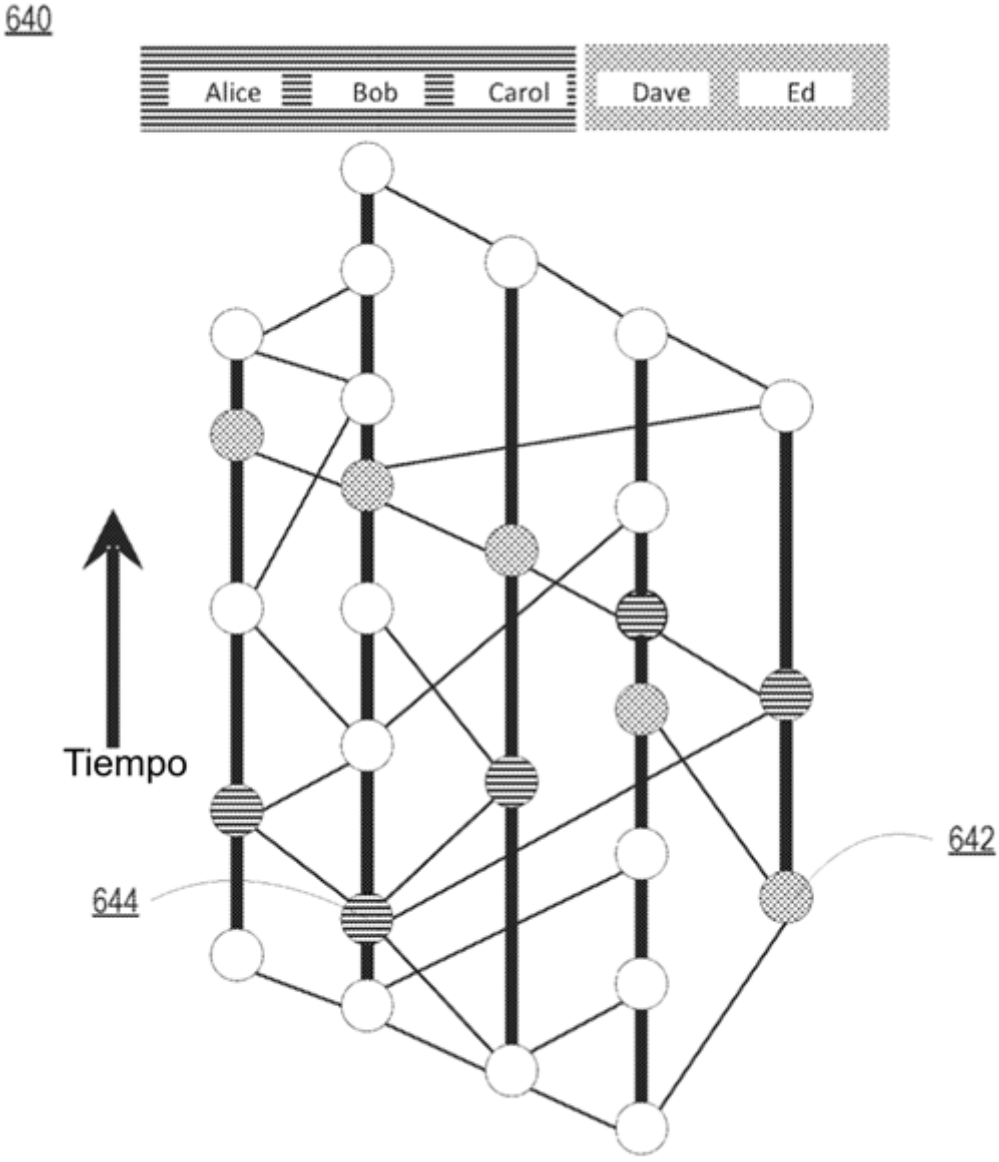


FIG. 6

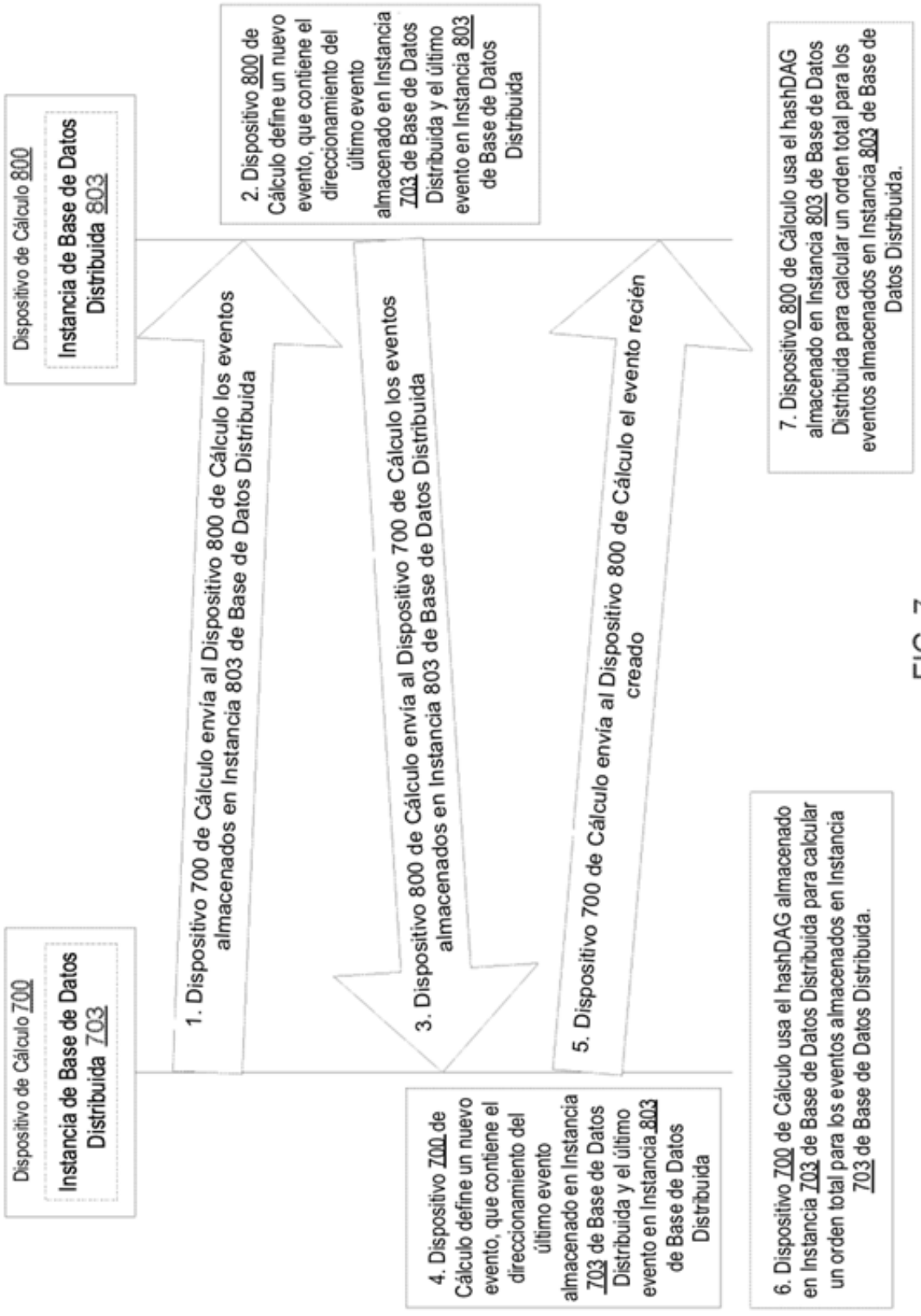
Dispositivo de Cálculo 800
Instancia de Base de Datos Distribuida 803
Dispositivo de Cálculo 700
Instancia de Base de Datos Distribuida 703
1. Dispositivo 700 de Cálculo envía al Dispositivo 800 de Cálculo los eventos almacenados en Instancia 803 de Base de Datos Distribuida
2. Dispositivo 800 de Cálculo define un nuevo evento, que contiene el direccionamiento del último evento almacenado en Instancia 703 de Base de Datos Distribuida y el último evento en Instancia 803 de Base de Datos Distribuida
3. Dispositivo 800 de Cálculo envía al Dispositivo 700 de Cálculo los eventos almacenados en Instancia 803 de Base de Datos Distribuida
4. Dispositivo 700 de Cálculo define un nuevo evento, que contiene el direccionamiento del último evento almacenado en Instancia 703 de Base de Datos Distribuida y el último evento en Instancia 803 de Base de Datos Distribuida
5. Dispositivo 700 de Cálculo envía al Dispositivo 800 de Cálculo el evento recién creado
6. Dispositivo 700 de Cálculo usa el hashDAG almacenado en Instancia 703 de Base de Datos Distribuida para calcular un orden total para los eventos almacenados en Instancia 703 de Base de Datos Distribuida.
7. Dispositivo 800 de Cálculo usa el hashDAG almacenado en Instancia 803 de Base de Datos Distribuida para calcular un orden total para los eventos almacenados en Instancia 803 de Base de Datos Distribuida.

