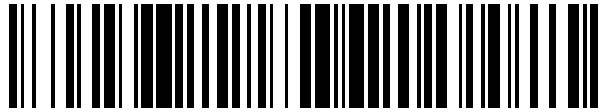


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 739 602**

51 Int. Cl.:

H04L 29/08 (2006.01)

H04L 12/751 (2013.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 86 Fecha de presentación y número de la solicitud internacional: **20.08.2014 PCT/GB2014/052556**
- 87 Fecha y número de publicación internacional: **26.02.2015 WO15025162**
- 96 Fecha de presentación y número de la solicitud europea: **20.08.2014 E 14772170 (8)**
- 97 Fecha y número de publicación de la concesión europea: **01.05.2019 EP 3036665**

54 Título: **Método para transmitir estructuras de datos de un ordenador a otro**

30 Prioridad:

22.08.2013 GB 201315028

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

03.02.2020

73 Titular/es:

**DEEP-SECURE LIMITED (100.0%)
1 Nimrod House, Sandy's Road
Malvern WR14 1JJ, GB**

72 Inventor/es:

WISEMAN, SIMON ROBERT

74 Agente/Representante:

AZNÁREZ URBIETA, Pablo

ES 2 739 602 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Método para transmitir estructuras de datos de un ordenador a otro

5 Campo de la invención

La presente invención se refiere a la transmisión de las estructuras de datos de gráficos acíclicos de un ordenador a otro a través de una red de comunicaciones.

10 Antecedentes

Los gráficos acíclicos pueden transmitirse de un ordenador a otro de forma descendente. Esto tiene la ventaja de que la aplicación del ordenador receptor puede comenzar a procesar el gráfico antes de que se haya recibido por completo.

15 El ordenador receptor creará nodos antes de los nodos a los que se refieren se hayan creado, por lo que cuando se crea un nodo es necesario actualizar los nodos a los que se refieren. Por lo tanto, el ordenador transmisor debe indicar al ordenador receptor la ubicación de los nodos que deben actualizarse al proporcionarles una referencia. Esto limita la forma en que se puede usar la memoria del ordenador receptor, que a su vez restringe la forma en que una aplicación puede ejecutarse en el ordenador receptor, hasta que la estructura de datos esté completa.

20 Es un objeto de realizaciones de la invención proporcionar un método de transferencia de una estructura de datos entre ordenadores que permite el uso eficiente de los recursos de memoria de ordenador y, en particular, permite una aplicación que se ejecuta en un ordenador receptor para utilizar los recursos de memoria al procesar una estructura de datos que se está recibiendo, sin necesidad de estrategias de asignación de memoria que resulten en una mala utilización.

25 El documento EP 1 244 017 A1 divulga un aparato de gestión de estructura de datos que gestiona una pluralidad de estructuras de datos mutuamente iguales almacenadas en una de una pluralidad de unidades de almacenamiento. Una unidad de almacenamiento de estructura de datos almacena una primera estructura de datos arbitraria, una unidad de modificación de datos modifica la primera estructura de datos, una unidad de extracción de datos modificada extrae solo los datos necesarios para realizar la primera estructura de datos en el lado de la fuente de modificación y una segunda estructura de datos en el la modificación del lado del objetivo es igual y genera los datos extraídos y modificados para reflejarlos en la segunda estructura de datos.

35 Sumario de la invención

40 Según la presente invención, se proporciona un método de transmisión de una estructura de datos del gráfico acíclico de un primer ordenador a un segundo ordenador que comprende las etapas de: transmitir una pluralidad de mensajes desde el primer ordenador al segundo ordenador, cada mensaje que contiene datos de un nodo de la estructura de datos de un gráfico acíclico y una o más referencias a ubicaciones de punteros secundarios en cualquier nodo principal transmitido previamente a ese nodo; almacenar los datos de cada nodo transmitido en el segundo ordenador; almacenar la dirección en la que se almacena cada nodo en la memoria del segundo ordenador en una tabla de direcciones; y almacenar un puntero secundario a cada nodo almacenado en el segundo ordenador en la ubicación del puntero secundario o en cada una de las ubicaciones a las que se hace referencia en el mensaje, en donde la o cada referencia comprende un índice en la tabla de direcciones.

50 Al crear una tabla de búsqueda para registrar la ubicación de los nodos que se han creado en el segundo ordenador, el primer ordenador puede usar índices en esta tabla para identificar qué nodos necesitan actualización, y así evitar la necesidad de que el primer ordenador sepa cómo el segundo ordenador está asignando nodos en la memoria.

55 El o cada referencia a una ubicación del puntero secundario puede comprender, además, un índice en una matriz de punteros secundario que pertenezca a un nodo. La referencia puede comprender el número de secuencia del nodo principal del nodo que se está transmitiendo. La referencia también puede comprender el índice en los secundarios del nodo principal al puntero al nodo que se está transmitiendo.

60 El ordenador receptor puede mantener una cola de direcciones de nodo que contiene la dirección de cada nodo que todavía tiene que ser completado. La cola puede comprender una lista de direcciones junto con un número de secuencia del nodo que está primero en la cola.

65 Una referencia se puede convertir en una dirección mediante el cálculo de una dirección en la cola de un número de secuencia comprendida en la referencia. La dirección de inicio del nodo y el tamaño de los datos de los nodos se pueden combinar para determinar la dirección del primer puntero secundario y se puede agregar un índice a los secundarios del nodo principal para obtener la dirección del puntero secundario al que se hace referencia.

La dirección de un nodo almacenada en la tabla puede ser retirada cuando el nodo se ha completado en el segundo

ordenador, lo que permite la ubicación de memoria en la que se almacena la ubicación de nodo para ser reutilizado.

5 La tabla de direcciones se puede representar como una lista de los marcos de la tabla que hace referencia una tabla de punteros. Las direcciones a los nodos incompletos pueden eliminarse de los marcos de la tabla a medida que los nodos se completan y los marcos de la tabla se reutilizan una vez que se quedan vacíos.

10 Como solo los nodos que necesitan actualización se almacenan en la tabla de búsqueda, esto reduce el tamaño de memoria necesario. Dado que los nodos se actualizan en el mismo orden en que se crean, se puede utilizar una cola para implementar la tabla de búsqueda.

15 Una aplicación que se ejecuta en el segundo equipo puede comenzar el procesamiento de la estructura de datos antes de que haya sido completamente recibida por el segundo equipo. El procesamiento de la estructura de datos puede suspenderse cuando la aplicación busca procesar una parte incompleta de la estructura de datos y reanudarse cuando esa parte de la estructura de datos se ha completado. Esto podría lograrse si la aplicación establece una variable cuando trata de procesar una parte incompleta de la estructura de datos. Cuando un nodo o nodos se crean y se actualizan los punteros para completar la parte relevante de la estructura de datos, la ubicación a la que se hace referencia puede compararse con la variable para determinar si la aplicación puede continuar.

20 El segundo equipo puede construir nodos en su memoria en respuesta a los mensajes recibidos de tal manera que las referencias de puntero secundario que no han sido completados pueden distinguirse del resto del nodo. Esto permite que el segundo ordenador valide las referencias a las ubicaciones de punteros secundarios contenidas en los mensajes al verificar si la ubicación a la que hacen referencia es un puntero secundario que no se ha completado. Se puede usar un bit de etiqueta asociado con cada ubicación de datos para distinguir entre una ubicación que contiene un puntero secundario que aún no se ha completado y una ubicación que contiene un valor de datos o un puntero secundario completado. Alternativamente, o adicionalmente, un puntero secundario que aún no se ha completado puede distinguirse de una ubicación que contiene un valor de datos o un puntero secundario completado al contener un valor especial que nunca se utiliza como valor de datos o puntero secundario completado. O al puntero se le puede asignar un valor NULO.

30 **Breve descripción de los dibujos**

Para que la invención pueda entenderse más claramente, se describirán ahora realizaciones preferidas de la invención, solamente a modo de ejemplo, con referencia a los dibujos adjuntos, en los que:

- 35 La figura 1 muestra un gráfico acíclico y cómo se puede transmitir el gráfico de manera descendente;
- La figura 2 muestra cómo se pueden validar las ubicaciones de punteros que necesitan actualizarse;
- La figura 3 muestra cómo una aplicación puede procesar un gráfico en paralelo con su construcción;
- La figura 4 muestra un método por el cual una aplicación puede sincronizarse con la construcción del gráfico;
- 40 La figura 5 muestra cómo se puede hacer referencia a las ubicaciones de punteros sin que ordenador transmisor tenga conocimiento de cómo el ordenador receptor asigna nodos en la memoria;
- La figura 6 muestra que se puede usar una pila para rastrear nodos incompletos;
- La figura 7 muestra un ejemplo más grande de cómo se puede usar una pila para rastrear nodos incompletos;
- La figura 8 muestra cómo una tabla de dos niveles puede contener la pila de nodos incompletos de manera eficiente;
- 45 La figura 9 muestra procesos realizados mediante un ordenador transmisor que implementa un método de acuerdo con una realización de la invención.
- La figura 10 muestra un ejemplo de datos transmitidos por un ordenador que realiza los procesos de la figura 9;
- La figura 11 muestra los procesos realizados por un ordenador receptor que implementa un método de acuerdo con la realización de la invención mostrada en las figuras 9 y 10;
- 50 La figura 12 muestra procesos adicionales realizados por un ordenador receptor que implementa un método de acuerdo con la realización de la invención de las figuras 9 a 11;
- La figura 13 muestra un ejemplo del uso de tablas de nivel superior y segundo nivel por parte de un ordenador receptor que implementa un método de acuerdo con la realización de la invención de las figuras 9 a 12; y
- 55 La figura 14 muestra procesos adicionales realizados por un ordenador receptor que implementa un método de acuerdo con la realización de la invención de las figuras 9 a 13.

