

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 615 855**

51 Int. Cl.:

G06F 12/02 (2006.01)

G06F 9/06 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **06.04.2012 PCT/US2012/032639**

87 Fecha y número de publicación internacional: **01.11.2012 WO2012148650**

96 Fecha de presentación y número de la solicitud europea: **06.04.2012 E 12776762 (2)**

97 Fecha y número de publicación de la concesión europea: **16.11.2016 EP 2702490**

54 Título: **Recolección de elementos no utilizados conservativa y números enteros etiquetados para la gestión de memoria**

30 Prioridad:

25.04.2011 US 201113093724

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

08.06.2017

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)
One Microsoft Way
Redmond, WA 98052, US**

72 Inventor/es:

**LUCCO, STEVEN y
MAN, CURTIS CHENG-CHENG**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 615 855 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

Recolección de elementos no utilizados conservativa y números enteros etiquetados para la gestión de memoria

Antecedentes

5 A modo de antecedentes relativos a algunos sistemas convencionales, se observa que los dispositivos informáticos han almacenado tradicionalmente información y han asociado aplicaciones. Para estos fines, es deseable la implementación de un esquema de gestión de memoria eficiente para lograr un rendimiento informático aumentado. Los avances en esquemas de gestión de memoria automáticos han sido particularmente deseables sobre esquemas de gestión de memoria manuales. Los algoritmos recolectores de elementos no utilizados, por ejemplo, son esquemas de gestión de memoria automáticos que intentan recuperar la memoria ocupada por objetos que ya no están en uso por un programa en particular.

10 El recolector de elementos no utilizados de seguimiento es el tipo más común de recolector de elementos no utilizados. El recolector de elementos no utilizados de seguimiento determina primero qué objetos son accesibles (o potencialmente alcanzables) y luego descarta todos los objetos restantes. Un objeto accesible puede definirse como un objeto para el que existe alguna variable en el entorno de programa que condujo a la variable, ya sea directamente o a través de la referencia de otros objetos accesibles. Más precisamente, los objetos son generalmente accesibles de dos maneras. En primer lugar, se supone que un conjunto de objetos diferenciado, conocido como raíces, es accesible. Normalmente, estas incluyen los objetos mencionados desde cualquier punto de la pila de llamadas (es decir, invocándose actualmente todas las variables y parámetros locales en funciones) y cualquier variable global. En segundo lugar, todo lo mencionado desde un objeto accesible se considera como accesible.

15 Una complicación con los recolectores de elementos no utilizados convencionales, en lenguajes con tipos de referencia y tipos de valores desencajados, es que el recolector de elementos no utilizados distingue entre qué variables de una pila de llamadas son valores regulares (por ejemplo, números enteros) y qué variables son referencias (por ejemplo, punteros). Tal distinción sería útil en tales sistemas para propósitos de distinción, ya que en caso contrario un número entero y una referencia podrían ser parecidos según se representa en la memoria. A este respecto, el recolector de elementos no utilizados sería capaz de usar la distinción para decidir si tratar un elemento en particular como una referencia y seguir la referencia, o ser simplemente un valor primitivo.

20 Las deficiencias descritas anteriormente de los esquemas de gestión de memoria de hoy en día solo pretenden proporcionar una visión de conjunto de algunos de los problemas de los sistemas convencionales y no pretenden ser exhaustivas. Otros problemas con el estado de la técnica y los beneficios correspondientes de algunas de las diversas realizaciones no limitativas pueden hacerse más evidentes tras la revisión de la siguiente descripción detallada.

25 Marlow et al: "*Making a Fast Curry: Push/Enter vs. Eval/Apply for Higher-order Languages*", ICFP'04: Procedimientos de la conferencia internacional de ACM SIGPLAN sobre la programación de funciones, 19 de septiembre de 2004 (19-9-2004), pp. 4-15, Nueva York, EE. UU., se refiere a hacer un curry: *push/enter* en comparación con *eval/apply* para lenguajes de orden superior. Este documento aborda lenguajes de orden superior que fomentan el *curry* que se implementan usando uno de dos modelos básicos de evaluación: *push/enter* o *eval/apply*. Los implementadores usan su intuición y juicios cualitativos para elegir un modelo u otro. El objetivo de este artículo es el de proporcionar, por primera vez, una base más sustancial para esta elección, basada en nuestra experiencia cualitativa y cuantitativa de implementar ambos modelos en un compilador de última generación para Haskell.

30 Boehm, H-J. et al: "*Garbage Collection in an Uncooperative Environment*", *Software Practice and Experience*, Wiley & Sons, Bognor Regis, Gran Bretaña, volumen 18 n.º (9), 1 de septiembre de 1988 (1-9-1988), pp. 807-820, ISSN: 0038-0644, DOI: 10.1002/SPE.4380180902 se refiere a la recolección de elementos no utilizados en un entorno no cooperativo. Este documento describe una técnica para la asignación de almacenamiento y recolección de elementos no utilizados en ausencia de cooperación importante del código usando el asignador. Esto limita la recolección de elementos no utilizados al tiempo realmente requerido para la recolección de elementos no utilizados. En particular, los programas de aplicación que rara vez o nunca hacen uso del recolector ya no se enfrentan a una penalización de rendimiento sustancial. Este enfoque simplifica en gran medida la implementación de lenguajes que soportan la recolección de elementos no utilizados. Asimismo, permite que los compiladores convencionales se usen con un recolector de elementos no utilizados, ya sea como medio principal de restauración de almacenamiento o como herramienta de ayuda para pruebas.

35 El documento US2007150868 se refiere a un código de borrado de pila de inserción para la recolección de elementos no utilizados conservativa. Este documento desvela una ubicación para insertar un código de borrado de pila en un procedimiento determinado que se debe ejecutar en un entorno de ejecución de un sistema informático. El código de borrado de pila se inserta en la ubicación del procedimiento. El código de borrado de pila se ejecuta durante la ejecución del procedimiento para borrar una pila. Asimismo, se describen y reivindican otras realizaciones.

El documento US2008098296 se refiere a la representación del contenido de lenguaje de marcado de hipertexto.

Este documento desvela generalmente un aspecto de la materia objeto descrita en esta memoria descriptiva que se puede incorporar en un procedimiento que incluye la representación del contenido de lenguaje de marcado de hipertexto (HTML), en un motor de representación HTML, a las primitivas de un motor de representación de gráficos vectoriales; la representación de las primitivas, en el motor de representación de gráficos vectoriales, para proporcionar una interfaz de usuario; la recepción de un evento de entrada a través de la interfaz de usuario; la determinación, en el motor de representación HTML, de un cambio en el aspecto de la interfaz de usuario basado en el evento de entrada; y la actualización de al menos una de las primitivas para la representación por parte del motor de representación de gráficos vectoriales de acuerdo con el cambio de aspecto para actualizar la interfaz de usuario.

Por lo tanto, el objeto de la invención es proporcionar un procedimiento mejorado y un dispositivo de gestión de memoria para la gestión de memoria.

Este objeto es resuelto por la materia objeto de las reivindicaciones independientes.

Las realizaciones preferentes se definen en las reivindicaciones dependientes.

En el presente documento se proporciona un sumario simplificado para ayudar a permitir una comprensión básica o general de diversos aspectos de realizaciones ejemplares y no limitativas que aparecen en la descripción más detallada y en los dibujos adjuntos. Sin embargo, este sumario no pretende ser una visión de conjunto extensa o exhaustiva. En cambio, el único propósito de este sumario es el de presentar algunos conceptos relacionados con algunas realizaciones no limitativas y ejemplares de una forma simplificada como un preámbulo a la descripción más detallada de las 10 diversas realizaciones siguientes.

De acuerdo con una o más realizaciones y la divulgación correspondiente, se describen diversos aspectos no limitativos en relación con la recolección de elementos no utilizados conservativa para la gestión de memoria. En uno de tales aspectos, se proporciona un procedimiento para el etiquetado de números enteros en el interior de un algoritmo de recolección de elementos no utilizados conservativo. El procedimiento puede incluir la identificación de objetos-raíz incluidos en una pila de llamadas, que pueden comprender representaciones de números enteros y representaciones de punteros. La realización puede incluir además el etiquetado de las representaciones de números enteros de manera que una representación de números enteros etiquetada se pueda distinguir de una representación de puntero no etiquetada. El procedimiento también puede incluir el seguimiento de los objetos-raíz a un conjunto de ubicaciones de memoria correspondiente. Para esta realización particular, se realiza un seguimiento posterior en la representación de puntero no etiquetada, mientras que el seguimiento posterior se omite en la representación de número entero etiquetada. El procedimiento puede además incluir la liberación de memoria asignada a un conjunto de objetos inaccesibles por la pila de llamadas.