FIG. 7

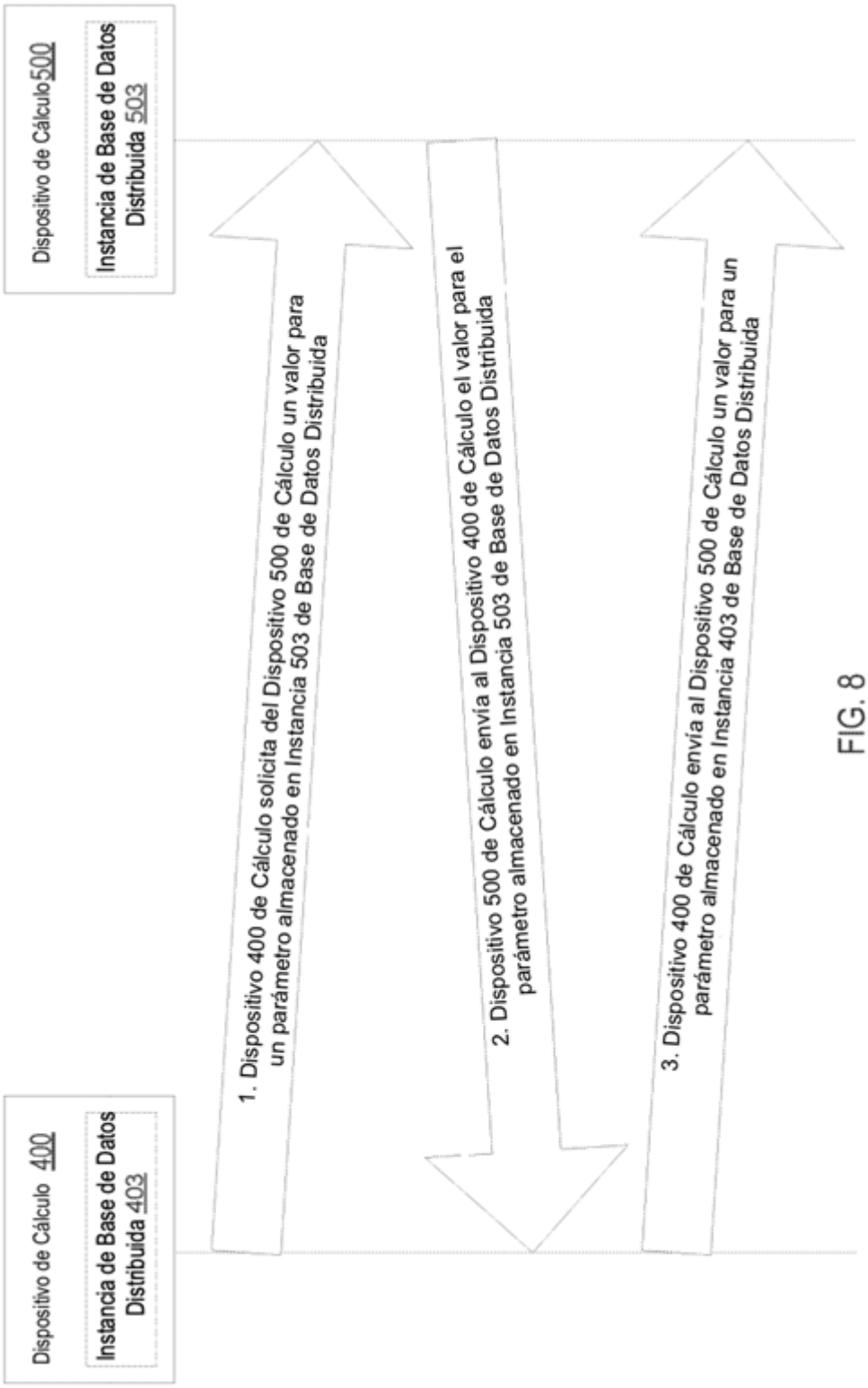
Dispositivo de Cálculo 500
Instancia de Base de Datos Distribuida 503
Dispositivo de Cálculo 400
Instancia de Base de Datos Distribuida 403
1. Dispositivo 400 de Cálculo solicita del Dispositivo 500 de Cálculo un valor para un parámetro almacenado en Instancia 503 de Base de Datos Distribuida
2. Dispositivo 500 de Cálculo envía al Dispositivo 400 de Cálculo el valor para el parámetro almacenado en Instancia 503 de Base de Datos Distribuida
3. Dispositivo 400 de Cálculo envía al Dispositivo 500 de Cálculo un valor para un parámetro almacenado en Instancia 403 de Base de Datos Distribuida