Descripción detallada de la invención

60 Un gráfico acíclico se puede transmitir de una manera de arriba hacia abajo mediante el envío de un nodo antes de sus nodos secundarios. Un nodo se envía como un mensaje que contiene los datos del nodo y un recuento del número de nodos secundarios. La figura 1 muestra cómo se puede transmitir el gráfico 101 de forma descendente como una serie de mensajes 111. Los nodos recibidos se construyen en un almacén de nodos 112. Cuando se construye un nodo, sus punteros secundarios se establecen en el valor del puntero NULO. El mensaje que construye un nodo contiene referencias a todos los punteros en los nodos recibidos previamente que deben completarse con un puntero al nuevo nodo.

Si las referencias a las ubicaciones de puntero dadas en un mensaje no son válidas, el receptor construirá un gráfico no válido. Esto puede dar lugar a errores graves en la aplicación que procesa el gráfico, por lo que debe evitarse.

5 Para permitir que las referencias sean validadas, posiciones de memoria que contienen punteros NULO se pueden distinguir de los lugares que contienen punteros completados o valores de datos.

En la realización ilustrada en la figura 2a las posiciones de memoria del almacén de nodo 202 se marcan con bits de etiqueta 201. Estos se pueden configurar para indicar que la ubicación de la memoria tiene un valor NULO.

10 Alternativamente, en la realización ilustrada en la figura 2B los punteros NULO se establecen en un valor especial 212 que nunca se usa como un valor de puntero 211 o datos 213.

15 Por lo tanto, si una referencia a una ubicación del puntero refiere a una ubicación que no tiene un conjunto de etiquetas asociado para indicar que la ubicación tiene un valor NULO, o no se establece en un valor especial, el ordenador receptor sabe que la ubicación no es válida, y el mensaje puede ser rechazado. Esto permite que al menos algunos mensajes inválidos sean rechazados.

20 Una ventaja de un ordenador receptor que construye un gráfico de una forma de arriba hacia abajo es que la aplicación en el ordenador receptor puede comenzar a procesar el gráfico antes de que se haya recibido en su totalidad. La figura 3 muestra un gráfico 301 que se ha recibido parcialmente, con los nodos 302 y 303 recibidos y el nodo 304 aún por recibir.

25 En este punto, el almacén de nodo contiene una representación 311 de nodo 302 incluye un puntero 312 a su secundario 303 que ha sido recibido y un puntero NULO 313 que será ajustado para que apunte a su secundario 304 una vez que se ha recibido.

30 Si durante el procesamiento del nodo recibido 302, almacenado como 311, la aplicación tiene que seguir el puntero al nodo 303 puede hacerlo como el puntero se ha completado 312. Sin embargo, si necesita seguir el puntero al nodo 304, debe esperar porque el puntero aún no se ha completado 313. Una vez que el receptor haya recibido y construido, el nodo 304 puntero 313 se completará de modo que se refiera al nodo almacenado y la aplicación pueda continuar con el procesamiento.

35 La figura 4 ilustra el método mediante el cual el generador de gráficos y la aplicación de procesamiento de árbol del ordenador receptor cooperan para permitir que la aplicación proceda con su procesamiento en paralelo con el gráfico que se está construyendo. Se utiliza una variable compartida por los dos procesos, denominada AppLocation en los diagramas de flujo, para sincronizar los dos procesos cuando sea necesario. La aplicación establece la variable cuando encuentra que debe seguir un puntero NULO. Una vez configurada la variable, la aplicación espera una señal del proceso del generador de árboles. Cuando el generador de gráficos crea un nodo, completa todos los punteros necesarios a ese nodo para completar el gráfico. Una vez que se actualiza el puntero, el constructor verifica la ubicación a la que se hace referencia con la variable AppLocation y, si es igual, señala a la aplicación que indique que el puntero ahora se ha completado y se puede leer para continuar el procesamiento.

45 Los mensajes enviados por el ordenador emisor incluyen referencias a ubicaciones del puntero secundario en los nodos previamente enviados. Estas referencias podrían ser la dirección de la ubicación del puntero en la memoria del ordenador receptor. Esto requeriría que el ordenador emisor comprenda y rastree cómo el ordenador receptor asigna su memoria. Esto es posible si la memoria solo se utiliza para crear el gráfico recibido, aunque no es deseable ya que significa que la implementación del remitente está vinculada a la implementación del receptor. Sin embargo, el uso exclusivo de la memoria por parte del constructor de gráficos del ordenador receptor es una seria limitación que impide que la aplicación utilice la memoria mientras se está construyendo el gráfico o requiere una asignación previa de recursos de memoria que da como resultado una mala utilización de la memoria. Una alternativa es usar el número de índice de nodo en las referencias, pero el ordenador receptor tendría que buscar en el almacén del nodo para ubicar el nodo, lo que reduce el rendimiento y lo hace impredecible.

55 La figura 5 muestra cómo se pueden implementar las referencias como índices sin buscar mediante el uso de una tabla de punteros de nodos que se construye en el ordenador receptor a medida que se crean los nodos. El almacén de nodos 501 contiene nodos que se han construido hasta ahora y esto crece hacia arriba desde el inicio de la memoria. La tabla de direcciones de nodos 502 contiene punteros a los nodos construidos hasta ahora y esto crece hacia abajo desde el final de la memoria, lo que proporciona una utilización óptima de la memoria. Una referencia 510 comprende dos índices. El primer índice 511 indexa la tabla de direcciones de nodo para encontrar la dirección de inicio del nodo al que se hace referencia. El segundo índice 512 es un índice dentro de la lista de punteros de nodo secundario de este nodo a la ubicación del puntero al que se hace referencia.

65 El ordenador receptor mantiene una tabla de punteros a nodos que han sido asignados y la dirección del nodo que contiene un puntero de referencia se obtiene mediante el uso de índice de nodo de la referencia para seleccionar la entrada de esta tabla. La tabla de nodos de los nodos se indexa por el índice secundario de la referencia para

seleccionar el puntero al que se hace referencia. En el ejemplo ilustrado en la figura 5, la referencia 510 es una referencia al segundo secundario del nodo B. 511 es el índice del nodo B (= 2) y 512 es el índice del segundo secundario (= 2) del nodo B.

5 El uso de la tabla de direcciones de nodos descrita anteriormente permite que la aplicación use el almacén de nodos como un almacén de almacenamiento dinámico, asignando bloques de memoria según sea necesario, ya que ordenador transmisor no necesita conocer la ubicación real de los nodos almacenados. Sin embargo, las aplicaciones generalmente requieren una pila además de un almacén de almacenamiento dinámico y, dado que la tabla de direcciones de nodos usa el otro extremo de la memoria, no hay ningún lugar para colocar la pila de aplicaciones sin la asignación previa de grandes cantidades de almacenamiento que provoque una mala utilización de la memoria.

15 Al estudiar la forma en que se utiliza el almacén de direcciones de nodo, se hace evidente que no es necesario mantener las direcciones de todos los nodos en él. Solo deben almacenarse los nodos que tienen algunos punteros secundarios a nodos que aún no se han creado. Además, con el ordenador transmisor enviando los nodos de forma descendente, los nodos se completan de manera descendente, de modo que la tabla de direcciones de los nodos funciona como una cola. Si se agregan nuevos nodos a un extremo de la cola, los nodos se completan en el orden en que están en cola y, por lo tanto, los nodos se pueden eliminar del otro extremo de la cola una vez que se completan. Como resultado, se necesita mucha menos memoria para la tabla de direcciones de nodos de lo que parece a primera vista.

20 Esto se demuestra en la figura 6. La figura 6a muestra un árbol 601 parcialmente completo recibido junto con la tabla de direcciones de nodo 602. La tabla de direcciones de nodo 602 solo contiene las direcciones de los nodos incompletos 603 y 604 del árbol parcialmente completo. Una vez que se recibe y construye el siguiente nodo enviado 605, el árbol 611 parcialmente completo, que se muestra en la figura 6b, tiene la tabla de direcciones de nodo correspondiente 612. Ahora contiene la dirección del nodo 605 incompleto recién recibido, pero ya no contiene la dirección del nodo 603, ya que ahora está completo.

30 Un ejemplo más grande se muestra en la figura 7. Esto muestra varias etapas en la construcción de una estructura de gráfico por parte de un ordenador receptor, con la cola de nodos de nodos incompletos mostrada en una caja sobre cada gráfico. Nuevamente, la tabla de direcciones solo necesita contener las direcciones de esos nodos en la cola, y que están parcialmente construidos.

35 El ordenador de transmisión puede mantener fácilmente un seguimiento del número de referencias incompletas en el ordenador receptor, por lo que puede utilizar un índice en la cola de nodo en sus referencias.

40 La eliminación de entradas de la tabla de direcciones de nodo cuando se completan los nodos significa que se requiere menos memoria, pero una implementación sencilla copiaría las direcciones de nodo restantes hasta el final de la memoria para recuperar el espacio. Esta copia lleva tiempo, lo que reduce el rendimiento y lo hace impredecible.

45 La copia puede evitarse implementando la cola como una lista con cada dirección asignada a una entrada de lista separada y cada entrada de lista que contenga la dirección de la siguiente entrada. Sin embargo, con este método el rendimiento se ve afectado por la necesidad de buscar en la lista para encontrar una entrada en particular.

50 Si una aplicación que se ejecuta en el ordenador receptor debe procesar el árbol en paralelo con su construcción, también requerirá algo de memoria. La aplicación podría usar el mismo almacén de nodos que el almacenamiento de almacenamiento dinámico, pero no hay espacio para una pila, ya que el receptor está utilizando ambos extremos de la memoria, por lo que debe asignarse suficiente espacio de pila para manejar el peor de los casos, lo que lleva a una mala utilización de la memoria.

55 La cola de nodos podría asignarse en el almacén de nodos, dejando el otro extremo de la memoria para que la aplicación lo use como una pila, pero tendría que ser un tamaño fijo. Esto limita el ancho del árbol que se puede transmitir. Se podría asignar una lista de marcos de cola de nodos, pero luego se requeriría una búsqueda para ubicar una entrada en la tabla.