En otro aspecto, se desvela un dispositivo de gestión de memoria configurado para implementar un algoritmo de recolección de elementos no utilizados conservativo con números enteros etiquetados. De acuerdo a dicha realización, el dispositivo de gestión de memoria incluye un procesador configurado para ejecutar componentes ejecutables por ordenador almacenados en memoria. Los componentes ejecutables por ordenador incluyen un componente de barrido, un componente de etiquetado, un componente de seguimiento y un componente de reciclaje. El componente de barrido está configurado para identificar objetos-raíz incluidos en una pila de llamadas, que puede comprender representaciones de números enteros y representaciones de punteros. El componente de etiquetado se configura entonces para el etiquetado de las representaciones de números enteros de manera que las representaciones de números enteros etiquetadas se distinguen de las representaciones de punteros no etiquetadas, mientras que el componente de seguimiento está configurado para localizar los objetos-raíz en un conjunto de ubicaciones de memoria correspondiente. Para esta realización, el componente de seguimiento está configurado para realizar un seguimiento posterior en las representaciones de punteros no etiquetadas, en la que el seguimiento posterior se omite en las representaciones de números enteros etiquetadas. El componente de reciclaje se configura entonces para liberar memoria asignada a un conjunto de objetos inaccesibles por la pila de llamadas.

En otro aspecto, se desvela un medio de almacenamiento legible por ordenador para el etiquetado de números enteros en el interior de un algoritmo de recolección de elementos no utilizados conservativo. Según dicha realización, el medio de almacenamiento legible por ordenador incluye instrucciones legibles por ordenador para hacer que al menos un procesador realice diversos actos. Por ejemplo, tales actos incluyen el etiquetado de un gráfico de objeto asociado con una pila de llamadas y la generación de un segmento de memoria de objetos durante una ejecución de la pila de llamadas. En este caso, los objetos-raíz incluidos en un gráfico de objeto etiquetado comprenden representaciones de números enteros etiquetadas y representaciones de punteros no etiquetadas, mientras que el segmento de memoria de objetos comprende objetos incluidos en una parte ejecutada de la pila de llamadas. Esta realización incluye además el seguimiento de objetos no ejecutados incluidos en una parte no ejecutada de la pila de llamadas a ubicaciones de memoria correspondientes en el segmento de memoria. Para esta realización particular, se realiza un seguimiento posterior en las representaciones de punteros no etiquetadas y se omite en las representaciones de números enteros etiquetadas. A continuación, se borran las ubicaciones de memoria correspondientes a objetos inaccesibles en el segmento de memoria, en las que los objetos inaccesibles son inaccesibles por la parte no ejecutada de la pila de llamadas.

Otras realizaciones y diversos ejemplos no limitativos, escenarios e implementaciones se describen con más detalle

a continuación.

Breve descripción de los dibujos

A continuación, se describen diversas realizaciones no limitativas con referencia a los dibujos adjuntos, en los que:

- 5 la figura 1 ilustra un sistema ejemplar que facilita la implementación de un algoritmo de recolección de elementos no utilizados conservativo según una realización;
 la figura 2 es un diagrama que ilustra un segmento de memoria de objetos según una realización;
 la figura 3 es un diagrama que ilustra un gráfico de objeto ejemplar según una realización;
 la figura 4 es un diagrama que ilustra una liberación de memoria ejemplar según una realización;
 10 la figura 5 es un diagrama de bloques que ilustra un dispositivo de gestión de memoria ejemplar según una realización;
 la figura 6 es un diagrama de flujo que ilustra una realización no limitativa ejemplar para implementar un algoritmo de recolección de elementos no utilizados conservativo según una realización;
 la figura 7 es un diagrama de bloques que ilustra una unidad de gestión de recursos ejemplar según una realización;
 15 la figura 8 es un diagrama de flujo que ilustra una realización no limitativa ejemplar para el etiquetado de números enteros según una realización;
 la figura 9 es un diagrama de bloques que representa entornos de red no limitativos ejemplares en los que se pueden implementar diversas realizaciones descritas en el presente documento; y
 20 la figura 10 es un diagrama de bloques que representa un sistema informático no limitativo o un entorno operativo en el que se pueden implementar uno o más aspectos de diversas realizaciones descritas en el presente documento.

Descripción detallada

Visión de conjunto

- 25 Como se ha indicado en los antecedentes, es deseable implementar un algoritmo de recolección de elementos no utilizados que distingue entre números enteros y punteros en una pila de llamadas. En diversas realizaciones, la gestión de memoria se rediseña alrededor de la compatibilidad de código original. En un aspecto, los objetos de secuencia de comandos son objetos que se gestionan menos y más se gestionan solo piezas originales de memoria, de manera que se elimina el recuento de referencias entre objetos. Además, se implementa un algoritmo de recolección de elementos no utilizados conservativo, en el que no se supone que todo lo que es un puntero es conocido. Con un recolector de elementos no utilizados de entorno en tiempo de ejecución de lenguaje común (CLR) por ejemplo, la pila está asignada rigurosamente. Sin embargo, con código original, por ejemplo, código C y código de secuencia de comandos, no se sabe qué hay en la pila. A este respecto, en lugar de usar un modelo de recuento de referencias, se contempla que los objetos interactúen directamente.

- 35 Las diversas realizaciones desveladas en el presente documento están dirigidas al etiquetado de números enteros en el interior de un algoritmo de recolección de elementos no utilizados conservativo. Con este fin, se observa que los aspectos desvelados en este presente documento facilitan una ejecución de secuencias de comandos (por ejemplo, el programa JavaScript) contra un modelo de objetos de documento que es poco probable que genere punteros falsos. Además, los aspectos desvelados en el presente documento reducen la probabilidad de punteros falsos, puesto que los punteros no etiquetados y los números enteros etiquetados son fácilmente distinguibles.

NÚMEROS ENTEROS ETIQUETADOS Y RECOLECCIÓN DE ELEMENTOS NO UTILIZADOS CONSERVATIVA

- 40 Han surgido varias cuestiones a medida que la experiencia de navegación web evoluciona desde una presentación plana de información con una interactividad mínima hasta una aplicación más enriquecida o una experiencia de miniaplicación con mucha interactividad en el lado del cliente. Más generalmente, la experiencia de navegación web se ha convertido en un híbrido de visualización de información y una interactividad más enriquecida con objetos en pantalla. Los desafíos particulares con esta evolución se basan en la adaptación del antiguo modelo de objetos de documento (DOM), originalmente diseñado principalmente para la presentación plana de información basada en código original de un cliente, a una experiencia que maneja fluidamente el código de secuencia de comandos, tal como objetos JavaScript.

- 50 La mejora de la velocidad es particularmente importante para facilitar una experiencia de usuario más deseable. Por ejemplo, con menús desplegados del pasado, la experiencia web parpadeó con retrasos basados en las comunicaciones con el servidor. Sin embargo, las secuencias de comandos permiten a los programas pequeños modificar el DOM al instante sin volver al servidor. Puesto que la gente quiere hacer más cosas al instante sin volver al servidor, hacer que el código de secuencia de comandos se ejecute rápidamente se ha convertido en un desafío.

- 55 Puesto que la experiencia del usuario está ampliamente influenciada por la eficacia de la secuencia de comando del DOM, es deseable cambiar el DOM tan rápido como sea posible para maximizar la respuesta interactiva. La comunicación entre los motores de secuencias de comandos y las clases originales del DOM era pobre en el pasado debido al uso de la automatización de vinculación e incrustación de objetos (OLE) que incluye un conjunto de

interfaces, por ejemplo, iDispatch, iActiveScript, etc. que hacen que cualquier objeto sea programable. Sin embargo, tales procedimientos son lentos, y por lo tanto se desean mejoras. Por consiguiente, los aspectos desvelados en el presente documento están dirigidos a mejorar la velocidad de ejecución de la secuencia de comando mediante el etiquetado de números enteros en el interior de un algoritmo de recolección de elementos no utilizados conservativo.

5 La figura 1 ilustra un sistema ejemplar que facilita la implementación de un algoritmo de recolección de elementos no utilizados conservativo según una realización. Como se ilustra, el sistema 100 puede incluir una unidad 110 de gestión de memoria, que está acoplada mediante la comunicación a la memoria 120. En un aspecto, la unidad 110 de gestión de memoria está configurada para implementar un algoritmo de recolección de elementos no utilizados conservativo para gestionar los elementos no utilizados en la memoria 120. Además, la unidad 110 de gestión de memoria está configurada para generar un gráfico 114 de objeto etiquetado asociado con una ejecución de la secuencia 112 de comandos. Por ejemplo, la secuencia 112 de comandos puede ser un programa JavaScript ejecutado contra un DOM, en el que el programa JavaScript incluye varios objetos que requieren una asignación de elementos no utilizados en la memoria 120. En una realización particular, para facilitar la distinción entre números enteros y punteros en el interior de la memoria 120, los objetos de secuencia de comandos correspondientes a valores de números enteros son etiquetados por la unidad 110 de gestión de memoria y se incluyen en el gráfico 114 de objetos etiquetados.

En un aspecto, la memoria asignada a objetos de secuencia de comandos comprende el almacenamiento de dichos objetos en un segmento de memoria. A continuación, haciendo referencia a la figura 2, se proporciona un diagrama de bloques de un segmento de memoria de objetos ejemplar según una realización. Como se muestra, el segmento 200 de memoria puede incluir una memoria 210 disponible, junto con la memoria asignada correspondiente a varios objetos 220, 230, 240, 250, 260 y 270. Para este ejemplo particular, los objetos 240 y 270 corresponden a valores de puntero, mientras que los objetos 220, 230, 250 y 260 corresponden a valores de números enteros. Concretamente, el objeto 240 es un valor de puntero que hace referencia al valor de número entero representado por el objeto 230, mientras que el objeto 270 es un valor de puntero que hace referencia al valor de número entero representado por el objeto 260.