FIG. 8

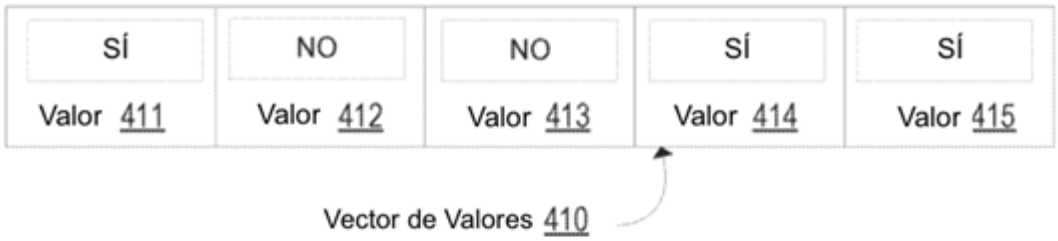


FIG. 9a

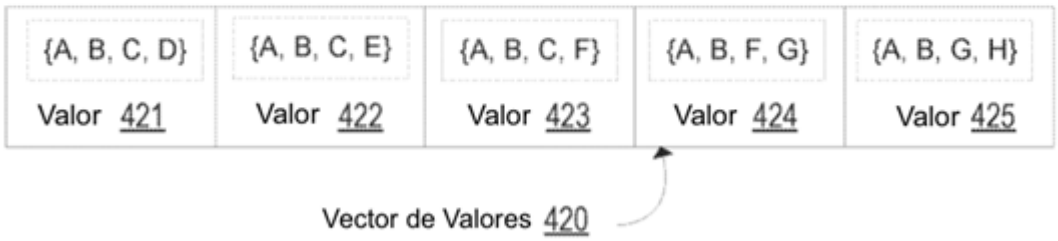


FIG. 9b


FIG. 9c


FIG. 10a


FIG. 10b


FIG. 10c

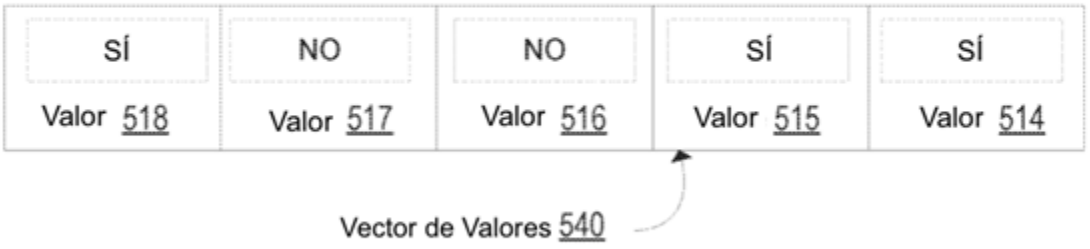


FIG. 10d

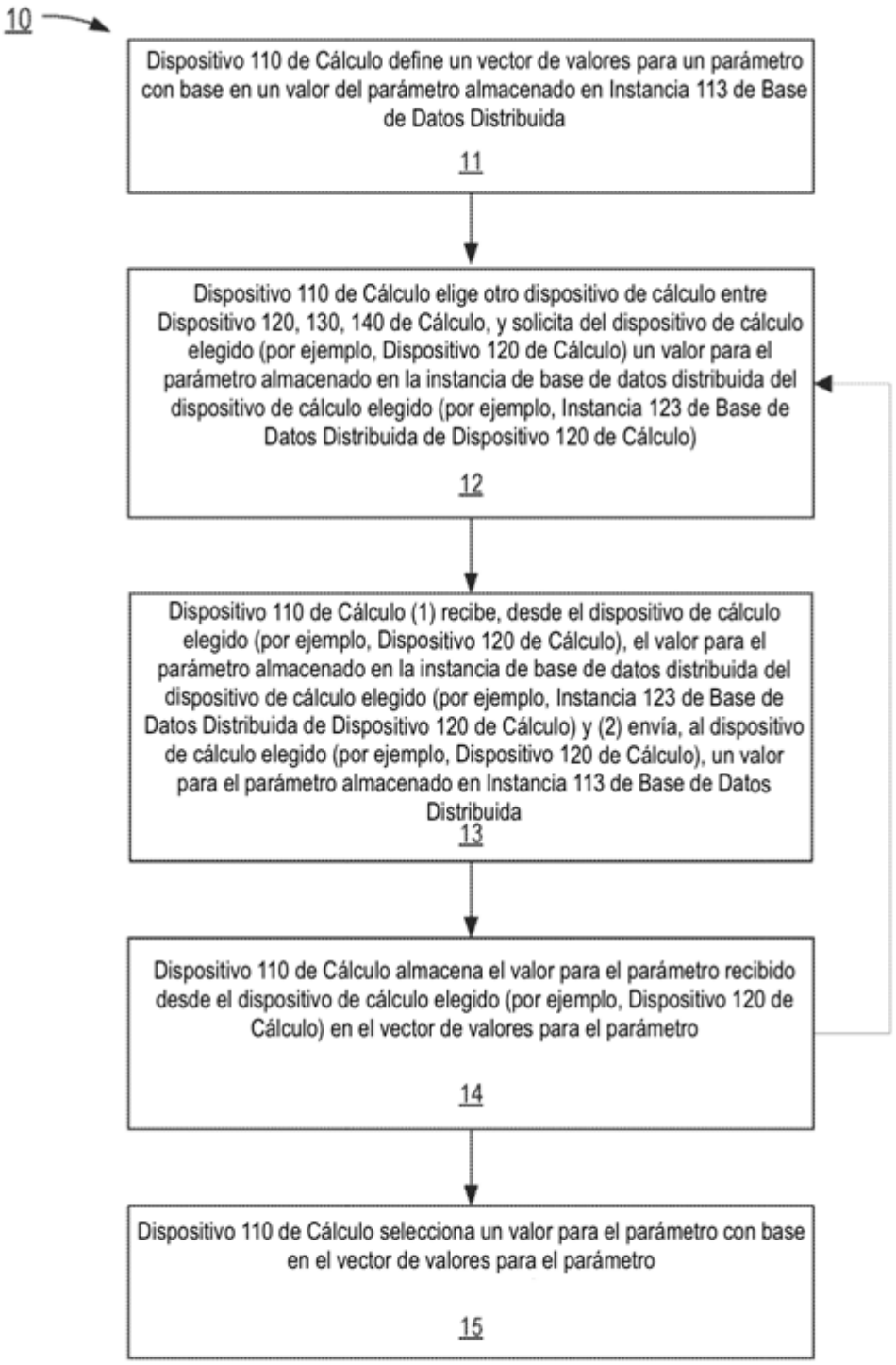
10
Dispositivo 110 de Cálculo define un vector de valores para un parámetro con base en un valor del parámetro almacenado en Instancia 113 de Base de Datos Distribuida
11
Dispositivo 110 de Cálculo elige otro dispositivo de cálculo entre Dispositivo 120, 130, 140 de Cálculo, y solicita del dispositivo de cálculo elegido (por ejemplo, Dispositivo 120 de Cálculo) un valor para el parámetro almacenado en la instancia de base de datos distribuida del dispositivo de cálculo elegido (por ejemplo, Instancia 123 de Base de Datos Distribuida de Dispositivo 120 de Cálculo)
12
Dispositivo 110 de Cálculo (1) recibe, desde el dispositivo de cálculo elegido (por ejemplo, Dispositivo 120 de Cálculo), el valor para el parámetro almacenado en la instancia de base de datos distribuida del dispositivo de cálculo elegido (por ejemplo, Instancia 123 de Base de Datos Distribuida de Dispositivo 120 de Cálculo) y (2) envía, al dispositivo de cálculo elegido (por ejemplo, Dispositivo 120 de Cálculo), un valor para el parámetro almacenado en Instancia 113 de Base de Datos Distribuida
13
Dispositivo 110 de Cálculo almacena el valor para el parámetro recibido desde el dispositivo de cálculo elegido (por ejemplo, Dispositivo 120 de Cálculo) en el vector de valores para el parámetro
14
Dispositivo 110 de Cálculo selecciona un valor para el parámetro con base en el vector de valores para el parámetro
15