60 Un enfoque que no afecte indebidamente el rendimiento es crear una estructura de dos niveles en el que la cola se divide en una lista de bloques de direcciones, con una tabla de nivel superior que contiene las direcciones de cada bloque de la lista. Esto se muestra en la figura 8. Los bloques son de un tamaño fijo. Por lo tanto, el índice de una entrada se convierte fácilmente en un índice en la tabla de nivel superior y un índice en el bloque al que se hace referencia en la tabla de nivel superior, dividiendo por el tamaño de los marcos para obtener un índice en la tabla de nivel superior de forma que el resto después de la división produce un índice en el marco seleccionado.

65 El número máximo de nodos incompletos que pueden almacenarse es ahora el tamaño de la tabla de nivel superior multiplicado por el tamaño de los marcos. Esto es efectivamente una restricción en el ancho de los árboles que se pueden transmitir. Los tamaños de tabla más grandes significan menos restricción, pero reducen la utilización de

5 memoria. Sin embargo, en la práctica, los tamaños se pueden elegir para imponer pocas restricciones sin una gran reducción en la utilización de la memoria. Por ejemplo, si la tabla de nivel superior puede contener 1000 entradas y los marcos de la tabla pueden contener 1000 punteros de nodo, el ancho del árbol se restringe a 1.000.000 nodos, lo que se corresponde con el tamaño de la memoria que probablemente esté disponible para almacenar el árbol y, por lo tanto, es efectivamente sin restricción.

Una vez que todos los nodos se han eliminado de un marco, se pueden reutilizar cuando se necesita otro marco.

10 En el ejemplo ilustrado en la figura 8, en la figura 8a la tabla de nivel superior 810 contiene un solo puntero a un marco de la tabla 811 que contiene punteros a nodos incompleta A y B. Después de la recepción de nodo G, la situación es tal como se ilustra en la figura 8b. La tabla de nivel superior 820 ahora contiene punteros a dos marcos de tabla, un primer marco de tabla 821 que contiene punteros al nodo B y un segundo marco de tabla 822 que contiene punteros al nodo G.

15 A medida que se agregan entradas a la cola, se llena el último bloque. Una vez completado se asigna un nuevo bloque al final. A medida que se eliminan las entradas de la cola, se vacía el primer bloque. Una vez completamente vacío, el primer bloque se elimina y su almacenamiento se puede reutilizar.

20 Las figuras 9 a 14 ilustran una realización adicional de la invención que emplea una técnica de arriba hacia abajo para transmitir datos desde un primer ordenador de transmisión a un segundo ordenador receptor. El método es implementado por el software apropiado que se ejecuta en los ordenadores respectivos.

La figura 9 ilustra las etapas realizadas por el ordenador transmisor para transmitir datos de manera descendente.

25 El proceso se ejecuta a lo largo de líneas similares a la ilustrada en la figura 1.

El transmisor utiliza una cola para seguir la pista de qué nodos deben ser enviados. También marca los nodos con un número de secuencia y una lista de nodos principales que hacen referencia a ellos junto con una indicación de qué secundario de ese nodo principal son.

30 El transmisor comienza poniendo el nodo raíz en la cola.

35 Mientras que la cola no está vacía, el transmisor quita el primer nodo de la cola y lo procesa. Para cada nodo procesado, el transmisor le asigna un número de secuencia y luego visita cada nodo secundario a su vez. Si el secundario no ha sido visitado, se agrega a la cola. Luego, el secundario se marca con el número de secuencia del nodo actual y el valor del índice del secundario actual. El transmisor puede determinar si un nodo ha sido visitado anteriormente porque aquellos que aún no han sido visitados no tienen un número de secuencia asignado.

40 Una vez que se han visitado todos los secundarios de un nodo, se envía un mensaje que describe el nodo. El mensaje contiene los datos del nodo y la cantidad de secundarios que tiene, lo que permite al receptor construir el nodo en la memoria. El mensaje también contiene la lista de marcas que se han agregado al nodo. Estas le indican al receptor qué nodos deben actualizarse para referirse al nuevo nodo.

45 La figura 10 muestra un ejemplo de transmisión del árbol ilustrado en la figura utilizando el proceso que se muestra en la figura 9. El árbol se muestra anotado con los números de secuencia que se asignan a los nodos y las marcas que están asociadas a ellos. Las marcas son un par de números, el primero es el número de secuencia del nodo principal y el segundo el índice en los secundarios de ese nodo al puntero al nodo marcado. Tenga en cuenta que el nodo D tiene dos primarios, es un subárbol común compartido de B y E, por lo que tiene dos marcas. El nodo raíz A no tiene primarios y, por lo tanto, no tiene marcas. La transmisión implica las siguientes etapas:

- 50
1. El nodo raíz A está en cola
 2. A se elimina de la cola y se le asigna el número de secuencia 1
 3. Los secundarios de A se visitan a su vez, B se pone en cola y se marca que es el primer secundario del nodo A
 - 55 4. E está en cola y marcado que es el segundo secundario de A
 5. Los datos del nodo A se envían, junto con una indicación de que tiene dos secundarios
 6. El siguiente nodo en la cola, B, se toma y se le asigna el número de secuencia 2
 7. Los secundarios de B se visitan a su vez, C se pone en cola y se marca como el primer secundario de B
 8. D está en cola y marcado como el segundo secundario de B
 - 60 9. Los datos del nodo B se envían, junto con una indicación de que tiene dos secundarios y que es el primer secundario de A
 10. El siguiente nodo en la cola, E, se toma y se le asigna el número de secuencia de 3
 11. Los secundarios de E son visitados a su vez. D ya ha sido visitado, por lo que no se agrega nuevamente a la cola, pero está marcado como el primer secundario de E además del segundo secundario de B.
 - 65 12. Los datos del nodo E se envían junto con una indicación de que tiene un secundario y que es el segundo secundario de A

13. Se toma el siguiente nodo en la cola, C, y se le asigna el número de secuencia 4.
14. Los secundarios de C son visitados a su vez, pero no hay ninguno. Así que los datos del nodo C se envían, junto con una indicación de que no tiene secundarios y que es el primer secundario de B
15. El siguiente nodo en la cola, D, se toma y se le asigna el número de secuencia 5.
16. Los secundarios de D son visitados a su vez, pero no hay ninguno. Así que los datos del nodo D se envían, junto con una indicación de que no tiene secundarios y que es el segundo secundario de B y el primer secundario de E

La cola ahora está vacía, por lo que la transmisión está completa

La figura 11 muestra las etapas realizadas por el ordenador receptor cuando recibe datos transmitidos por el ordenador transmisor para crear una copia del árbol que está transmitiendo el ordenador transmisor.

El receptor construye nodos en posiciones de memoria consecutivas. Recibe un mensaje que contiene los datos de un nodo y lo escribe en la siguiente ubicación libre en la memoria. A continuación, obtiene el número de nodos secundarios del mensaje y agrega un puntero NULO a los datos del nodo para cada uno.

A continuación, el receptor visita cada una de las marcas en el mensaje. Cada marca se refiere a uno de los punteros secundarios de un nodo creado anteriormente. La información de la marca se convierte a la dirección de memoria del puntero secundario mediante una técnica que se describe a continuación.

Después, el receptor comprueba si la posición direccionada contiene un puntero NULO. Si no es así, la marca no es válida, ya que los punteros NULO solo aparecen en la ubicación de un puntero secundario que aún no se ha establecido. Si la ubicación direccionada contiene un valor NULO, la dirección del nodo recién construido se escribe en esta ubicación. El receptor luego pasa a la siguiente marca. De este modo, al asignar punteros nulos a punteros secundarios no configurados, el ordenador receptor puede detectar ciertos errores en la transmisión de datos, en particular protege contra la actualización del puntero más de una vez.

La figura 12 muestra cómo el ordenador receptor puede convertir las marcas a una dirección de memoria, sin ninguna búsqueda y sin usar una gran cantidad de memoria.

El receptor mantiene una cola de direcciones de nodo que contiene la dirección de cada nodo que aún no se ha escrito completamente. Los nodos se almacenan en la cola en el orden en que se transmitirán, que es el mismo orden en que se completarán.

Lógicamente, la cola es una lista de direcciones de nodo junto con el número de secuencia del nodo que es primero en la cola. Una marca se convierte en una dirección tomando primero el número de secuencia del nodo de la marca y calculando qué dirección en la cola es la del nodo al que se hace referencia al restar el primer número de secuencia en la cola del número de secuencia de la marca para producir un desplazamiento en la cola a la dirección del nodo referenciado. A continuación, se examinan los datos del nodo para determinar su longitud. A medida que el primer puntero secundario sigue los datos del nodo, agregar la dirección de inicio del nodo y el tamaño de los datos del nodo juntos dan la dirección del primer puntero secundario. El índice secundario de la marca se agrega a esta dirección para obtener la dirección del puntero secundario al que se hace referencia.

La porción superior de la figura primera figura muestra tres nodos, el nodo raíz es rojo y este tiene dos secundarios: Azul y Verde. Azul tiene un secundario que aún no se ha recibido y Verde tiene dos secundarios que aún no se han recibido. La cola contiene las direcciones de los dos nodos incompletos. Cuando se recibe el siguiente nodo, se coloca después del nodo Verde y las marcas indican cuál de los punteros NULOS en los nodos Azul y Verde deben actualizarse para referirse al nuevo nodo.

La parte inferior de la figura muestra cómo la marca se convierte en la dirección del puntero de secundario.

El número de secuencia del nodo cuya dirección está primera en la cola se resta del número de secuencia del nodo de la marca para producir un índice en la cola a la dirección del nodo referenciado por la marca. La memoria en esta dirección se lee para determinar el tamaño de los datos de los nodos a los que se hace referencia. Este tamaño se agrega a la dirección de inicio del nodo y al índice de puntero secundario de la marca para producir la dirección del puntero secundario al que hace referencia la marca.

Para permitir el uso eficiente de la memoria, la cola se lleva a cabo por un nivel superior y tablas de segundo nivel. Estas se muestran en la figura 13.

La tabla de nivel superior contiene una secuencia de direcciones que se refieren a tablas de segundo nivel. Tiene una longitud fija que no se puede cambiar si se llena. La secuencia puede comenzar en cualquier lugar de la tabla y, si es necesario, se ajusta al principio de la tabla. Esto se implementa anotando la ubicación de la primera dirección en la secuencia.