A este respecto, los objetos de secuencia de comandos correspondientes a valores de números enteros pueden etiquetarse de manera que se puedan distinguir fácilmente de los valores de puntero. A continuación, haciendo referencia a la figura 3, se proporciona un diagrama que ilustra un gráfico de objeto ejemplar que facilita un mapeo de objetos etiquetados/no etiquetados en el segmento de memoria. Como se ilustra, el gráfico 300 de objeto mapea objetos incluidos en una parte 310 de pila de llamadas no ejecutada al segmento 320 de memoria. Para estos fines y otros relacionados, se observa que el segmento 320 de memoria y la memoria 330 disponible son generalmente análogos a la memoria 210 disponible del segmento 200 de memoria, respectivamente.

En un aspecto, el gráfico 300 de objeto se utiliza para determinar qué objetos del segmento 320 de memoria son accesibles por la parte 310 de pila de llamadas no ejecutada. Concretamente, los objetos-raíz incluidos en la parte 310 de pila de llamadas no ejecutada están localizados a las ubicaciones de memoria correspondientes en el segmento 320 de memoria, en el que se etiquetan los objetos-raíz correspondientes a valores de números enteros. Se contempla entonces que un seguimiento posterior de objetos-raíz se realice solamente en valores de puntero no etiquetados accesibles para identificar objetos accesibles mencionados por esos valores de puntero. Por el contrario, se omite un seguimiento posterior en valores de números enteros etiquetados accesibles. Para este ejemplo particular, puesto que el objeto 390 accesible es un objeto-raíz que corresponde a un valor de puntero, no está etiquetado. Por lo tanto, se realiza un seguimiento posterior en el objeto 390 accesible, que identifica el objeto 380 accesible que corresponde a un valor de número entero. Sin embargo, se omite un seguimiento posterior en los objetos 340 y 370 accesibles, ya que estos son objetos-raíz correspondientes a valores de números enteros etiquetados.

Se observa que el gráfico 300 de objeto se puede usar para identificar objetos inaccesibles por la parte 310 de pila de llamadas no ejecutada. En este ejemplo particular, los objetos 350 y 360 inaccesibles se consideran inaccesibles, ya que no corresponden a objetos-raíz en la parte 310 de pila de llamadas no ejecutada, ni tampoco son mencionados por un objeto de puntero accesible.

Al identificar objetos inaccesibles, la memoria asignada a tales objetos puede ser borrada. A continuación, haciendo referencia a la figura 4, se proporciona un diagrama que ilustra una liberación de memoria ejemplar según una realización. Como se ilustra, el gráfico 400 de objeto incluye el segmento 420 de memoria, que es generalmente análogo al segmento 200 y 320 de memoria, en el que el segmento 420 de memoria representa una liberación de memoria previamente asignada a objetos 350 y 360 inaccesibles. Concretamente, el segmento 420 de memoria incluye ahora memoria 450 y 460 liberada, además de la memoria 430 disponible. Sin embargo, en un aspecto, los objetos 440, 470, 480 y 490 accesibles se conservan en sus lugares de almacenamiento originales (es decir, el segmento 420 de memoria no se colapsa).

A continuación, haciendo referencia a la figura 5, un diagrama de bloques ilustra una unidad de gestión de memoria ejemplar configurada para implementar un algoritmo de recolección de elementos no utilizados conservativo según diversos aspectos. Como se ilustra, la unidad 500 de gestión de memoria puede incluir un componente 510 de procesador, un componente de 520 memoria, un componente 530 de barrido, un componente 540 de etiquetado, un

componente 550 de seguimiento y un componente 560 de reciclaje.

5 En un aspecto, el componente 510 de procesador está configurado para ejecutar instrucciones legibles por ordenador relacionadas con la realización de cualquiera de una pluralidad de funciones. El componente 510 de procesador puede ser un único procesador o una pluralidad de procesadores dedicados a analizar la información a comunicar desde la unidad 500 de gestión de memoria y/o generar información que pueden usar el componente 520 de memoria, el componente 530 de barrido, el componente 540 de etiquetado, el componente 550 de seguimiento y/o el componente 560 de reciclaje. Adicionalmente o como alternativa, el componente 510 de procesador puede configurarse para controlar uno o más componentes de la unidad 500 de gestión de memoria.

10 En otro aspecto, el componente 520 de memoria está acoplado al componente 510 de procesador y configurado para almacenar instrucciones legibles por ordenador ejecutadas por el componente 510 de procesador. El componente 520 de memoria también puede configurarse para almacenar cualquiera de una pluralidad de otros tipos de datos que incluyen datos generados por cualquiera del componente 530 de barrido, el componente 540 de etiquetado, el componente 550 de seguimiento y/o el componente 560 de reciclaje. El componente 520 de memoria puede configurarse en varias configuraciones diferentes, incluyendo como memoria de acceso aleatorio, memoria respaldada por batería, disco duro, cinta magnética, etc. También pueden implementarse diversas características sobre el componente 520 de memoria, tales como la compresión y copia de seguridad automática, por ejemplo, el uso de una configuración de matriz redundante de discos independientes.

20 Como se ilustra, la unidad 500 de gestión de memoria también puede incluir un componente 530 de barrido. De acuerdo a dicha realización, el componente 530 de barrido está configurado para identificar objetos-raíz incluidos en una pila de llamadas, que comprenden representaciones de números enteros y representaciones de punteros. En este caso, se contempla que dicha pila de llamadas esté asociada con una secuencia de comandos ejecutada contra un modelo de objeto particular. Por ejemplo, la pila de llamadas puede asociarse con una compilación de un programa JavaScript, lo que facilita la modificación de un modelo de objetos de documento durante el tiempo de ejecución.

25 Como se ilustra, la unidad 500 de gestión de memoria puede incluir además un componente 540 de etiquetado que está configurado para etiquetar las representaciones de números enteros incluidos en la pila de llamadas. En este caso, se observa que el etiquetado de números enteros se realiza de tal manera que las representaciones de números enteros etiquetados se distinguen fácilmente de las representaciones de punteros no etiquetados. Con este fin, se observa además que el componente 540 de etiquetado se puede configurar para etiquetar números enteros en cualquiera de una pluralidad de maneras. Por ejemplo, el componente 540 de etiquetado se puede configurar para transformar las representaciones de números enteros en una estructura de datos diferente. En una realización particular, dicha transformación puede incluir tener un componente 540 de etiquetado configurado para dedicar a una etiqueta al menos un bit en la estructura de datos diferente.

35 En un aspecto, el dispositivo 500 de gestión de memoria incluye además el componente 550 de seguimiento. De acuerdo a tal realización, el componente 550 de seguimiento está configurado para localizar los objetos-raíz identificados por el componente 530 de barrido en un conjunto de ubicaciones de memoria correspondientes. Después de un seguimiento inicial de los objetos-raíz, se contempla que se realice un seguimiento posterior de objetos relacionados. Sin embargo, en una realización particular, el seguimiento posterior solo se realiza en las representaciones de punteros no etiquetadas y se omite en las representaciones de números enteros etiquetadas.

40 En otro aspecto, el dispositivo 500 de gestión de memoria también incluye el componente 560 de reciclaje, que está configurado para liberar memoria asignada a un conjunto de objetos inaccesibles por la pila de llamadas. En una realización particular, el componente 560 de reciclaje puede configurarse para colocar objetos llamados por la pila de llamadas sobre un segmento de memoria. De acuerdo a tal realización, el componente 560 de reciclaje puede entonces configurarse además para iniciar un algoritmo de recolección de elementos no utilizados basado en el hecho de que un tamaño del segmento de memoria excede un umbral. En otra realización, el componente 560 de reciclaje puede configurarse para conservar respectivamente objetos accesibles en ubicaciones de almacenamiento originales de un segmento de memoria. Por ejemplo, el componente 560 de reciclaje puede configurarse para mantener una separación de objetos de segmentos de memoria según un conjunto fijo de límites en el interior del segmento de memoria.

50 La figura 6 es un diagrama de flujo que ilustra una realización no limitativa ejemplar para implementar un algoritmo de recolección de elementos no utilizados conservativo según una realización. En 600, se identifican los objetos-raíz incluidos en una pila de llamadas, que comprenden representaciones de números enteros y representaciones de punteros. A continuación, en 610, las representaciones de números enteros se etiquetan de manera que las representaciones de números enteros etiquetadas se distinguen de las representaciones de punteros no etiquetadas. Los objetos-raíz se localizan entonces en un conjunto de ubicaciones de memoria correspondiente en 620 de manera que se realiza un seguimiento posterior en las representaciones de punteros y se omite en las representaciones de números enteros. La memoria asignada a objetos inaccesibles por la pila de llamadas se libera después en 630.

A continuación, haciendo referencia a la figura 7, un diagrama de bloques ilustra una unidad de gestión de recursos

ejemplar configurada para etiquetar números enteros según diversos aspectos. Como se ilustra, la unidad 700 de gestión de recursos puede incluir un componente 710 de procesador, un componente 720 de memoria, un componente 730 de etiquetado, un componente 740 de segmento de memoria, un componente 750 de seguimiento, un componente 760 de liberación, un componente 770 de supervisión y un componente 780 gráfico.