FIG. 11

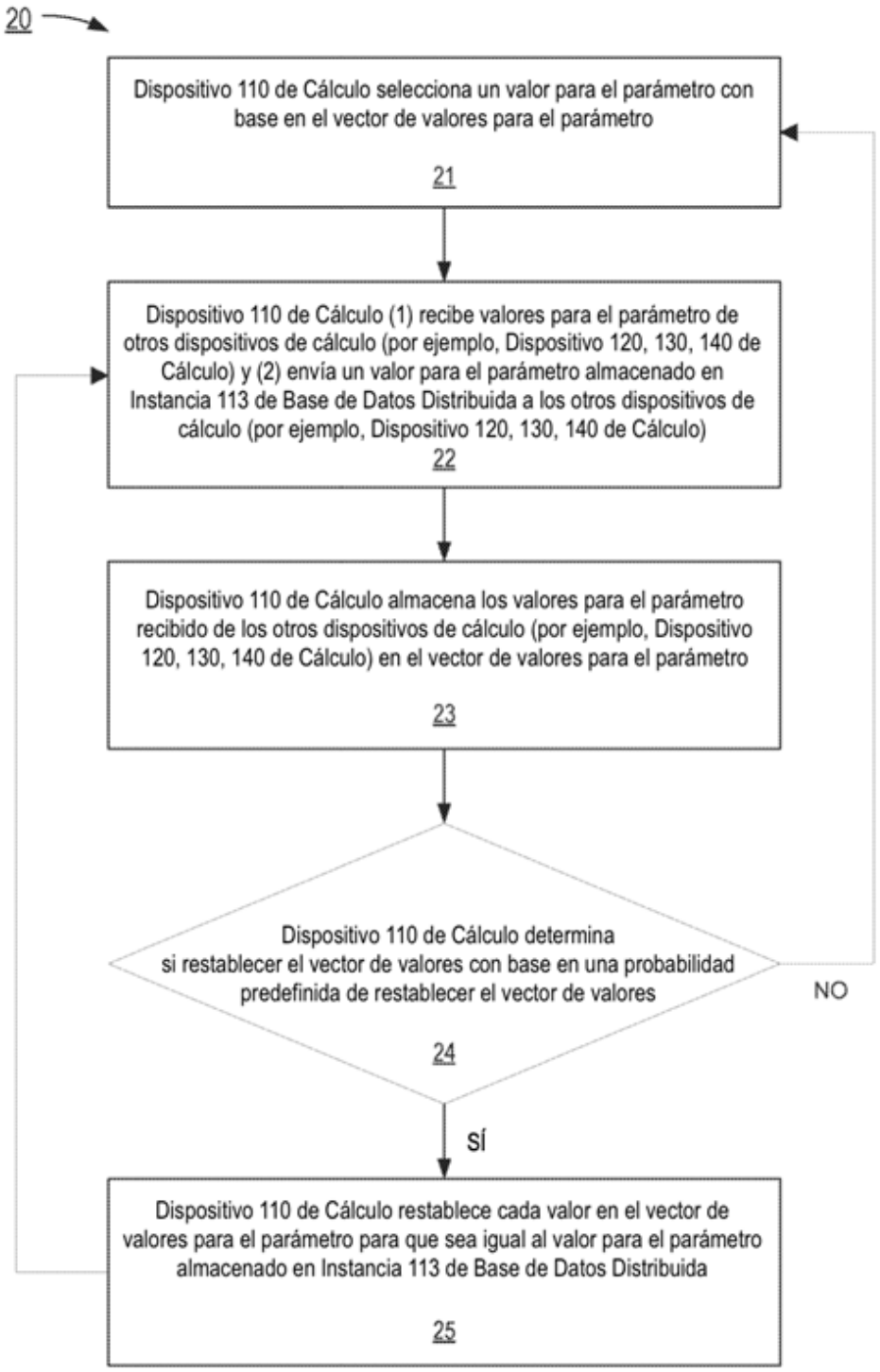
20
Dispositivo 110 de Cálculo selecciona un valor para el parámetro con base en el vector de valores para el parámetro
21
Dispositivo 110 de Cálculo (1) recibe valores para el parámetro de otros dispositivos de cálculo (por ejemplo, Dispositivo 120, 130, 140 de Cálculo) y (2) envía un valor para el parámetro almacenado en Instancia 113 de Base de Datos Distribuida a los otros dispositivos de cálculo (por ejemplo, Dispositivo 120, 130, 140 de Cálculo)
22
Dispositivo 110 de Cálculo almacena los valores para el parámetro recibido de los otros dispositivos de cálculo (por ejemplo, Dispositivo 120, 130, 140 de Cálculo) en el vector de valores para el parámetro
23
Dispositivo 110 de Cálculo determina si restablecer el vector de valores con base en una probabilidad predefinida de restablecer el vector de valores
24
NO
SÍ
Dispositivo 110 de Cálculo restablece cada valor en el vector de valores para el parámetro para que sea igual al valor para el parámetro almacenado en Instancia 113 de Base de Datos Distribuida
25