Las tablas de segundo nivel contienen secuencias de direcciones de nodos. Esta secuencia no se completa, pero no es necesario que comience al principio de la tabla y no es necesario que complete la tabla.

5 Trabajando desde la parte superior izquierda en la figura 13, el primer diagrama muestra la cola que tiene un elemento en él, A. Hay una tabla de segundo nivel. La tabla de nivel superior contiene un puntero a la tabla de segundo nivel al principio. La tabla de segundo nivel contiene el puntero a A también al principio.

10 El segundo diagrama muestra el estado después se añade el punto B a la cola. El puntero a B se almacena en el siguiente lugar libre en la tabla del último segundo nivel.

El tercer diagrama muestra el estado después del primer puntero, para A, se retira desde la parte frontal de la cola. Esto se implementa incrementando el desplazamiento de la "primera dirección" asociado con el bloque de segundo nivel, de modo que no se trate de copiar la memoria.

15 El cuarto diagrama muestra un puntero a C agrega a la cola. El bloque de segundo nivel está lleno, por lo que se crea un nuevo bloque de segundo nivel para acomodar el nuevo puntero. La dirección de esto se agrega al bloque de nivel superior.

20 Pasando ahora a la segunda fila de diagramas en la figura y comenzando en el lado izquierdo, el quinto diagrama muestra que se agrega un puntero a D donde hay espacio para este último bloque de segundo nivel, por lo que se agrega allí.

25 Como se muestra en el sexto diagrama, cuando el puntero a B se elimina de la cola, el primero de los segundos bloques de nivel se vacía. Por lo tanto, este bloque se descarta y el puntero se elimina del bloque de nivel superior incrementando la "primera dirección". Si esto llega al final del bloque de nivel superior, se ajusta al inicio del bloque.

30 En el séptimo diagrama, cuando se añade un puntero a E a la cola no hay lugar en el último bloque segundo nivel, por lo que se asigna un nuevo bloque de segundo nivel. Pero cuando el puntero al nuevo bloque de segundo nivel se agrega al nivel superior, no hay espacio para que vaya al final, por lo que se enrolla y se coloca al principio.

La figura 14 muestra la lógica utilizada para asignar un índice en la secuencia de nodos a una dirección de nodo.

35 El número de secuencia del primer nodo en la cola se resta del índice para dar un índice relativo. Este índice relativo se divide por el tamaño de las tablas de segundo nivel para producir un número entero y el resto. El número entero es un índice en la secuencia de las tablas de segundo nivel y el resto es un desplazamiento dentro de la tabla seleccionada. El resto es el desplazamiento dentro del bloque de segundo nivel a la dirección del nodo.

40 Se añade el número entero con el índice de la primera dirección en la tabla de nivel superior para dar un desplazamiento en la tabla de nivel superior, pero esto es valor, entonces se envuelve alrededor mediante el uso de una función de módulo que tiene el resto del valor cuando se divide por la longitud de la tabla de nivel superior. Esto da un desplazamiento en la tabla de primer nivel a la ubicación de la tabla de segundo nivel relevante. La tabla de nivel superior se indexa con este valor para producir la dirección de la tabla de segundo nivel.

45 A continuación, se accede a la segunda tabla de nivel para obtener el índice de la primera dirección dentro de ella. Este valor se agrega a la dirección de la tabla de segundo nivel y al resto para generar la dirección de nodo.

Las realizaciones anteriores se describen solamente a modo de ejemplo. Muchas variaciones son posibles sin apartarse de la invención, como se define en las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un método para transmitir una estructura de datos de gráficos acíclicos (101, 301, 601, 611) desde un primer ordenador a un segundo ordenador que comprende las etapas de:
- 5 transmitiendo una pluralidad de mensajes (111, 510) desde el primer ordenador al segundo ordenador, cada mensaje contiene datos de un nodo de la estructura de datos del gráfico acíclico y una o más referencias (510) a ubicaciones de punteros secundarios en cualquier nodo principal transmitido previamente a ese nodo almacenar los datos de cada nodo transmitido en la memoria (112, 210, 310, 501) del segundo ordenador; almacenar la dirección en la que se almacena cada nodo en la memoria del segundo ordenador en una tabla de direcciones (502, 602, 612); y
- 10 almacenar un puntero secundario a cada nodo almacenado en el segundo ordenador en la ubicación del puntero secundario o en cada uno de los puntos a los que se hace referencia en el mensaje, en donde la o cada referencia a la ubicación del puntero secundario comprende un índice en la tabla de direcciones.
- 15 2. Un método según la reivindicación 1, en el que la o cada referencia (510) a una ubicación de puntero secundario comprende además un índice (512) en una matriz de punteros secundarios que pertenecen a un nodo.
3. Un método según cualquier reivindicación anterior, en el que la dirección de un nodo almacenado en la tabla se elimina cuando el nodo se ha completado en el segundo ordenador.
- 20 4. Un método según cualquier reivindicación anterior, en el que la tabla de direcciones se representa como una lista de cuadros de tabla (811, 821, 822) referenciada por una tabla (810, 820) de punteros.
5. Un método según la reivindicación 4, en el que la dirección de un nodo almacenado en los marcos de la tabla se elimina de los marcos de la tabla cuando los punteros a los nodos secundarios se han almacenado en todas las ubicaciones de punteros secundarios y los marcos de la tabla se reutilizan una vez que están vacíos.
- 25 6. Un método según cualquier reivindicación anterior, en el que una aplicación que se ejecuta en el segundo ordenador comienza a procesar la estructura de datos antes de que haya sido recibida completamente por el segundo ordenador.
- 30 7. Un método según la reivindicación 6, en el que el procesamiento de la estructura de datos se suspende cuando la aplicación busca procesar una parte incompleta de la estructura de datos, y se reanuda cuando se ha completado esa parte de la estructura de datos.
- 35 8. Un método según la reivindicación 7, en el que la aplicación establece una variable cuando trata de procesar una parte incompleta de la estructura de datos, y cuando se crean nodos y se actualizan los punteros, la ubicación a la que se hace referencia se compara con la variable para determinar si la aplicación puede continuar.
- 40 9. Un método según cualquier reivindicación anterior, en el que el segundo ordenador construye nodos en su memoria en respuesta a los mensajes recibidos de tal manera que las referencias de puntero secundario que no se han completado pueden distinguirse del resto del nodo.
- 45 10. Un método según la reivindicación 9, en el que el segundo ordenador valida las referencias a ubicaciones de punteros secundarios contenidas en los mensajes comprobando si la ubicación a la que hacen referencia es un puntero secundario que no se ha completado.
11. Un método según la reivindicación 9 o 10, en el que un bit de etiqueta (201) asociado con cada ubicación de datos se usa para distinguir entre una ubicación que contiene un puntero secundario que aún no se ha completado y una ubicación que contiene un valor de datos o un puntero secundario completado.
- 50 12. Un método según la reivindicación 9 o 10, en el que un puntero secundario que aún no se ha completado se distingue de una ubicación que contiene un valor de datos o un puntero secundario completado que contiene un valor especial (212) que nunca se usa como valor de datos o puntero secundario completado.
- 55 13. Un método según la reivindicación 9 o 10, en el que a un puntero secundario que aún no se ha completado se le asigna un valor NULO.