5 Igual que el componente 510 de procesador en la unidad 500 de gestión de memoria, el componente 710 de procesador está configurado para ejecutar instrucciones legibles por ordenador relacionadas con la realización de cualquiera de una pluralidad de funciones. El componente 710 de procesador puede ser un único procesador o una pluralidad de procesadores dedicados a analizar la información que ha de comunicarse desde la unidad 700 de gestión de recursos y/o generar información que se puede usar por el componente 720 de memoria, el componente 730 de etiquetado, el componente 740 de segmento de memoria, el componente 750 de seguimiento, el componente 760 de liberación, el componente 770 de supervisión y/o el componente 780 gráfico. Adicionalmente o como alternativa, el componente 710 de procesador puede configurarse para controlar uno o más componentes de la unidad 700 de gestión de recursos.

15 En otro aspecto, el componente 720 de memoria está acoplado al componente 710 de procesador y configurado para almacenar instrucciones legibles por ordenador ejecutadas por el componente 710 de procesador. El componente 720 de memoria también puede configurarse para almacenar cualquiera de una pluralidad de otros tipos de datos que incluyen datos generados por cualquiera del componente 730 de etiquetado, el componente 740 de segmento de memoria, el componente 750 de seguimiento, el componente 760 de liberación, el componente 770 de supervisión y/o el componente 780 gráfico. En este caso, se observa que el componente 720 de memoria es análogo al componente 520 de memoria en la unidad 500 de gestión de memoria. Por consiguiente, se puede apreciar que cualquiera de las características/configuraciones del componente 520 de memoria mencionadas anteriormente también son aplicables al componente 720 de memoria.

25 Como se ilustra, la unidad 700 de gestión de recursos también puede incluir el componente 730 de etiquetado. De acuerdo a dicha realización, el componente 730 de etiquetado está configurado para etiquetar un gráfico de objeto asociado con una pila de llamadas. En este caso, se debe observar que los objetos-raíz incluidos en un gráfico de objeto etiquetado pueden comprender representaciones de números enteros etiquetadas y representaciones de punteros no etiquetadas. Para algunas realizaciones, se debe observar además que la unidad 700 de gestión de recursos puede incluir también un componente 780 gráfico, que está configurado para realizar gráficos de objetos etiquetados por el componente 730 de etiquetado.

30 En un aspecto, la unidad 700 de gestión de recursos está configurada para gestionar objetos incluidos en un segmento de memoria. Para facilitar dicha gestión, la unidad 700 de gestión de recursos puede incluir el componente 740 de segmento de memoria, que está configurado para generar un segmento de memoria de objetos durante una ejecución de la pila de llamadas. Para esta realización particular, se contempla que este segmento de memoria de objetos pueda comprender objetos incluidos en una parte ejecutada de la pila de llamadas.

35 En otro aspecto, la unidad 700 de gestión de recursos puede incluir además el componente 750 de seguimiento. De acuerdo a dicha realización, el componente 750 de seguimiento está configurado para localizar objetos no ejecutados incluidos en una parte no ejecutada de la pila de llamadas en las ubicaciones de memoria correspondientes en el segmento de memoria. En este caso, al igual que el componente 550 de seguimiento en el dispositivo 500 de gestión de memoria, se contempla que el componente 750 de seguimiento pueda configurarse para realizar un seguimiento posterior de objetos relacionados después de un seguimiento inicial de los objetos-raíz de la pila de llamadas. Además, se contempla que dicho seguimiento posterior solo se realice en las representaciones de punteros no etiquetadas y se omita en las representaciones de números enteros etiquetadas.

45 Como se ilustra, la unidad 700 de gestión de recursos también puede incluir un componente 760 de liberación. De acuerdo a dicha realización, el componente 760 de liberación está configurado para borrar ubicaciones de memoria correspondientes a objetos en el segmento de memoria que se consideran inaccesibles por la parte no ejecutada de la pila de llamadas. En un aspecto, el componente 760 de liberación puede configurarse además para conservar respectivamente un almacenamiento de objetos accesibles por la parte no ejecutada de la pila de llamadas en sus ubicaciones de memoria originales en el interior del segmento de memoria. En otro aspecto, la unidad 700 de gestión de recursos también puede incluir un componente 770 de supervisión, que está configurado para supervisar un tamaño del segmento de memoria durante la ejecución de la pila de llamadas. Para esta realización particular, el componente 760 de liberación puede entonces configurarse para activar un borrado de memoria asignado a objetos inaccesibles basado en el tamaño del segmento de memoria que excede un umbral particular.

55 La figura 8 es un diagrama de flujo que ilustra una realización no limitativa ejemplar para etiquetar números enteros según una realización. En 800, se etiqueta un gráfico de objeto asociado con una pila de llamadas de manera que los objetos-raíz incluidos en un gráfico de objeto etiquetado comprenden representaciones de números enteros etiquetadas y representaciones de punteros no etiquetadas. En 810, se genera entonces un segmento de memoria de objetos durante una ejecución de la pila de llamadas, que comprende objetos incluidos en una parte ejecutada de la pila de llamadas. A continuación, en 820, los objetos no ejecutados incluidos en una parte no ejecutada de la pila de llamadas se localizan hasta ubicaciones de memoria correspondientes en el segmento de memoria. Para esta realización particular, un seguimiento posterior solo se realiza en las representaciones de punteros y no en las

representaciones de números enteros. Las ubicaciones de memoria correspondientes a objetos de segmentos de memoria inaccesibles por la parte no ejecutada de la pila de llamadas se borran entonces en 830.

ENTORNOS DE RED Y DISTRIBUIDOS EJEMPLARES

5 Un experto en la materia puede apreciar que las diversas realizaciones para etiquetar números enteros en el interior de un algoritmo de recolección de elementos no utilizados conservativo descrito en el presente documento pueden implementarse en relación con cualquier ordenador u otro dispositivo cliente o servidor que pueda desplegarse como parte de una red informática o de un entorno informático distribuido, y pueda conectarse a cualquier tipo de almacén de datos. A este respecto, las diversas realizaciones descritas en el presente documento pueden implementarse en cualquier sistema informático o entorno que tenga cualquier número de unidades de memoria o de almacenamiento, 10 y cualquier número de aplicaciones y procedimientos que se produzcan a través de cualquier número de unidades de almacenamiento. Esto incluye, pero no se limita a, un entorno con ordenadores servidor y ordenadores cliente desplegados en un entorno de red o un entorno informático distribuido, que tiene almacenamiento remoto o local.

15 La figura 9 proporciona un diagrama esquemático no limitativo de un entorno informático de red o distribuido ejemplar. El entorno informático distribuido comprende objetos o dispositivos 910, 912, etc., informáticos y objetos o dispositivos 920, 922, 924, 926, 928, etc., informáticos que pueden incluir programas, procedimientos, almacenes de datos, lógica programable, etc., como se representa por las aplicaciones 930, 932, 934, 936, 938. Se puede apreciar que los objetos o dispositivos 910, 912, etc., informáticos y los objetos o dispositivos 920, 922, 924, 926, 928, etc., informáticos pueden comprender dispositivos diferentes, tales como PDA, dispositivos de audio/vídeo, teléfonos móviles, reproductores de MP3, portátiles, etc.

20 Cada objeto o dispositivo 910, 912, etc., informático y los objetos o dispositivos 920, 922, 924, 926, 928, etc., informáticos, pueden comunicarse con uno o más objetos o dispositivos 910, 912, etc., informáticos y los objetos o dispositivos 920, 922, 924, 926, 928, etc., informáticos mediante la red 940 de comunicaciones, directa o indirectamente. Aunque se ilustra como un único elemento en la figura 9, la red 940 puede comprender otros objetos informáticos y dispositivos informáticos que proporcionan servicios al sistema de la figura 9, y/o puede representar 25 múltiples redes interconectadas, que no se muestran. Cada objeto o dispositivo 910, 912, etc., o 920, 922, 924, 926, 928, etc., informático también puede contener una aplicación, tal como las aplicaciones 930, 932, 934, 936, 938, que podrían hacer uso de una interfaz de programación de aplicaciones (API), u otro objeto, software, firmware y/o hardware, adecuados para la comunicación con una infraestructura para la información o la implementación de la misma como un servicio desde cualquier plataforma tal como se proporciona según diversas realizaciones.

30 Hay una variedad de sistemas, componentes y configuraciones de red que soportan entornos informáticos distribuidos. Por ejemplo, los sistemas informáticos pueden conectarse entre sí mediante sistemas de cableado o sistemas inalámbricos, redes locales o redes ampliamente distribuidas. Actualmente, muchas redes están acopladas a Internet, lo que proporciona una infraestructura para la computación ampliamente distribuida y abarca muchas 35 redes diferentes, aunque cualquier infraestructura de red puede ser usada para comunicaciones ejemplares hechas incidentes con técnicas tal como se describe en diversas realizaciones.