FIG. 12

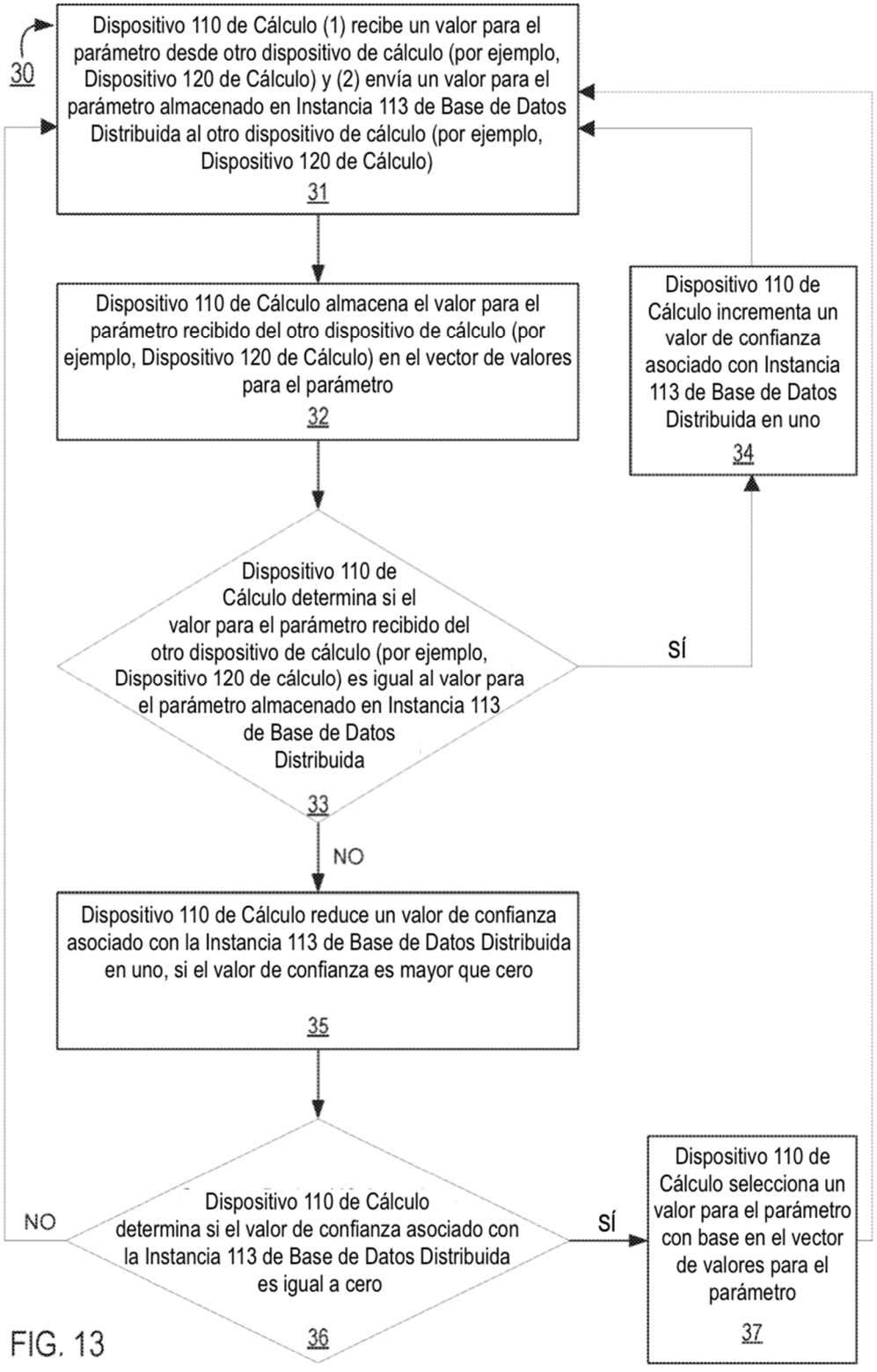
30
Dispositivo 110 de Cálculo (1) recibe un valor para el parámetro desde otro dispositivo de cálculo (por ejemplo, Dispositivo 120 de Cálculo) y (2) envía un valor para el parámetro almacenado en Instancia 113 de Base de Datos Distribuida al otro dispositivo de cálculo (por ejemplo, Dispositivo 120 de Cálculo)
31
Dispositivo 110 de Cálculo almacena el valor para el parámetro recibido del otro dispositivo de cálculo (por ejemplo, Dispositivo 120 de Cálculo) en el vector de valores para el parámetro
32
Dispositivo 110 de Cálculo incrementa un valor de confianza asociado con Instancia 113 de Base de Datos Distribuida en uno
34
Dispositivo 110 de Cálculo determina si el valor para el parámetro recibido del otro dispositivo de cálculo (por ejemplo, Dispositivo 120 de cálculo) es igual al valor para el parámetro almacenado en Instancia 113 de Base de Datos Distribuida
33
SÍ
NO
Dispositivo 110 de Cálculo reduce un valor de confianza asociado con la Instancia 113 de Base de Datos Distribuida en uno, si el valor de confianza es mayor que cero
35
Dispositivo 110 de Cálculo determina si el valor de confianza asociado con la Instancia 113 de Base de Datos Distribuida es igual a cero
36
NO
SÍ
Dispositivo 110 de Cálculo selecciona un valor para el parámetro con base en el vector de valores para el parámetro
37