Figura 1

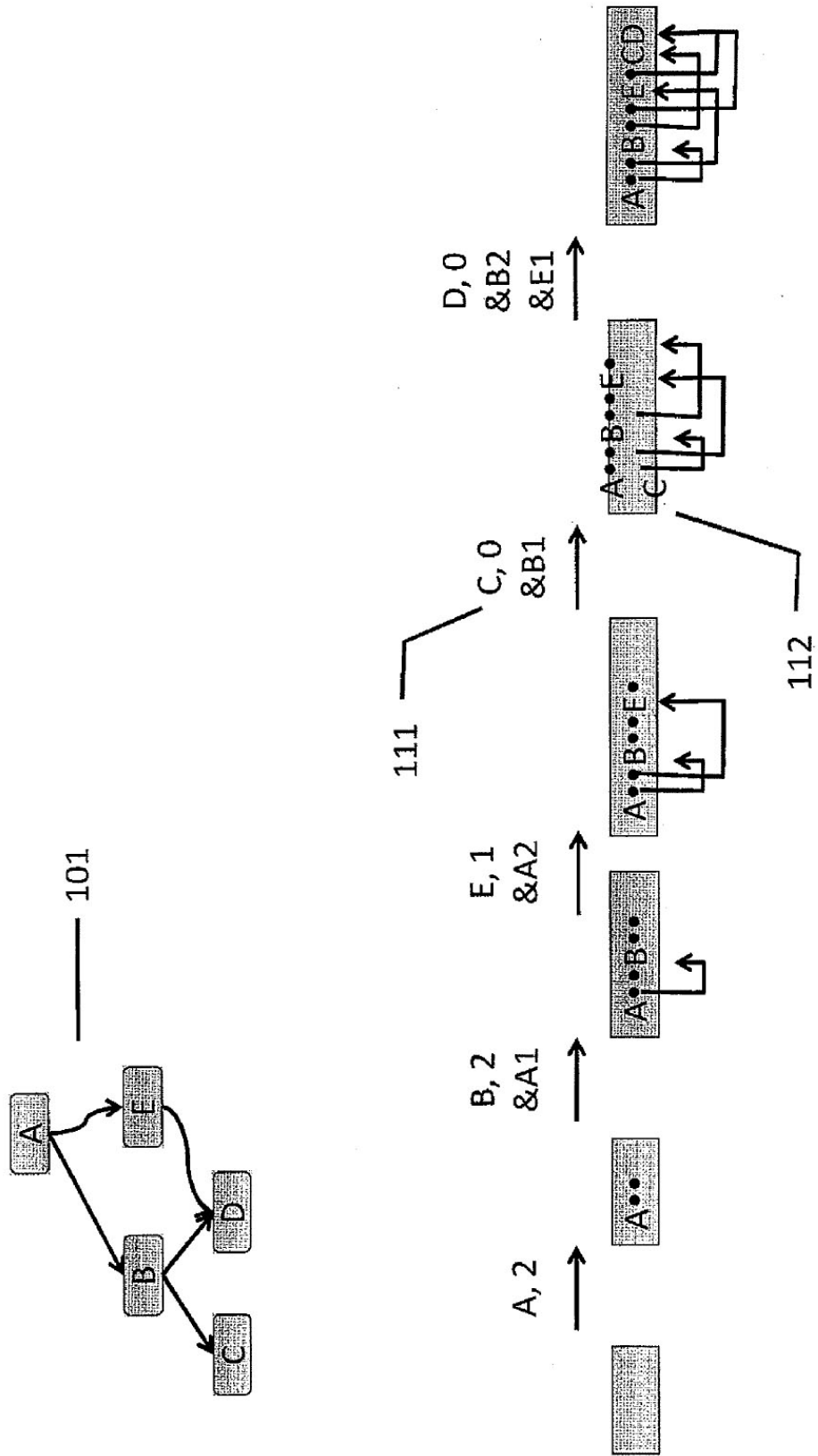


Figura 2

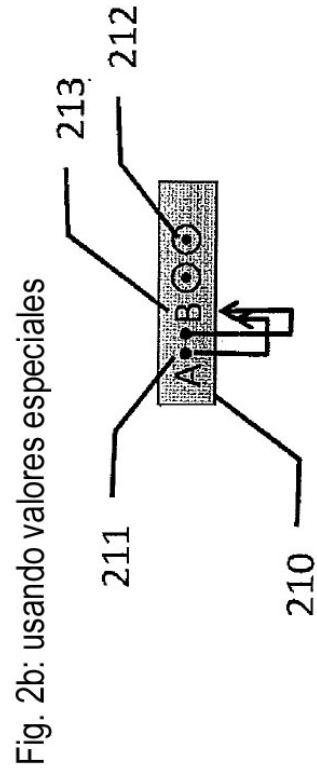
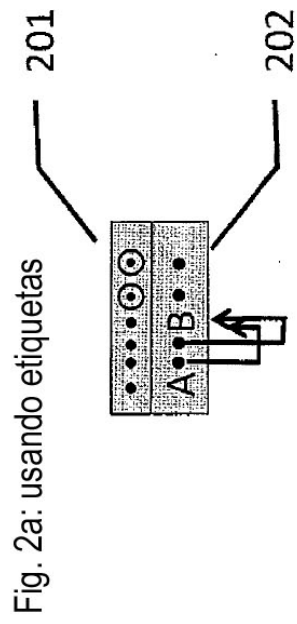