De este modo, se pueden usar una multitud de topologías de red e infraestructuras de red, tales como arquitecturas cliente/servidor, punto a punto o híbridas. En una arquitectura cliente/servidor, particularmente un sistema de red, un cliente suele ser un ordenador que accede a recursos de red compartidos proporcionados por otro ordenador, por ejemplo, un servidor. En la ilustración de la figura 9, como ejemplo no limitativo, los objetos o dispositivos 920, 922, 924, 926, 928, etc., informáticos pueden considerarse clientes y los objetos o dispositivos 910, 912, etc., informáticos 40 pueden considerarse servidores en los que los objetos o dispositivos 910, 912, etc., informáticos proporcionan servicios de datos, tales como la recepción de datos desde los objetos o dispositivos 920, 922, 924, 926, 928, etc., informáticos, almacenamiento de datos, procesamiento de datos, transmisión de datos a los objetos o dispositivos 920, 922, 924, 926, 928, etc., informáticos, aunque cualquier ordenador puede considerarse un cliente, un servidor o 45 ambos, dependiendo de las circunstancias. Cualquiera de estos dispositivos informáticos puede procesar datos o solicitar servicios o tareas que puedan implicar la gestión de memoria y técnicas relacionadas como se describe en el presente documento para una o más realizaciones.

Normalmente, un servidor es un sistema informático remoto accesible a través de una red remota o local, tal como Internet o las infraestructuras de red inalámbricas. El procedimiento cliente puede estar activo en un primer sistema 50 informático y el procedimiento servidor puede estar activo en un segundo sistema informático, comunicándose entre sí mediante un medio de comunicaciones, proporcionando de este modo funcionalidad distribuida y permitiendo que múltiples clientes aprovechen las capacidades de recopilación de información del servidor. Cualquier objeto de software usado según el perfil de usuario se puede proporcionar de forma independiente o se puede distribuir a través de múltiples dispositivos u objetos informáticos.

55 En un entorno de red en el que la red/bus 940 de comunicaciones es Internet, por ejemplo, los objetos o dispositivos 910, 912, etc., informáticos pueden ser servidores web con los que se comunican los objetos o dispositivos 920, 922, 924, 926, 928, etc., informáticos a través de cualquiera de una serie de protocolos conocidos, tales como el protocolo de transferencia de hipertexto (HTTP). Como se ha mencionado, los objetos o dispositivos 910, 912, etc., informáticos pueden servir también como objetos o dispositivos 920, 922, 924, 926, 928, etc., informáticos, o

viceversa, como puede ser característico de un entorno informático distribuido.

DISPOSITIVO INFORMÁTICO EJEMPLAR

5 Como se ha mencionado, varias realizaciones descritas en el presente documento se aplican a cualquier dispositivo en el que puede ser deseable implementar una infraestructura para etiquetar números enteros en el interior de un algoritmo de recolección de elementos no utilizados conservativo. Por lo tanto, se entiende que se contemplan dispositivos manuales, portátiles y otros dispositivos informáticos y objetos informáticos de todo tipo para uso en relación con las diversas realizaciones descritas en el presente documento, es decir, en cualquier lugar en que un dispositivo pueda proporcionar alguna funcionalidad en relación con la implementación de un algoritmo de recolección de elementos no utilizados conservativo con números enteros etiquetados. Por consiguiente, el ordenador remoto de finalidad general descrito a continuación en la figura 10 es solo un ejemplo, y las realizaciones de la presente divulgación se pueden implementar con cualquier cliente que tenga interoperabilidad e interacción de red/bus.

15 Aunque no es necesario, cualquiera de las realizaciones puede implementarse parcialmente a través de un sistema operativo, para ser usado por un desarrollador de servicios para un dispositivo u objeto, y/o incluido en el software de aplicación que opera en relación con el (los) componente(s) operativo(s). El software puede describirse en el contexto general de instrucciones ejecutables por ordenador, tales como módulos de programa, que son ejecutadas por uno o más ordenadores, tales como estaciones de trabajo de cliente, servidores u otros dispositivos. Los expertos en la materia apreciarán que las interacciones de red se pueden practicar con una variedad de configuraciones y protocolos de sistemas informáticos.

20 La figura 10 ilustra, de este modo, un ejemplo de un entorno 1000 de sistema informático adecuado en el que se pueden implementar una o más de las realizaciones, aunque como se ha explicado anteriormente, el entorno 1000 de sistema informático es solo un ejemplo de un entorno informático adecuado y no pretende sugerir ninguna limitación en cuanto al ámbito de uso o funcionalidad de cualquiera de las realizaciones. El entorno 1000 informático no debe interpretarse como que tenga alguna dependencia o necesidad relacionada con alguno de los componentes o con una combinación de los mismos ilustrados en el entorno 1000 operativo ejemplar.

30 Con referencia a la figura 10, un dispositivo remoto ejemplar para implementar una o más realizaciones en el presente documento puede incluir un dispositivo informático de finalidad general en forma de un ordenador 1010 portátil. Los componentes del ordenador 1010 portátil pueden incluir, pero no se limitan a, una unidad 1020 de procesamiento, una memoria 1030 de sistema y un bus 1021 de sistema que acopla varios componentes del sistema incluyendo la memoria de sistema a la unidad 1020 de procesamiento.

35 El ordenador 1010 incluye, normalmente, una variedad de medios legibles por ordenador y puede ser cualquier medio disponible al que pueda acceder el ordenador 1010. La memoria 1030 de sistema puede incluir medios de almacenamiento informáticos en forma de memoria volátil y/o no volátil tal como la memoria de solo lectura (ROM) y/o memoria de acceso aleatorio (RAM). A modo de ejemplo no limitativo, la memoria 1030 también puede incluir un sistema operativo, programas de aplicación, otros módulos de programa y datos de programa. El software se puede incorporar en un disco versátil digital (DVD), disco compacto (CD) u otro medio de almacenamiento extraíble también.

40 Un usuario puede introducir comandos e información en el ordenador 1010 a través de dispositivos 1040 de entrada. Un monitor u otro tipo de dispositivo de visualización está también conectado al bus 1021 de sistema a través de una interfaz, tal como la interfaz 1050 de salida. Además de un monitor, los ordenadores también pueden incluir otros dispositivos de salida periféricos, tales como altavoces y una impresora, que pueden conectarse a través de la interfaz 1050 de salida.

45 El ordenador 1010 puede operar en un entorno de red o distribuido usando conexiones lógicas a uno o más ordenadores remotos, tal como el ordenador 1070 remoto. El ordenador 1070 remoto puede ser un ordenador personal, un servidor, un enrutador, un PC en red, un dispositivo de pares u otro nodo de red común o cualquier otro dispositivo de consumo o transmisión de medios remoto, y puede incluir cualquiera o todos los elementos descritos anteriormente con respecto al ordenador 1010. Las conexiones lógicas representadas en la figura 10 incluyen una red 1071, como una red de área local (LAN) o una red de área amplia (WAN), pero también pueden incluir otras redes/buses. Tales entornos de red son comunes en hogares, oficinas, redes informáticas corporativas, intranet e Internet.

50 Como se ha mencionado anteriormente, aunque se han descrito realizaciones ejemplares en relación con diversos dispositivos informáticos, redes y arquitecturas publicitarias, los conceptos subyacentes pueden aplicarse a cualquier sistema de red y cualquier dispositivo o sistema informático en el que es deseable gestionar la memoria.

55 Existen múltiples maneras de implementar una o más de las realizaciones descritas en el presente documento, por ejemplo, una API adecuada, kit de herramientas, código de controlador, sistema operativo, control, objeto de software independiente o descargable, etc., que permite a las aplicaciones y servicios usar la gestión de memoria. Las realizaciones pueden contemplarse desde la perspectiva de una API (u otro objeto de software), así como desde un objeto de software o hardware que facilita la provisión de gestión de memoria según una o más de las

realizaciones descritas. Diversas implementaciones y realizaciones descritas en el presente documento pueden tener aspectos que están totalmente en hardware, parcialmente en hardware y parcialmente en software, así como en software.

5 La palabra "ejemplar" se usa en el presente documento para indicar que sirve como ejemplo, caso o ilustración. Para evitar dudas, la materia objeto desvelada en el presente documento no está limitada por tales ejemplos. Además, cualquier aspecto o diseño descrito en el presente documento como "ejemplar" no debe interpretarse necesariamente como preferente o ventajoso con respecto a otros aspectos o diseños, ni pretende excluir estructuras y técnicas ejemplares equivalentes conocidas por los expertos en la materia. Además, en la medida en que los términos "incluye", "tiene", "contiene" y otras palabras similares se usan en la descripción detallada o en las reivindicaciones, para evitar dudas, tales términos pretenden ser inclusivos de manera similar al término "que comprende" como una palabra de transición abierta sin excluir ningún elemento adicional u otros elementos.

10 Como se ha mencionado, las diversas técnicas descritas en el presente documento pueden implementarse en relación con hardware o software o, cuando sea apropiado, con una combinación de ambos. Como se usan en el presente documento, los términos "componente", "sistema" y similares pretenden referirse igualmente a una entidad relacionada con el ordenador, ya sea hardware, una combinación de hardware y software, software o software en ejecución. Por ejemplo, un componente puede ser, pero no está limitado a ser, un procedimiento que se ejecuta en un procesador, un procesador, un objeto, un ejecutable, un hilo de ejecución, un programa y/o un ordenador. A modo de ilustración, tanto una aplicación que se ejecuta en el ordenador como el ordenador pueden ser un componente. Uno o más componentes pueden residir en el interior de un procedimiento y/o hilo de ejecución y un componente puede localizarse en un ordenador y/o distribuirse entre dos o más ordenadores.