FIG. 13

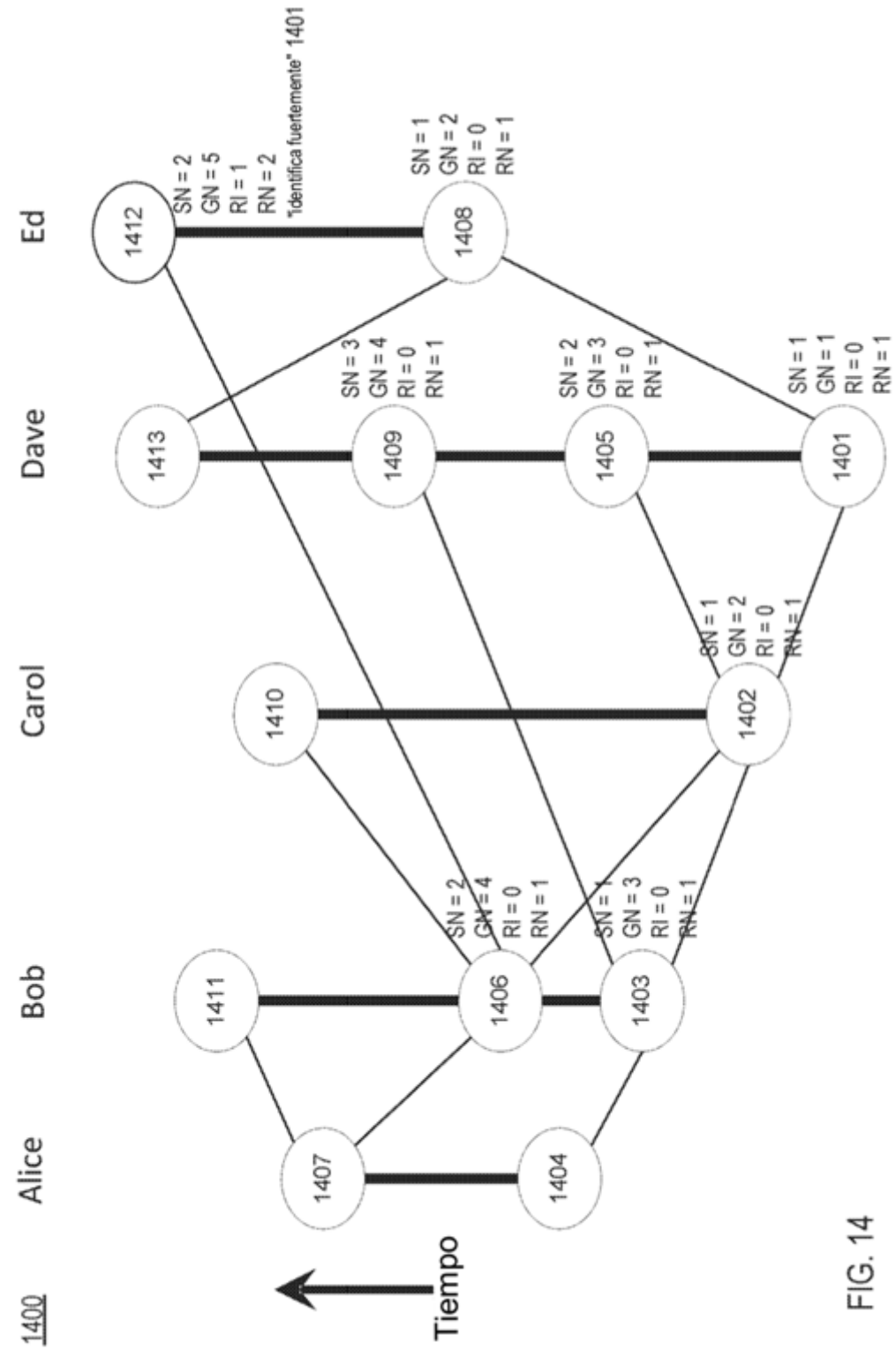
1400
Alice
Bob
Carol
Dave
Ed
Tiempo
1412
SN = 2
GN = 5
RI = 1
RN = 2
"identifica fuertemente" 1401
1408
SN = 1
GN = 2
RI = 0
RN = 1
1413
1409
SN = 3
GN = 4
RI = 0
RN = 1
1405
SN = 2
GN = 3
RI = 0
RN = 1
1401
SN = 1
GN = 1
RI = 0
RN = 1
1410
1402
SN = 1
GN = 2
RI = 0
RN = 1
1411
1406
SN = 2
GN = 4
RI = 0
RN = 1
1403
SN = 1
GN = 3
RI = 0
RN = 1
1407
1404

FIG. 14

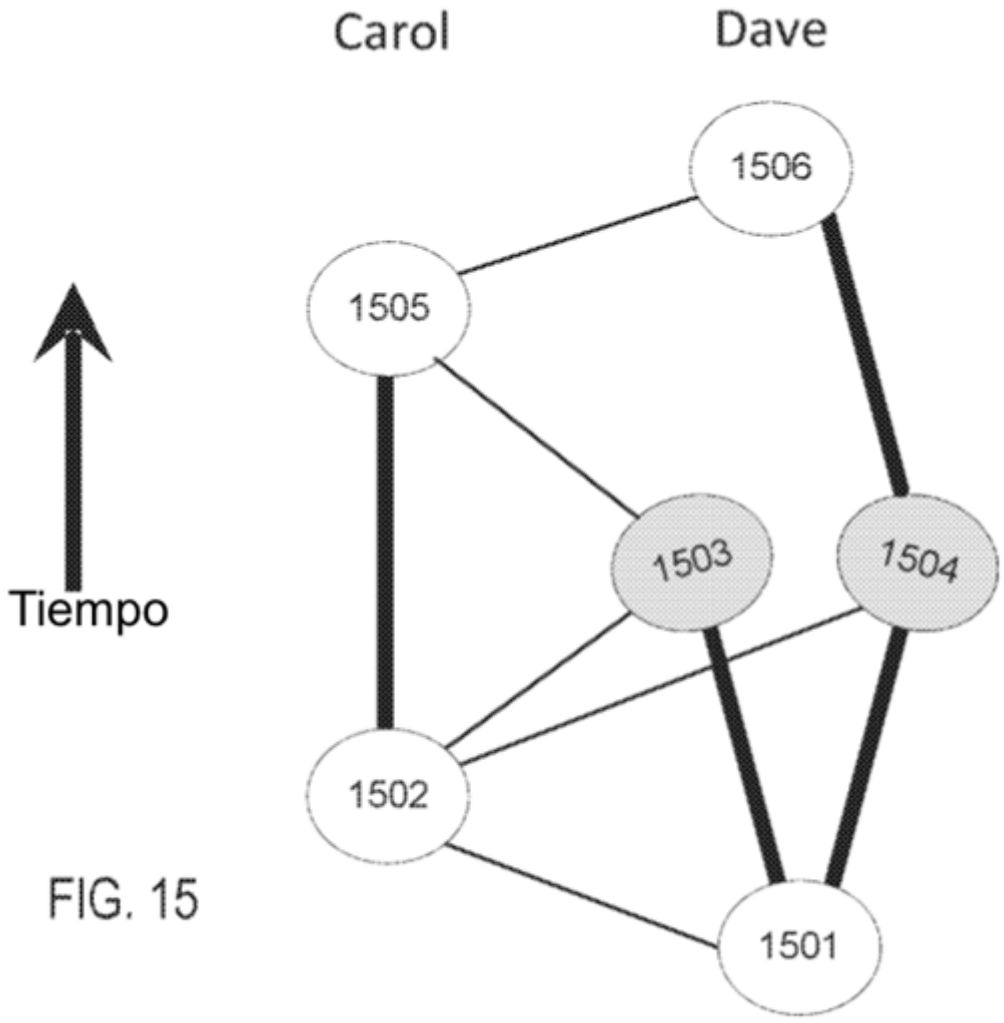


FIG. 15

Un evento es una tupla e = {d, h, t, c, s} donde:

$d$ = datos(e) = los datos de "carga útil", que pueden incluir transacciones.
$h$ = direccionamientos(e) = una lista de direccionamientos de los padres del evento, padre propio primero.
$t$ = tiempo(e) = datos reivindicados del creador y hora de creación del evento.
$c$ = creador(e) = número de ID del creador.
$s$ = $sig(c)$ = firma digital del creador de {d, h, t, c}.

$n$ = el número de miembros en la población
$m$ = $1 + \lfloor 2n/3 \rfloor$
primero = el evento único que no tiene padres
$E$ = el conjunto de todos los eventos
$\mathbb{T}$ = conjunto de todos los posibles pares (hora, fecha)
$\mathbb{B}$ = {verdadero, falso}
$\mathbb{N}$ = $\{0, 1, 2, ...\}$

antepasado : $E \times E \to B$
antepasadoPropio : $E \times E \to B$
ver : $E \times E \to B$
verFuertemente : $E \times E \to B$
rondadePadre : $E \to \mathbb{N}$
testigo : $E \to \mathbb{B}$
ronda : $E \to \mathbb{N}$
rondaDif. : $E \times E \to \mathbb{I}$
votos : $E \times E \times \mathbb{B} \to \mathbb{N}$
fracciónde Voto : $E \times E \to \mathbb{R}$
voto : $E \times E \to \mathbb{B}$
decidir : $E \times E \to \mathbb{B}$
todosFamosos : $\mathbb{I} \to 2^E$
famoso : $E \to \mathbb{B}$
rondaRecibida : $E \to \mathbb{N}$
tiempoRecibido : $E \to \mathbb{T}$

FIG. 16a

$$
\begin{aligned}
\text{antepasado}(x,y) &= (x=y) \vee (\exists z \in \text{padres}(x) : \text{antepasado}(z,y)) \\
\text{antepasadoPropio}(x,y) &= \text{antepasado}(x,y) \wedge ((\text{padrePropio}(x)=y) \vee \text{antepasadoPropio}(\text{padrePropio}(x),y)) \\
\text{ver}(x,y) &= \text{antepasado}(x,y) \wedge \neg(\exists a,b,c \in E : \\
&\quad (\text{antepasado}(y,a) \wedge \text{antepasado}(y,b) \wedge c \in \text{padres}(x) \wedge c \in \text{padres}(b))) \wedge \\
&\quad \text{creador}(a) = \text{creador}(b) = \text{creador}(c) \\
\text{verFuertemente}(x,y) &= \text{ver}(x,y) \wedge (\exists S \in 2^E : (|S| = m) \wedge (z \in S \iff (\text{ver}(x,z) \wedge \text{ver}(z,y)))) \\
\text{rondadePadre}(x) &= \begin{cases} 0 & \text{si } x = \text{primero} \\ \max_{y \in \text{padres}(x)} \text{ronda}(y) & \text{de otra manera} \end{cases} \\
\text{testigo}(x) &= \exists S \in 2^E : (|S| = m \wedge \\
&\quad (\forall y \in S : (\text{ronda}(y) = \text{rondadePadre}(x) \wedge \text{verFuertemente}(x,y)))) \\
\text{ronda}(x) &= \begin{cases} 1 + \text{rondadePadre}(x) & \text{si testigo}(x) \\ \text{rondadePadre}(x) & \text{de otra manera} \end{cases} \\
\text{rondaDif.}(x,y) &= \text{ronda}(x) - \text{ronda}(y) \\
\text{votos}(x,y,v) &= |\{z \in \mathrm{E} \mid \text{ver}(x,y) \wedge \text{rondaDif.}(x,z) = 1 \wedge \\
&\quad \text{verFuertemente}(x,z) \wedge \text{voto}(z,y) = v\}| \\
\text{fracciónVoto}(x,y) &= \text{votos}(x, \text{verdadero}) / (\text{votos}(x, \text{verdadero}) + \text{votos}(x, \text{falso})) \\
\text{voto}(x,y) &= \begin{cases} \text{ver}(x,y) & \text{si rondaDif.}(x,y) = 1 \\ (\text{fracciónVoto}(x,y) >= 1/2) & \text{si } (\text{rondaDif.}(x,y) \bmod 5 \neq 1) \vee \\ & \quad |\text{fracciónVoto}(x,y) - 1/2| > 1/6 \\ (1 = \mathrm{LSB}(\text{firma}(x))) & \text{de otra manera} \end{cases} \\
\text{decidir}(x,y) &= \text{voto}(x,y) \wedge (\text{rondaDif.}(x,y) \bmod 5 \neq 1) \wedge (\text{fracciónVoto}(x,y) > 2/3) \\
\text{todosFamosos}(r) &= \{x \in \mathrm{E} \mid \text{famoso}(x) \wedge \text{ronda}(x) = 4\} \\
\text{famoso}(x) &= \text{testigo}(x) \wedge \exists y \in E : \text{decidir}(y,x) \\
\text{rondaRecibida}(x) &= \min_{r \in \mathbb{N}} (|\{y \in E \mid \text{ronda}(y) = r \wedge \text{famoso}(y) \wedge \text{ver}(y,x)\}| \; / \\
&\quad |\{y \in E \mid \text{ronda}(y) = r \wedge \text{famoso}(y)\}| >= 1/2) \\
\text{tiempoRecibido}(x) &= \text{mediana}(\{\text{tiempo}(y) | y \in E \wedge \text{ver}(y,x) \wedge \\
&\quad (\exists z \in E : \text{ronda}(z) = \text{rondaRecibida}(x) \wedge \text{antepasadoPropio}(z,y\}) \wedge \\
&\quad \neg(\exists w \in E : \text{antepasadoPropio}(y,w) \wedge \text{ver}(w,x))\})
\end{aligned}
$$

FIG. 16b

Un evento es una tupla e = {d, h, t, c, s} donde:

$d$ ≡ datos(e) ≡ los datos de "carga útil", que pueden incluir transacciones.
$h$ ≡ direccionamientos(e) ≡ una lista de direccionamientos de los padres del evento, padre propio primero.
$t$ ≡ tiempo(e) ≡ datos reivindicados del creador y hora de creación del evento.
$i$ ≡ creador(e) ≡ número de ID del creador.
$s$ ≡ $sig(e)$ ≡ firma digital del creador de {d, h, t, c}.
$n$ ≡ el número de miembros en la población
$c$ ≡ frecuencia de rondas de monedas (por ejemplo, c = 6)
$E$ ≡ (el conjunto de todos los eventos) $\cup \{\varnothing\}$
$\mathbb{T}$ ≡ conjunto de todos los posibles pares (hora, fecha)
$\mathbb{B}$ ≡ {verdadero, falso}
$\mathbb{N}$ ≡ $\{0, 1, 2, ...\}$

padres : $E \to 2^E$
padrePropio : $E \to E$
antepasado : $E \times E \to \mathbb{B}$
antepasadoPropio : $E \times E \to \mathbb{B}$
ver : $E \times E \to \mathbb{B}$
verFuertemente : $E \times E \to \mathbb{B}$
rondadePadre : $E \to \mathbb{N}$
rondaInc : $E \to \mathbb{B}$
ronda : $E \to \mathbb{N}$
testigo : $E \to \mathbb{B}$
rondaDif. : $E \times E \to \mathbb{I}$
votos : $E \times E \times \mathbb{B} \to \mathbb{N}$
verdaderaFrac. : $E \times E \to \mathbb{R}$
decidir : $E \times E \to \mathbb{B}$
voto : $E \times E \to \mathbb{B}$
famoso : $E \to \mathbb{B}$
rondaRecibida : $E \to \mathbb{N}$
tiempoRecibido : $E \to \mathbb{T}$

FIG. 17a

$$\text{padres}(x) = \text{conjunto de padres de evento x.}$$

$$\text{padrePropio}(x) = \text{el padre propio de evento x, o } \varnothing \text{ si ninguno}$$

$$\text{antepasado}(x,y) = (x \neq \varnothing) \wedge ((x = y) \vee (\exists z \in \text{padres}(x) : \text{antepasado}(z,y)))$$

$$\text{antepasadoPropio}(x,y) = (x \neq \varnothing) \wedge ((x = y) \vee \text{antepasadoPropio}(\text{antepasadoPropio}(x), y))$$

$$\text{ver}(x,y) = \text{antepasado}(x,y) \wedge \neg(\exists a, b \in E : \text{creador}(y) = \text{creador}(a) = \text{creador}(b) \wedge \text{antepasado}(x,a) \wedge \text{antepasado}(x,b) \wedge \neg\, \text{antepasadoPropio}(a,b) \wedge \neg\, \text{antepasadoPropio}(b,a))$$

$$\text{verFuertemente}(x,y) = \text{ver}(x,y) \wedge (\exists S \in 2^E : (|S| > 2n/3) \wedge (z \in S \iff (\text{ver}(x,z) \wedge \text{ver}(z,y))))$$

$$\text{rondadePadre}(x) = \text{máx}(\{0\} \cup \{\text{ronda}(y) | y \in \text{padres}(x)\})$$

$$\text{ronda}\,\text{Inc}(x) = \exists S \in 2^E : (|S| > 2n/3 \wedge (\forall y \in S : (\text{ronda}(y) = \text{rondadePadre}(x) \wedge \text{verFuertemente}(x,y))))$$

$$\text{ronda}(x) = \text{rondadePadre}(x) + \begin{cases} 1 & \text{si ronda}\,\text{Inc}(x) \\ 0 & \text{de otra manera} \end{cases}$$

$$\text{testigo}(x) = (\text{padrePropio}(x) = \varnothing) \vee (\text{ronda}(x) > \text{ronda}(\text{padrePropio}(x)))$$

$$\text{rondaDif.}(x,y) = \text{ronda}(x) - \text{ronda}(y)$$

$$\text{votos}(x,y,v) = |\{z \in \text{E} \mid \text{rondaDif.}(x,z) = 1 \wedge \text{verFuertemente}(x,z) \wedge \text{voto}(z,y) = v\}|$$

$$\text{frac.Verdadera}(x,y) = \frac{\text{votos}(x,y,\text{verdadero})}{(\text{votos}(x,y,\text{verdadero}) + \text{votos}(x,y,\text{falso}))}$$

$$\text{decidir}(x,y) = (x \neq \varnothing) \wedge (\text{rondaDif.}(x,y) > 1) \wedge \text{decidir}(\text{padrePropio}(x), y) \vee (\text{testido}(x) \wedge (\text{rondaDif.}(x,y) \mod c \neq 1) \wedge \neg(\tfrac{1}{3} \leq \text{frac.Verdadera}(x,y) \leq \tfrac{2n}{3})))$$

$$\text{voto}(x,y) = \begin{cases} \text{voto}(\text{padrePropio}(x), y) & \text{si } (\neg\ \text{testigo}(x)) \vee \text{decidir}(\text{padrePropio}(x), y) \\ 1 = \text{bitMedio}(\text{signature}(x)) & \text{si } \text{testigo}(x) \\ & \quad \wedge \neg\ \text{decidir}(\text{padrePropio}(x), y) \\ & \quad \wedge (\text{rondaDif.}(x,y) \neq 1) \\ & \quad \wedge (\text{rondaDif.}(x,y) \mod c = 1) \\ \text{frac.Verdadera}(x,y) \geq \frac{1}{2} & \text{de otra manera} \end{cases}$$

$$\text{famoso}(x) = \text{testigo}(x) \wedge \exists y \in E : \text{decidir}(y,x) \wedge \text{voto}(y,x)$$

$$\text{rondaRecibida}(x) = \text{mín}_{r \in \mathbb{N}} \frac{|\{y \in E | (\text{ronda}(y) = r) \wedge \text{famoso}(y) \wedge \text{ver}(y,x)\}|}{|\{y \in E | (\text{ronda}(y) = r) \wedge \text{famoso}(y)\}|} \geq 1/2)$$

$$\text{tiempoRecibido}(x) = \text{mediana}(\{\text{tiempo}(y) | y \in E \wedge \text{ver}(y,x) \wedge (\exists z \in E : \text{ronda}(z) = \text{rondaRecibida}(x) \wedge \text{antepasadoPropio}(z,y)\} \wedge \neg(\exists w \in E : \text{antepasadoPropio}(y,w) \wedge \text{ver}(w,x))\})$$

FIG. 17b