Figura 3

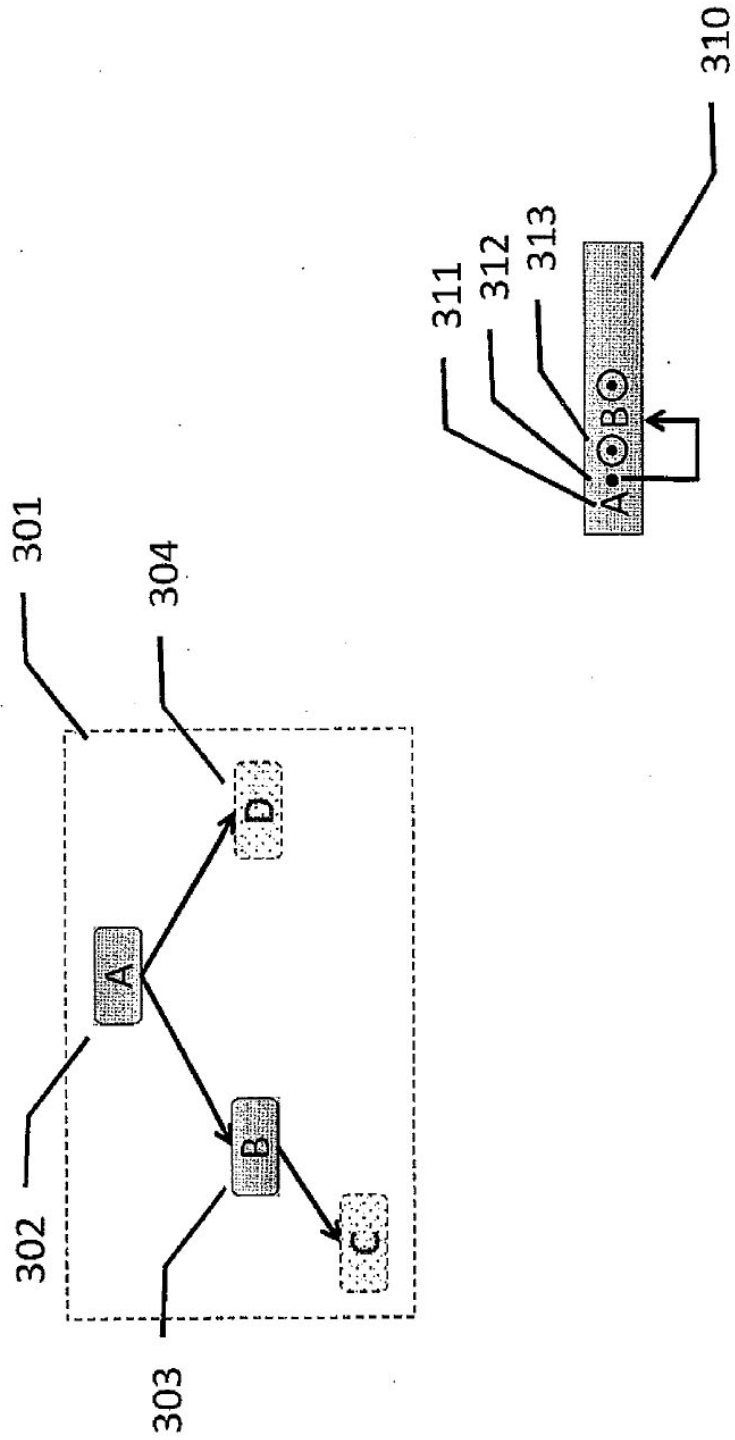


Figura 4

Fig. 4a: Constructor de gráficos

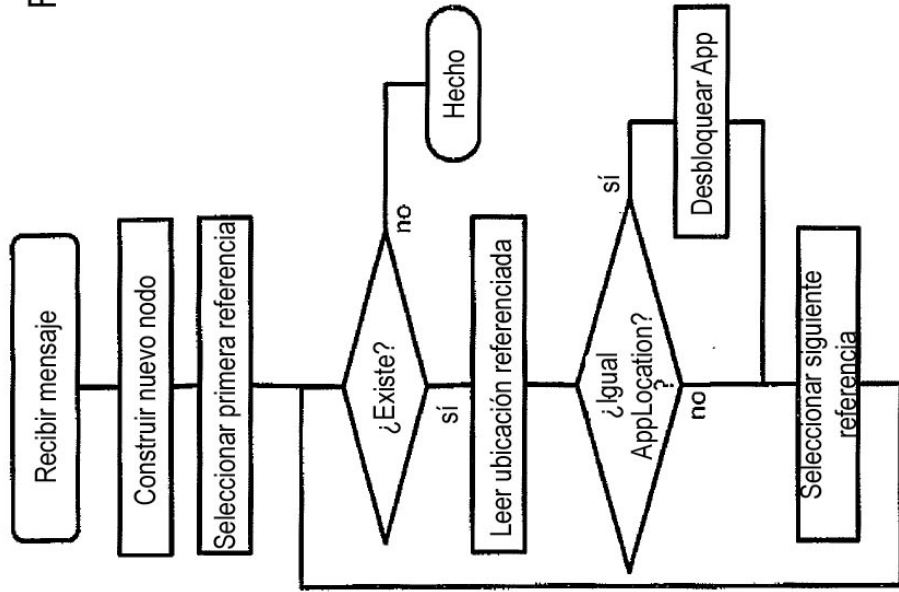


Fig. 4b: Aplicación

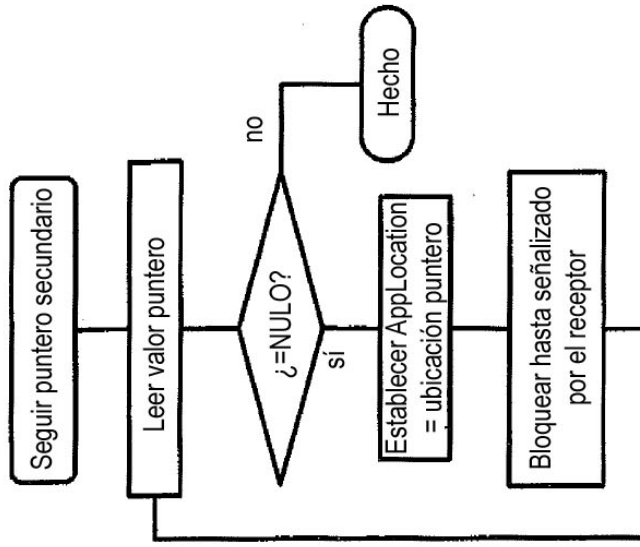


Figura 5

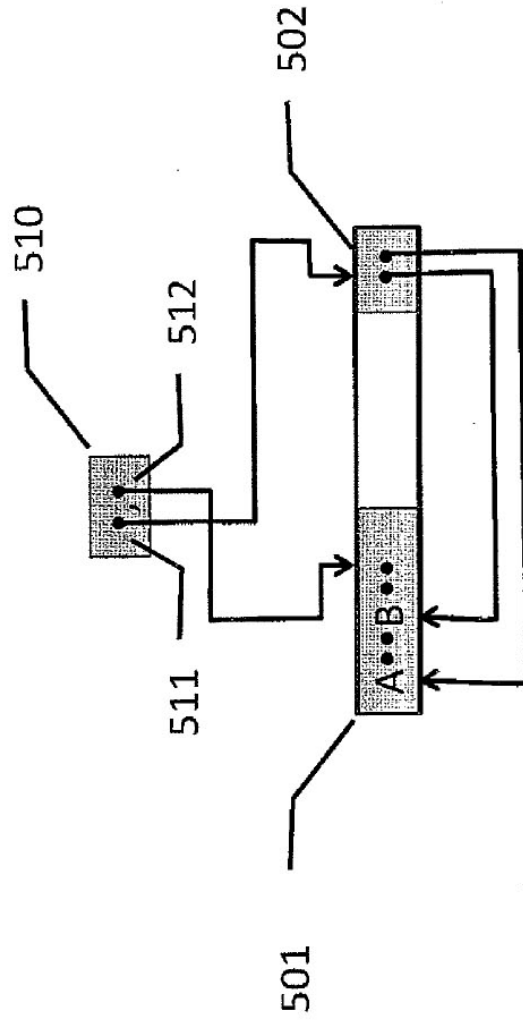