15 Los sistemas mencionados anteriormente se han descrito con respecto a la interacción entre varios componentes. Se puede apreciar que tales sistemas y componentes pueden incluir aquellos componentes o subcomponentes especificados, algunos de los componentes o subcomponentes especificados, y/o componentes adicionales, y según diversas permutaciones y combinaciones de lo que se ha mencionado anteriormente. Los subcomponentes también pueden implementarse como componentes acoplados comunicativamente a otros componentes en lugar de incluirse en el interior de componentes principales (jerárquicos). Además, se observa que uno o más componentes pueden combinarse en un único componente que proporciona funcionalidad agregada o dividirse en varios subcomponentes separados, y se pueden proporcionar una o más capas intermedias, tal como una capa de gestión, para acoplarse comunicativamente con tales subcomponentes con el fin de proporcionar funcionalidad integrada. Cualquier componente descrito en el presente documento también puede interactuar con uno o más componentes no descritos específicamente en el presente documento, pero generalmente conocidos por los expertos en la materia.

20 En vista de los sistemas ejemplares descritos anteriormente, se pueden apreciar metodologías que se pueden implementar según la materia objeto desvelada con referencia a los diagramas de flujo de las diversas figuras. Aunque con fines de simplicidad de explicación, las metodologías se muestran y describen como una serie de bloques, debe entenderse y apreciarse que la materia objeto reivindicada no está limitada por el orden de los bloques, ya que algunos bloques pueden ocurrir en órdenes diferentes y/o simultáneamente con otros bloques a partir de lo que se representa y describe en el presente documento. Cuando se ilustra un flujo no secuencial o ramificado a través del diagrama de flujo, se puede apreciar que se pueden implementar otras diversas ramificaciones, trayectorias de flujo y órdenes de los bloques que consiguen el mismo resultado o un resultado similar. Además, no todos los bloques ilustrados pueden ser necesarios para implementar las metodologías descritas a continuación.

25 Aunque en algunas realizaciones se ilustra una perspectiva del lado del cliente, para evitar dudas debe entenderse que existe una perspectiva de servidor correspondiente, o viceversa. De manera similar, cuando se practica un procedimiento, se puede proporcionar un dispositivo correspondiente que tiene almacenamiento y al menos un procesador configurado para practicar dicho procedimiento a través de uno o más componentes.

30 Aunque las diversas realizaciones se han descrito en relación con las realizaciones preferentes de las diversas figuras, debe entenderse que pueden usarse otras realizaciones similares o se pueden hacer modificaciones y adiciones a la realización descrita para realizar la misma función sin desviarse de allí. Aún más, se pueden implementar uno o más aspectos de las realizaciones descritas anteriormente en una pluralidad de chips o dispositivos de procesamiento o a través de los mismos, y el almacenamiento puede verse afectado de manera similar a través de una pluralidad de dispositivos. Por lo tanto, la presente invención no debe limitarse a ninguna realización única, sino que debe interpretarse en amplitud y ámbito según las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un procedimiento, que comprende:
 - la identificación (600) de objetos-raíz incluidos en una pila de llamadas, en el que los objetos-raíz comprenden representaciones de números enteros y representaciones de punteros;
 - 5 el etiquetado (610) de las representaciones de números enteros, en el que una representación de números enteros etiquetada se puede distinguir de una representación de puntero no etiquetada;
 - el seguimiento (620) de los objetos-raíz en un conjunto de ubicaciones de memoria correspondientes, en el que se realiza un seguimiento posterior en la representación de puntero no etiquetada, y en el que
 - 10 el seguimiento posterior es omitido en la representación de números enteros etiquetada; y
 - la liberación (630) de memoria asignada a un conjunto de objetos inaccesibles por la pila de llamadas.
2. El procedimiento según la reivindicación 1, en el que el etiquetado comprende la transformación de las representaciones de números enteros en una estructura de datos diferente.
3. El procedimiento según la reivindicación 2, en el que la transformación comprende la dedicación a una etiqueta de al menos un bit en la estructura de datos diferente.
- 15 4. El procedimiento según la reivindicación 1, en el que la liberación comprende la conservación, respectivamente, de objetos accesibles en las ubicaciones de almacenamiento originales de un segmento de memoria.
5. El procedimiento según la reivindicación 4, en el que la conservación comprende el mantenimiento de una separación de objetos del segmento de memoria según un conjunto fijo de límites en el interior del segmento de memoria.
- 20 6. El procedimiento según la reivindicación 1, que comprende además:
 - la ejecución de la pila de llamadas, en el que los objetos llamados se colocan sobre un segmento de memoria; y
 - la comparación de un tamaño del segmento de memoria con un umbral, en el que un lanzamiento de un algoritmo de recolección de elementos no utilizados se basa en si el tamaño del segmento de memoria excede el umbral.
- 25 7. El procedimiento según la reivindicación 6, en el que la ejecución comprende la compilación de un programa JavaScript.
8. El procedimiento según la reivindicación 6, en el que la ejecución comprende la modificación de un modelo de objeto de documento.
9. Un dispositivo (500) de gestión de memoria, que comprende:
 - 30 una memoria (520) que tiene componentes ejecutables por ordenador almacenados en la misma; y
 - un procesador (510) acoplado mediante comunicación a la memoria, estando el procesador configurado para ejecutar los componentes ejecutables por ordenador, comprendiendo los componentes ejecutables por ordenador:
 - 35 un componente (530) de barrido configurado para identificar objetos-raíz incluidos en una pila de llamadas, en el que los objetos-raíz comprenden representaciones de números enteros y representaciones de punteros;
 - un componente (540) de etiquetado configurado para etiquetar las representaciones de números enteros, en el que una representación de números enteros etiquetada se puede distinguir de una representación de puntero no etiquetada;
 - 40 un componente (550) de seguimiento configurado para localizar los objetos-raíz en un conjunto de ubicaciones de memoria correspondientes, en el que se realiza un seguimiento posterior en la representación de puntero no etiquetada, y en el que el seguimiento posterior es omitido en la representación de número entero etiquetada; y
 - 45 un componente (560) de reciclaje configurado para liberar memoria asignada a un conjunto de objetos inaccesibles por la pila de llamadas.
10. El dispositivo (500) de gestión de memoria según la reivindicación 9, en el que el componente (540) de etiquetado está configurado para transformar las representaciones de números enteros en una estructura de datos diferente.
11. El dispositivo (500) de gestión de memoria según la reivindicación 10, en el que el componente (540) de etiquetado está configurado para dedicar a una etiqueta al menos un bit en la estructura de datos diferente.
- 50 12. El dispositivo (500) de gestión de memoria según la reivindicación 9, en el que el componente (560) de reciclaje está configurado para conservar, respectivamente, objetos accesibles en ubicaciones de almacenamiento originales de un segmento (200; 320; 420) de memoria.

13. El dispositivo (500) de gestión de memoria según la reivindicación 12, en el que el componente (560) de reciclaje está configurado para mantener una separación de objetos de segmentos de memoria según un conjunto fijo de límites en el interior del segmento (200; 320; 420) de memoria.

5 14. El dispositivo (500) de gestión de memoria según la reivindicación 9, en el que el componente (560) de reciclaje está configurado para colocar objetos llamados por la pila de llamadas sobre un segmento (200; 320; 420) de memoria, y en el que el componente de reciclaje está configurado además para lanzar un algoritmo de recolección de elementos no utilizados que se basa en si un tamaño del segmento de memoria excede un umbral.

15. El dispositivo (500) de gestión de memoria según la reivindicación 9, en el que la pila de llamadas está asociada con una compilación de un programa JavaScript.