Figura 6

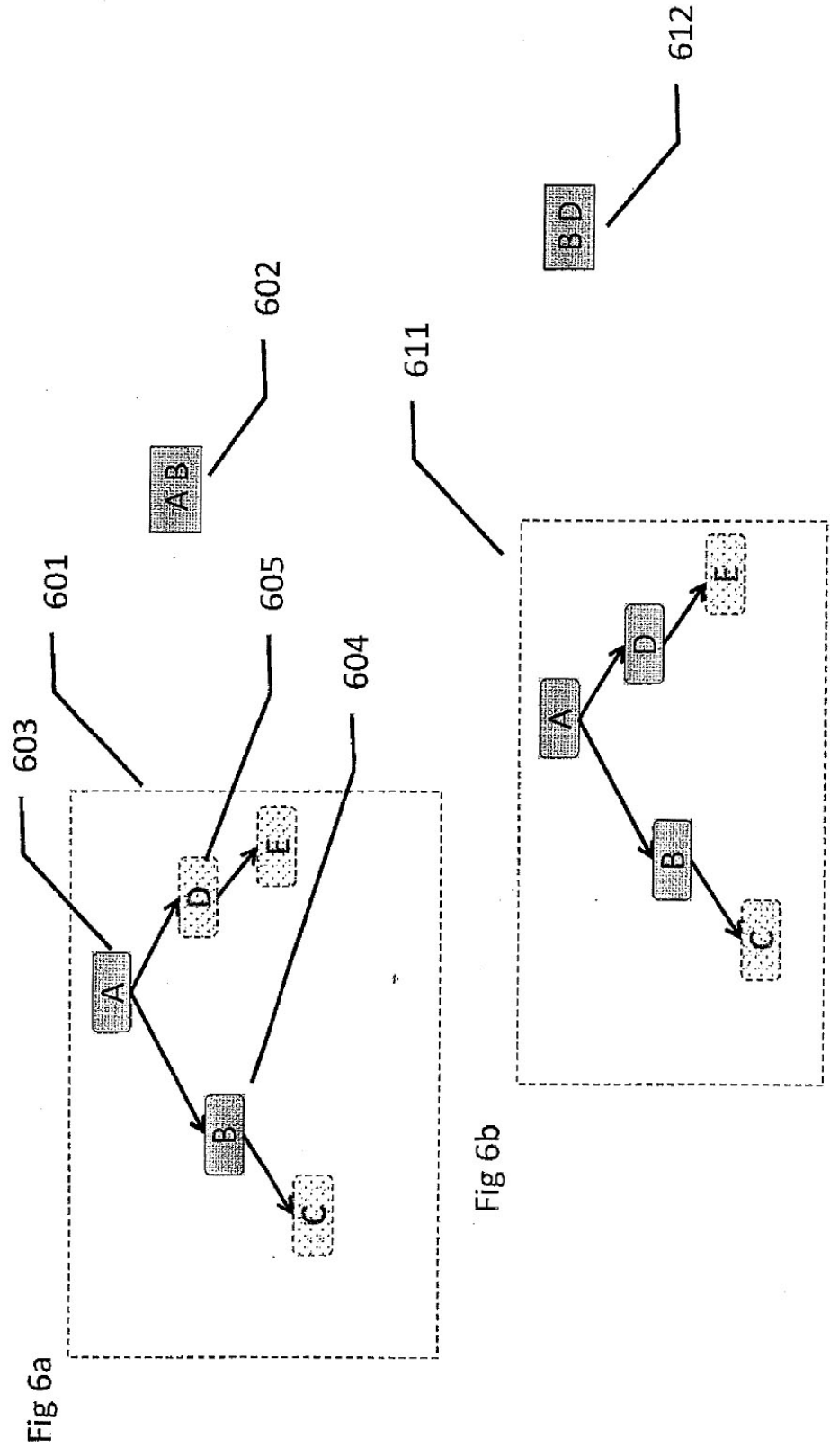


Fig 6a

Fig 6b

Figura 7

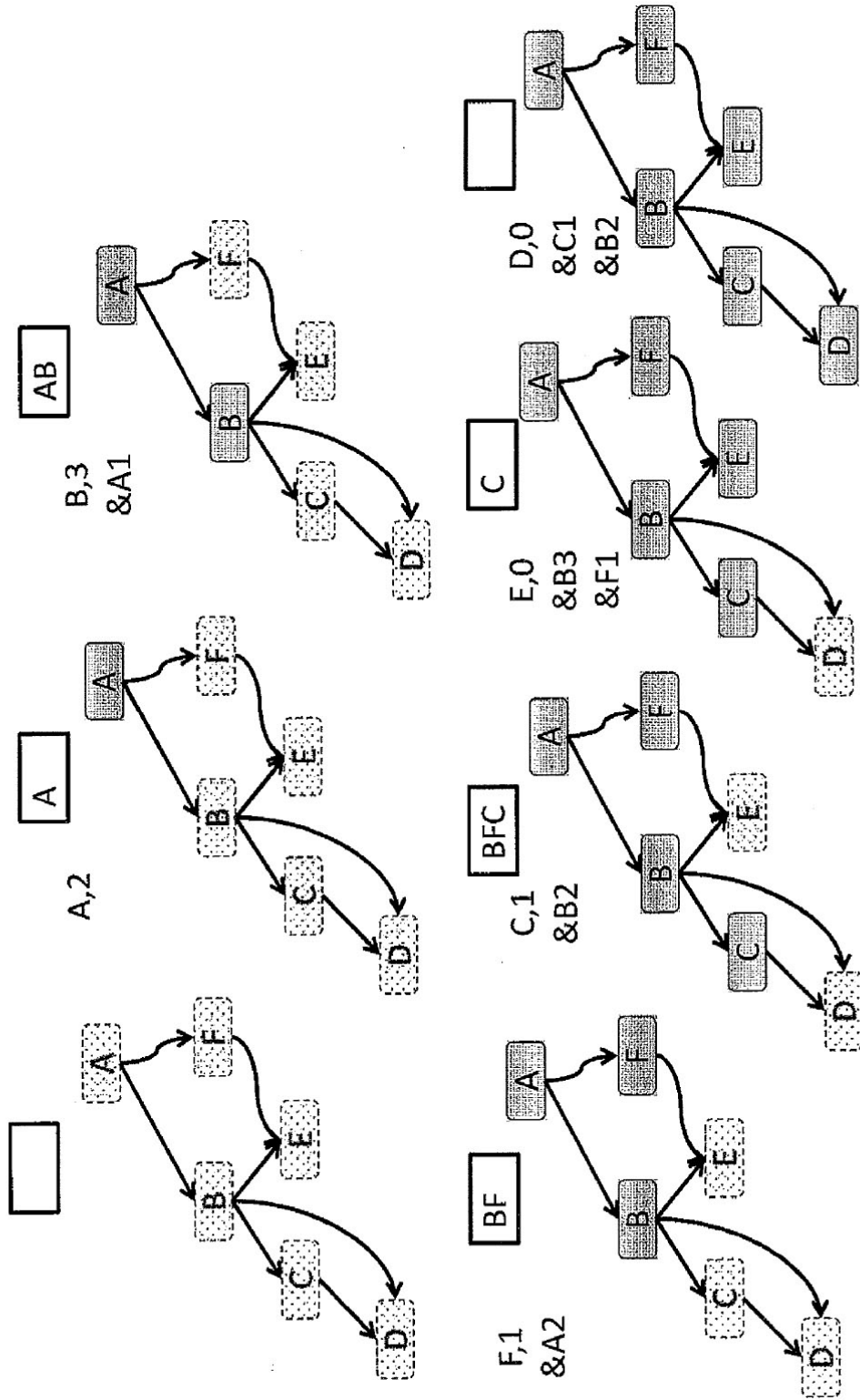


Figura 8: Cola de nodos de dos niveles

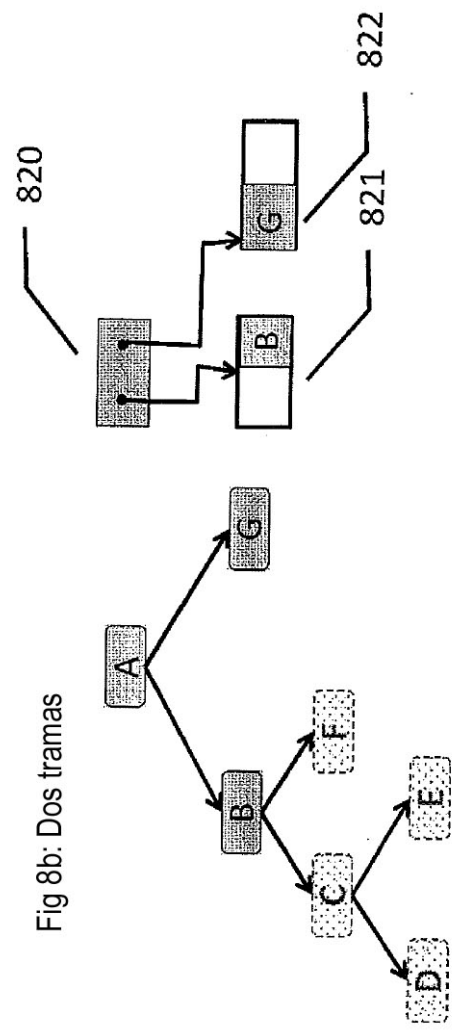
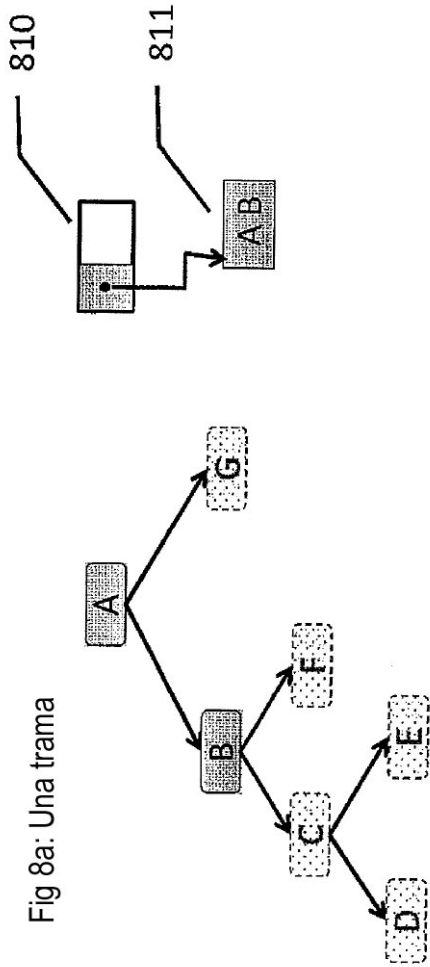