10

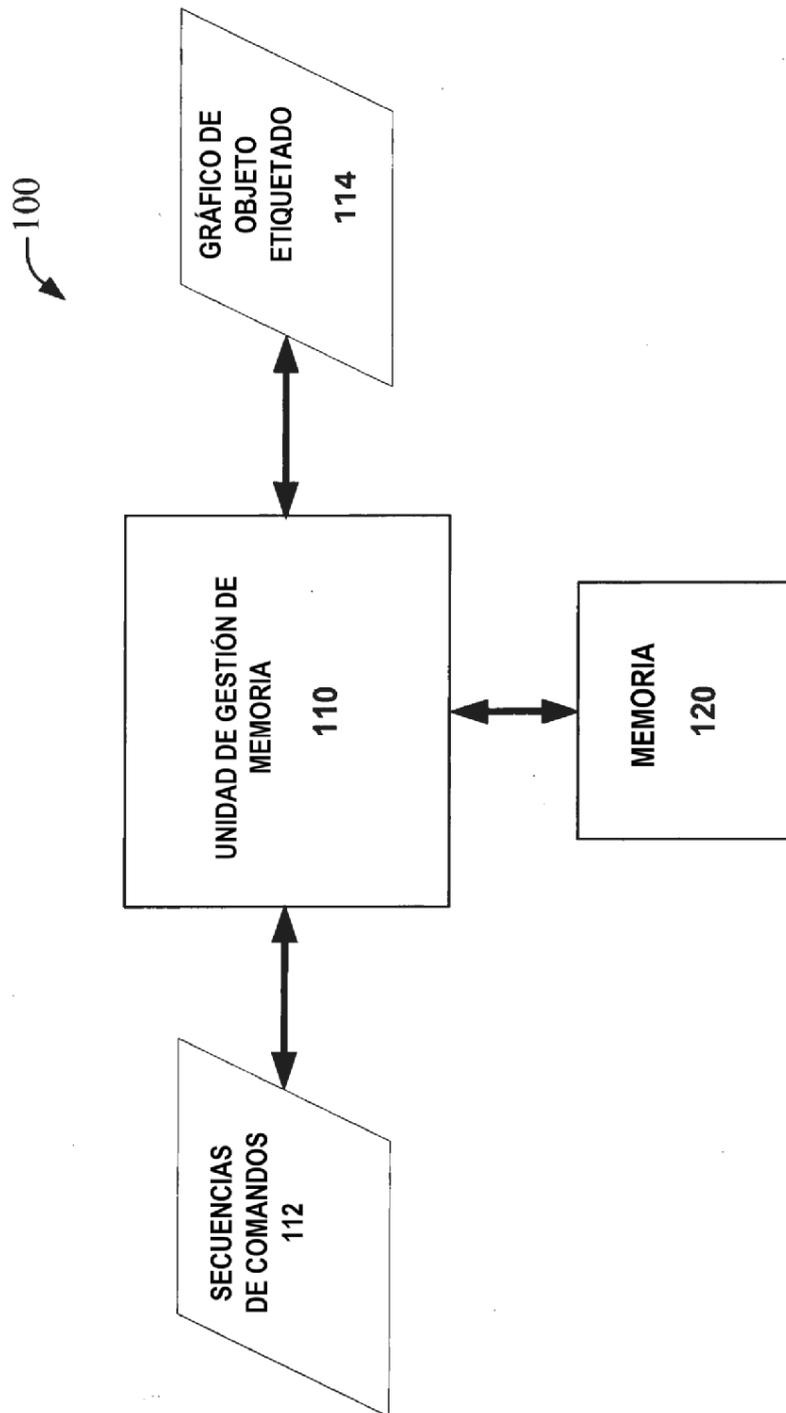


FIG. 1

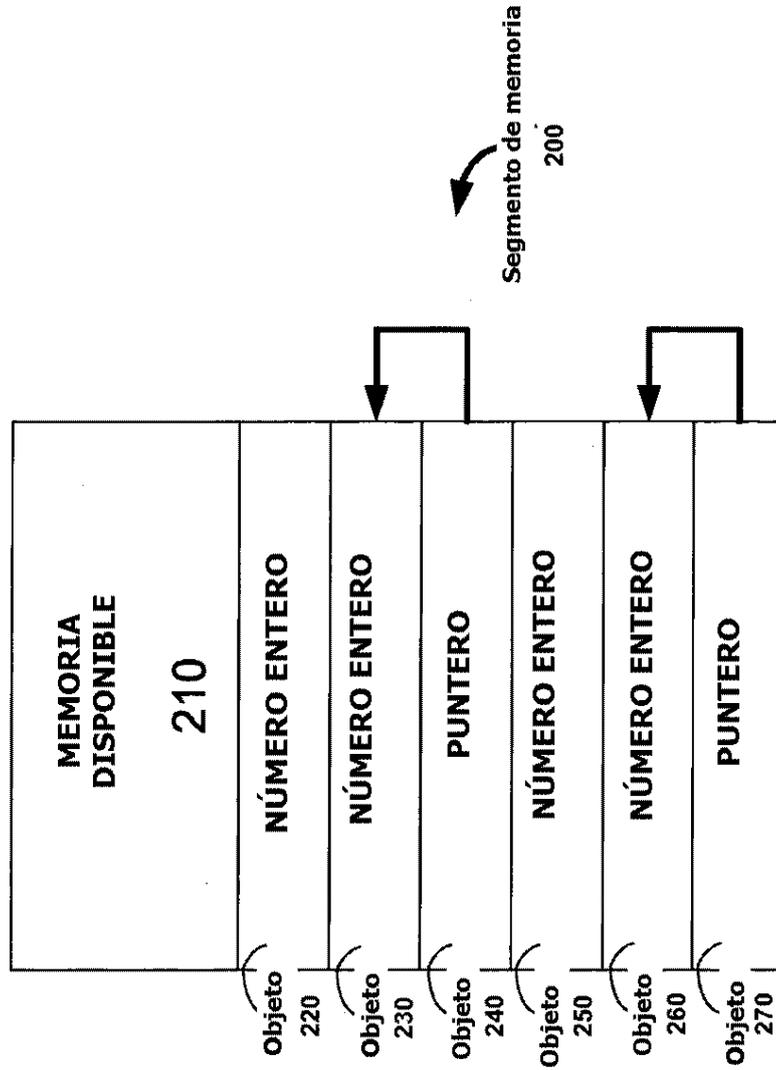


FIG. 2

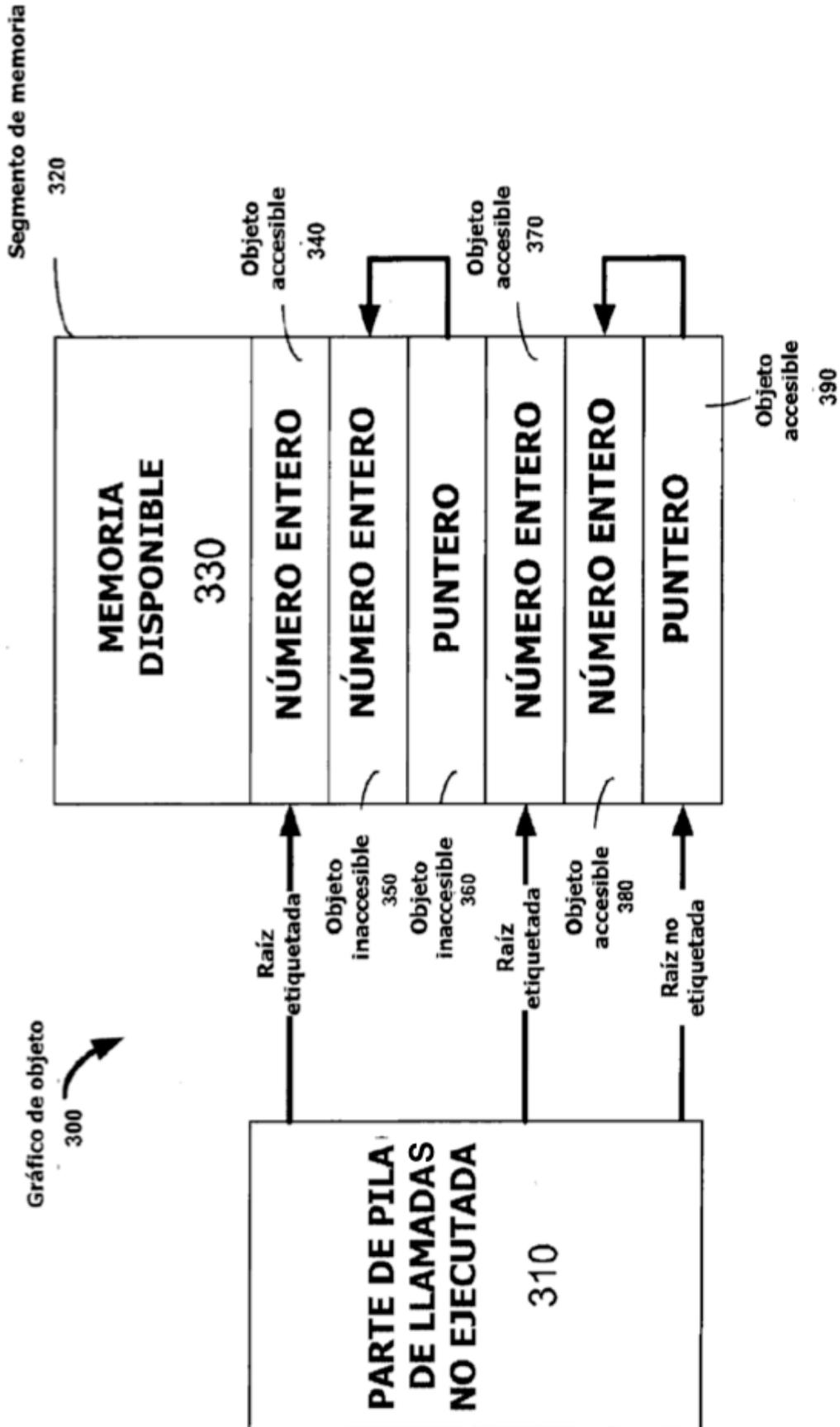


FIG. 3

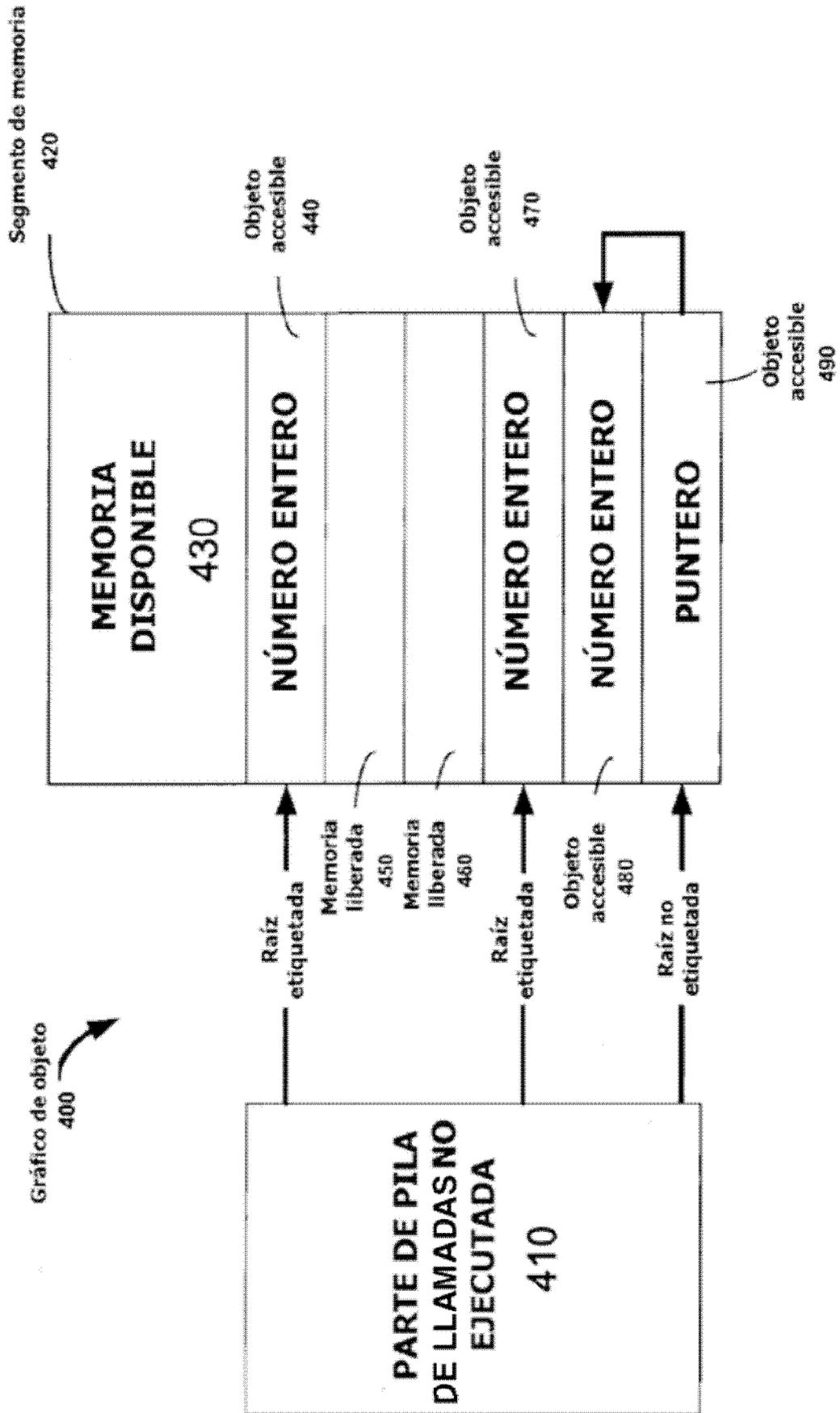