Figura 9

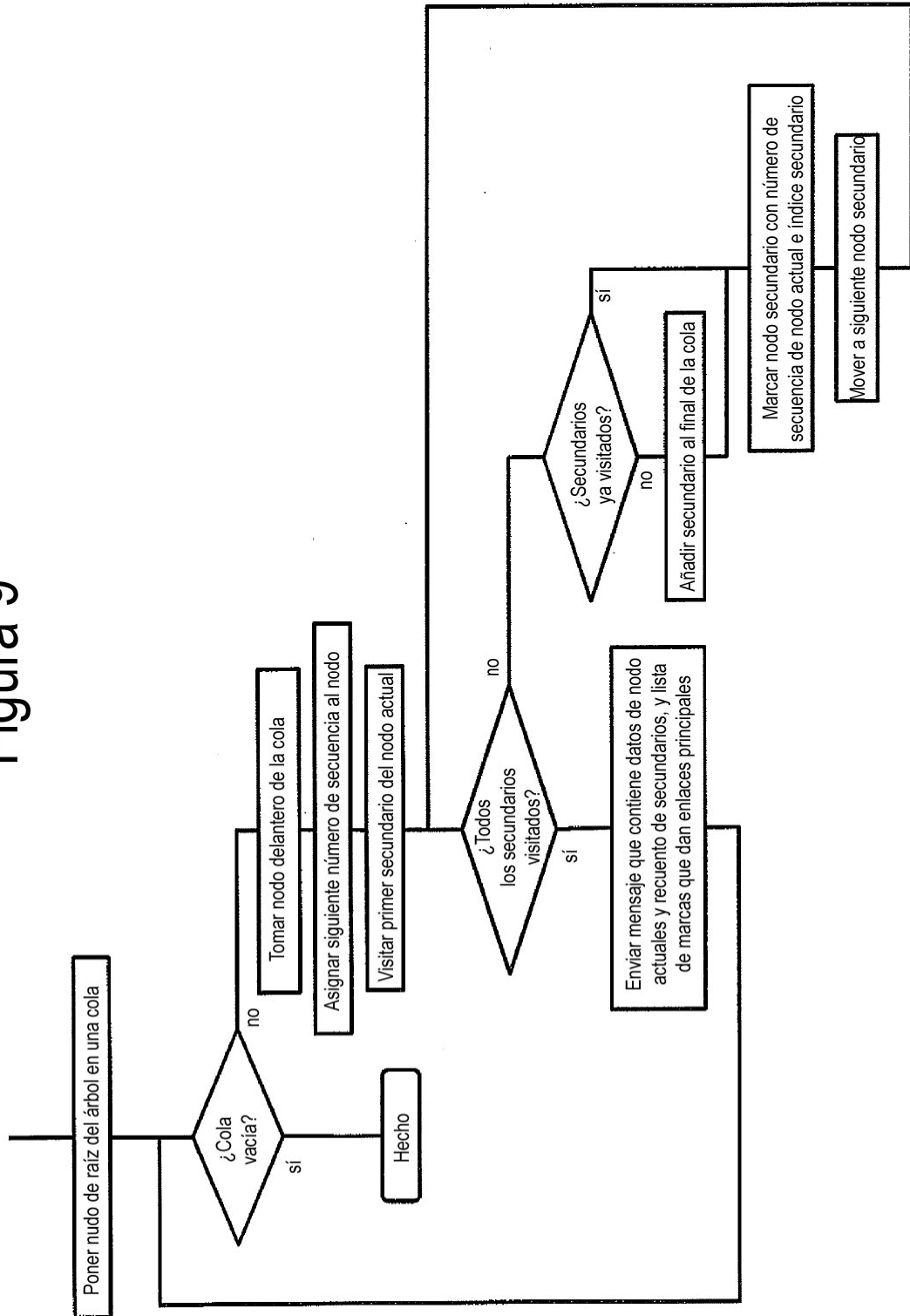


Figura 10

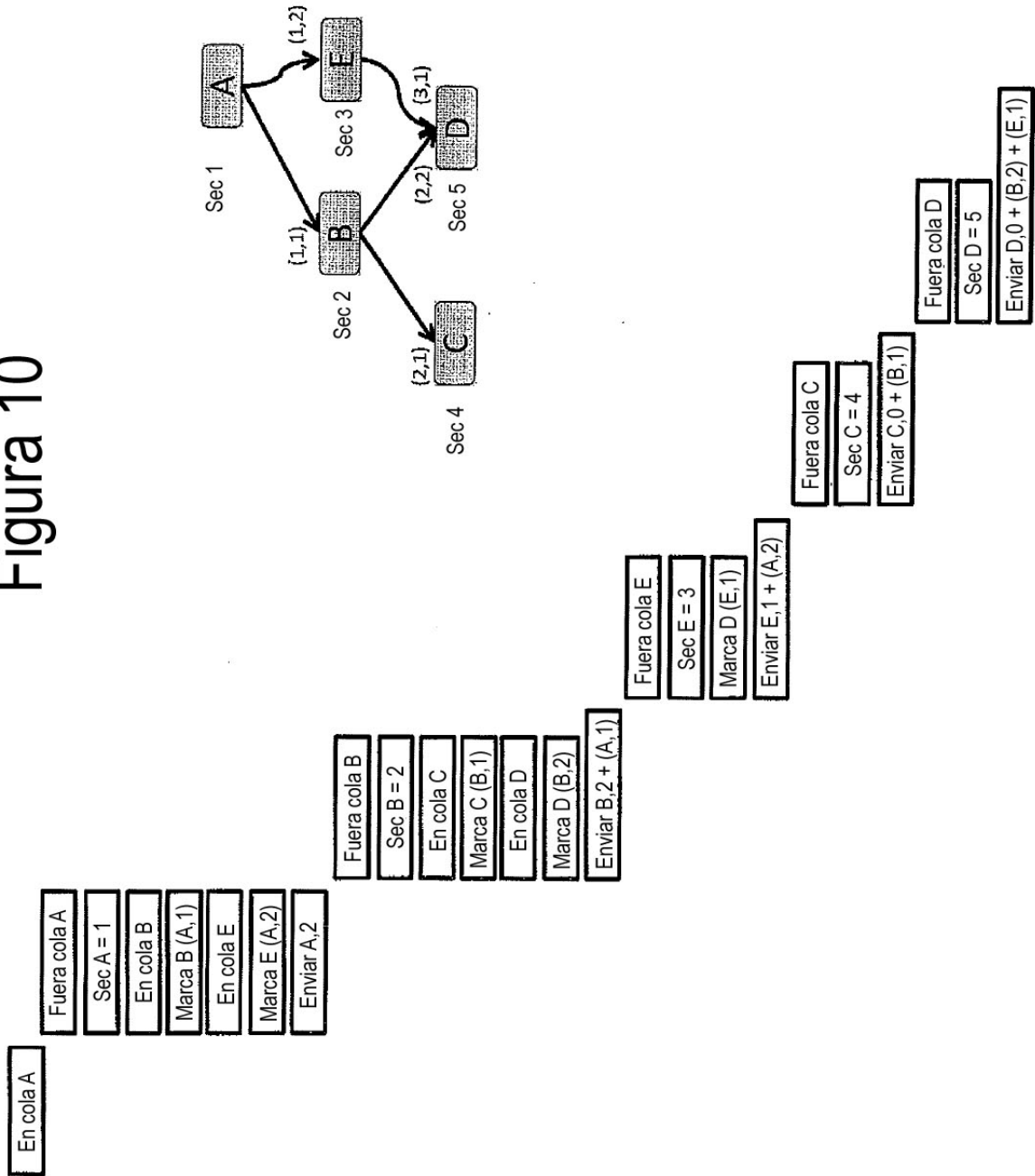


Figura 11

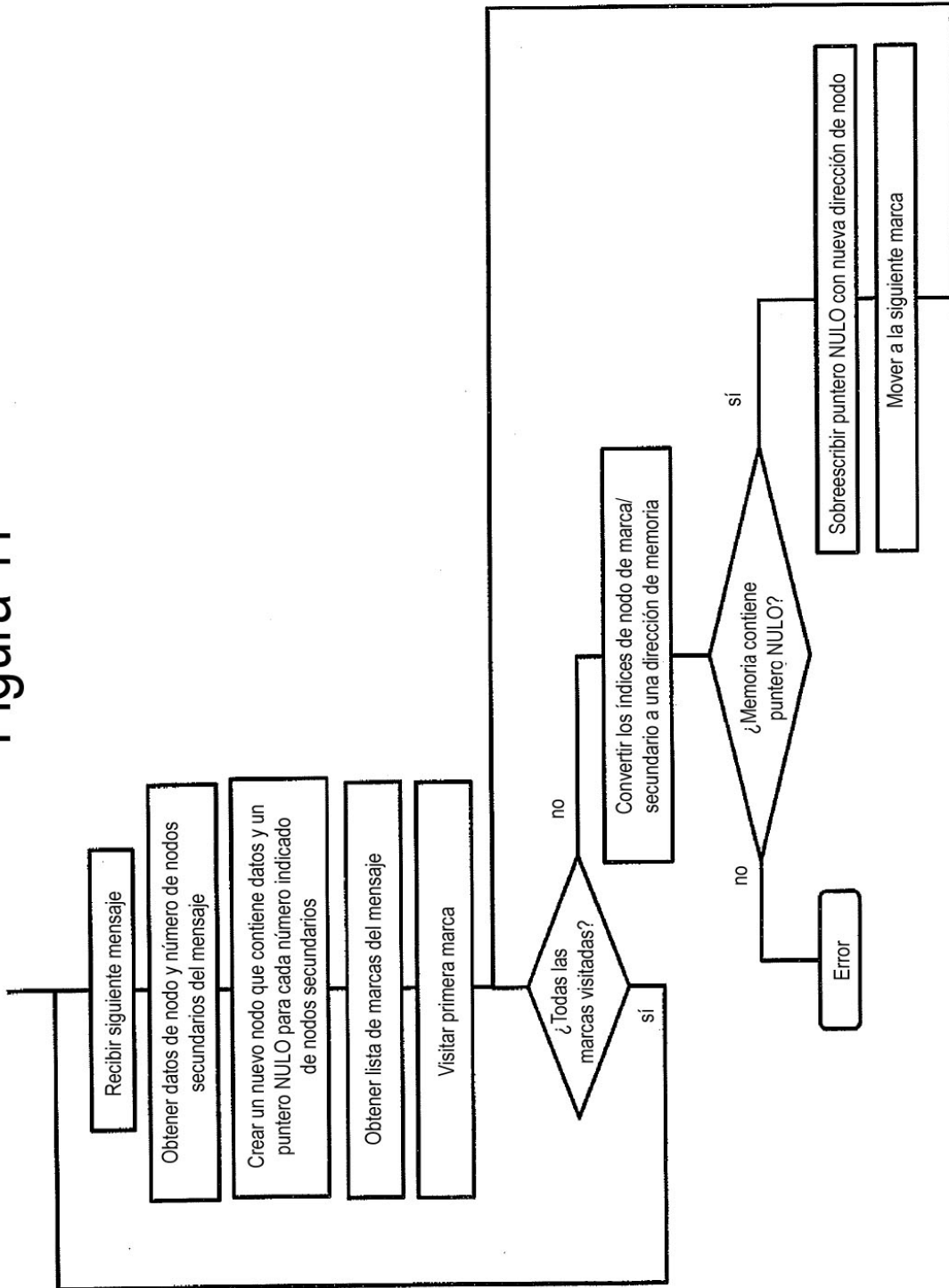


Figura 12

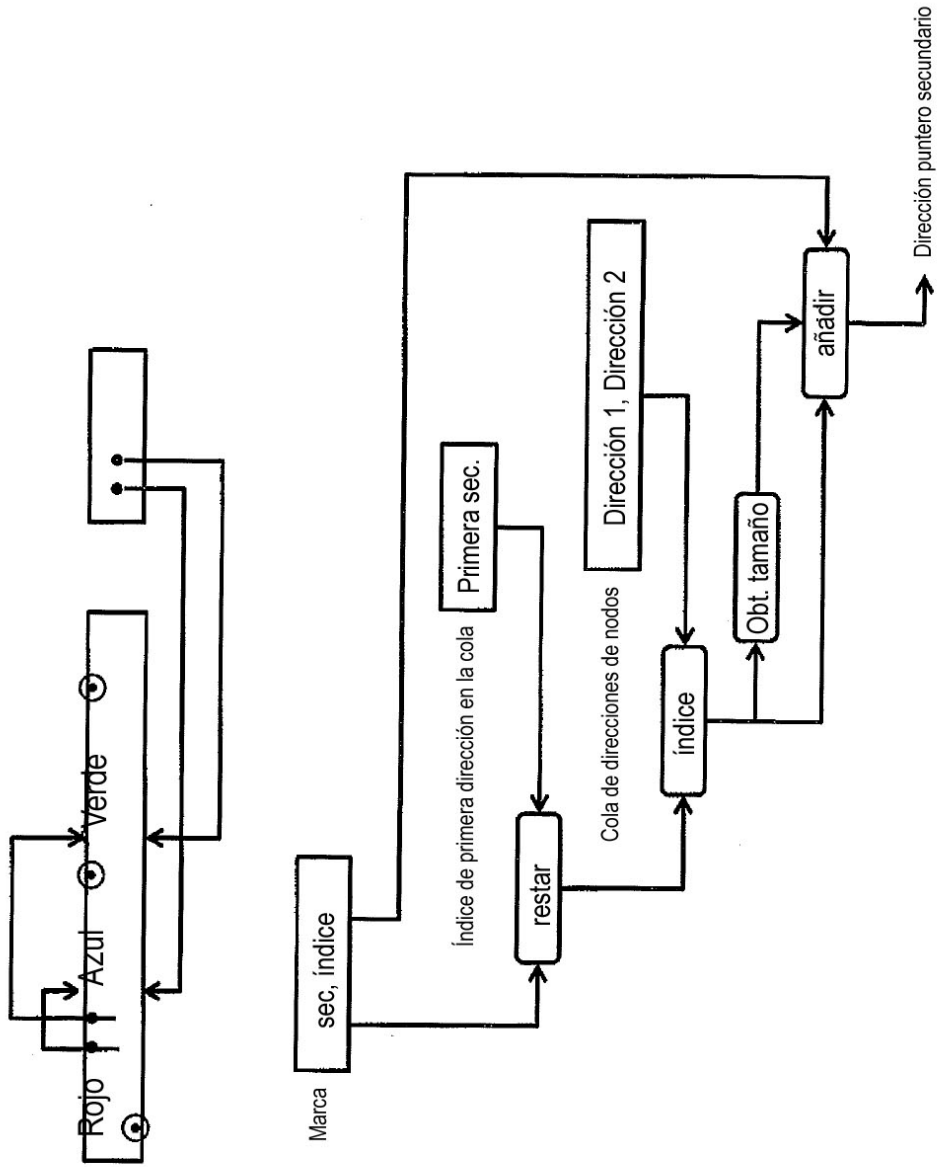


Figura 13

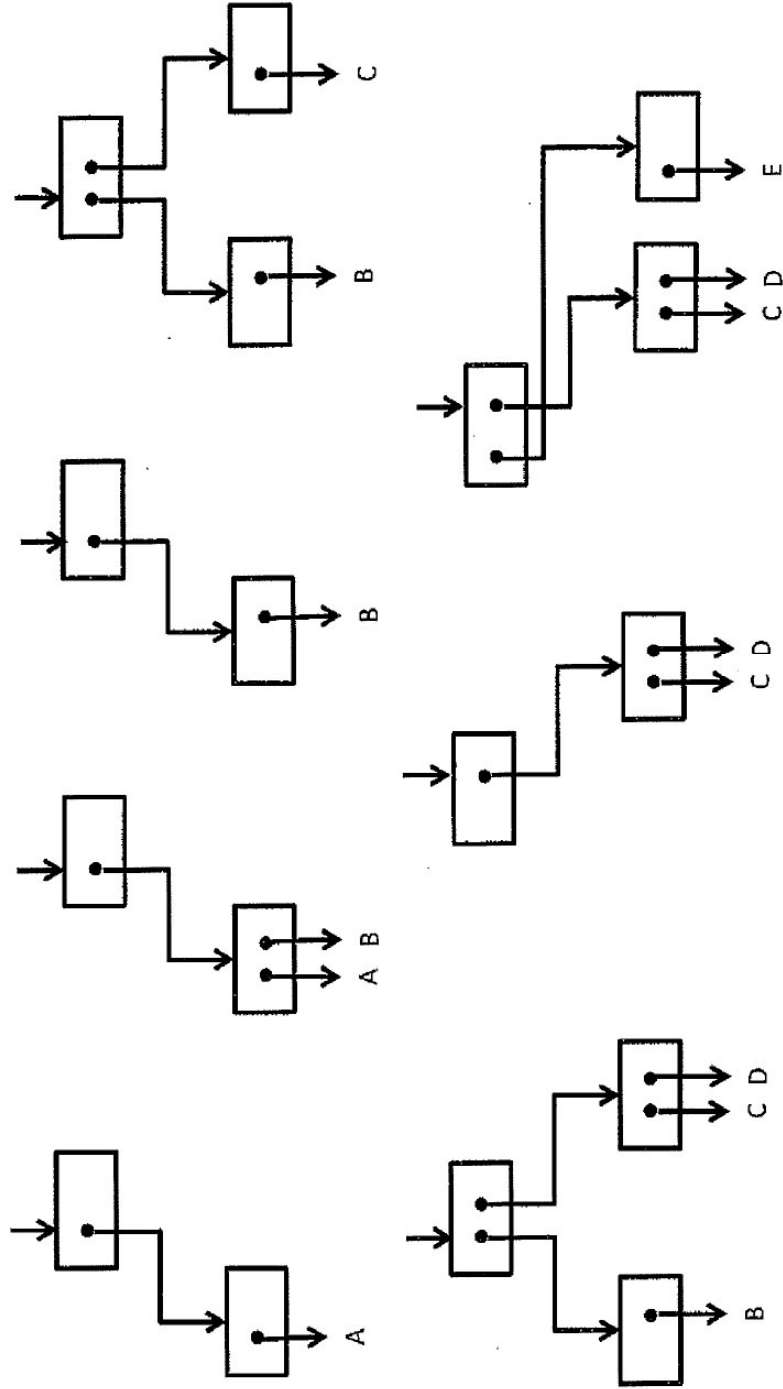


Figura 14