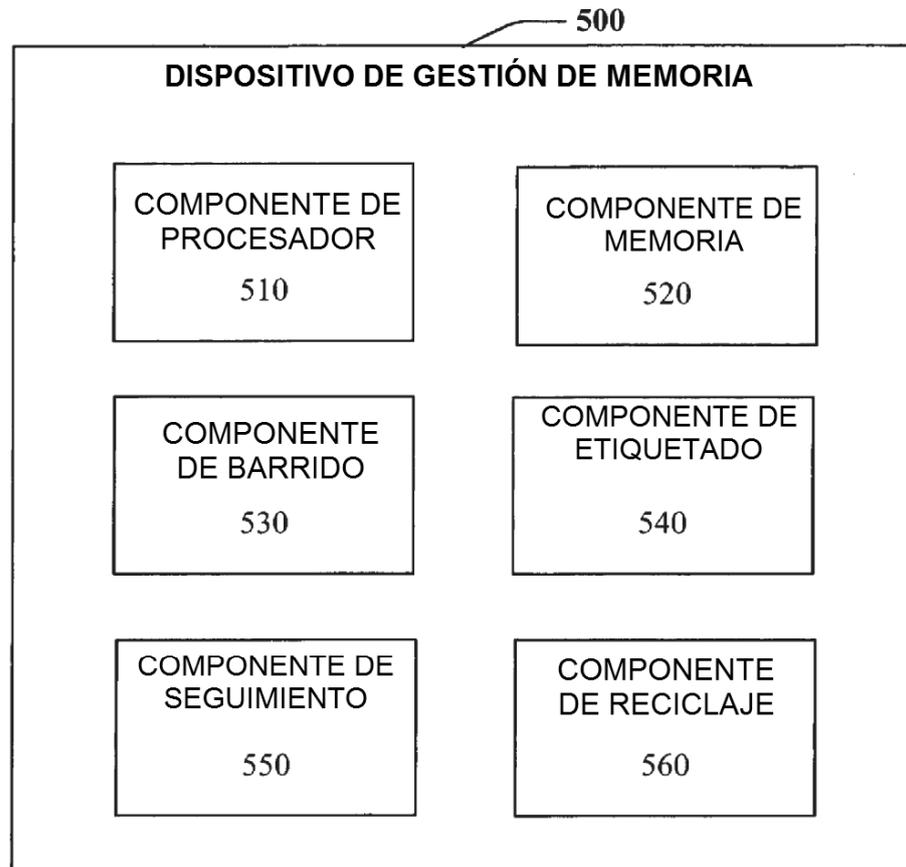


FIG. 4



API: interfaz de programación de aplicaciones

FIG. 5

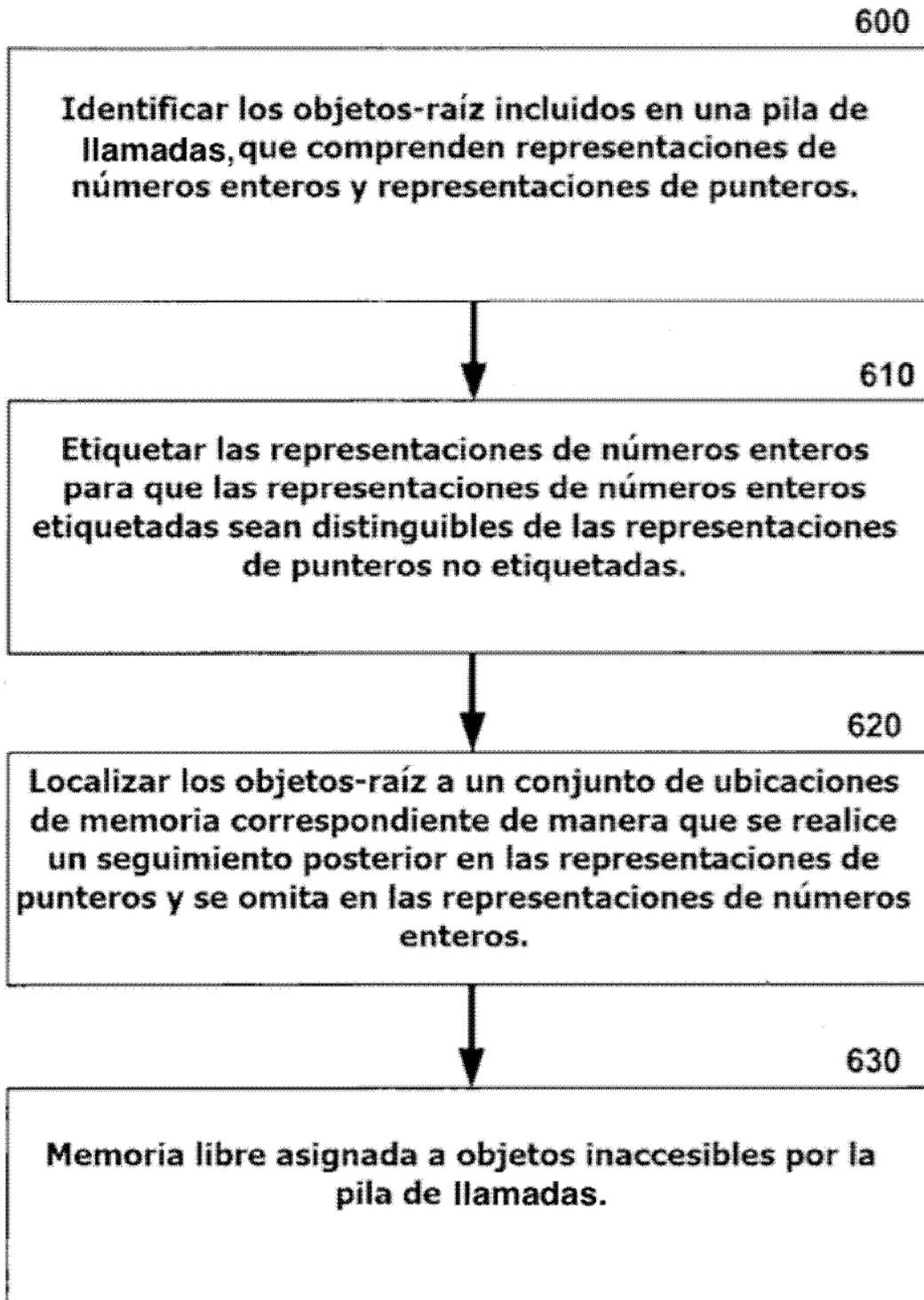


FIG. 6

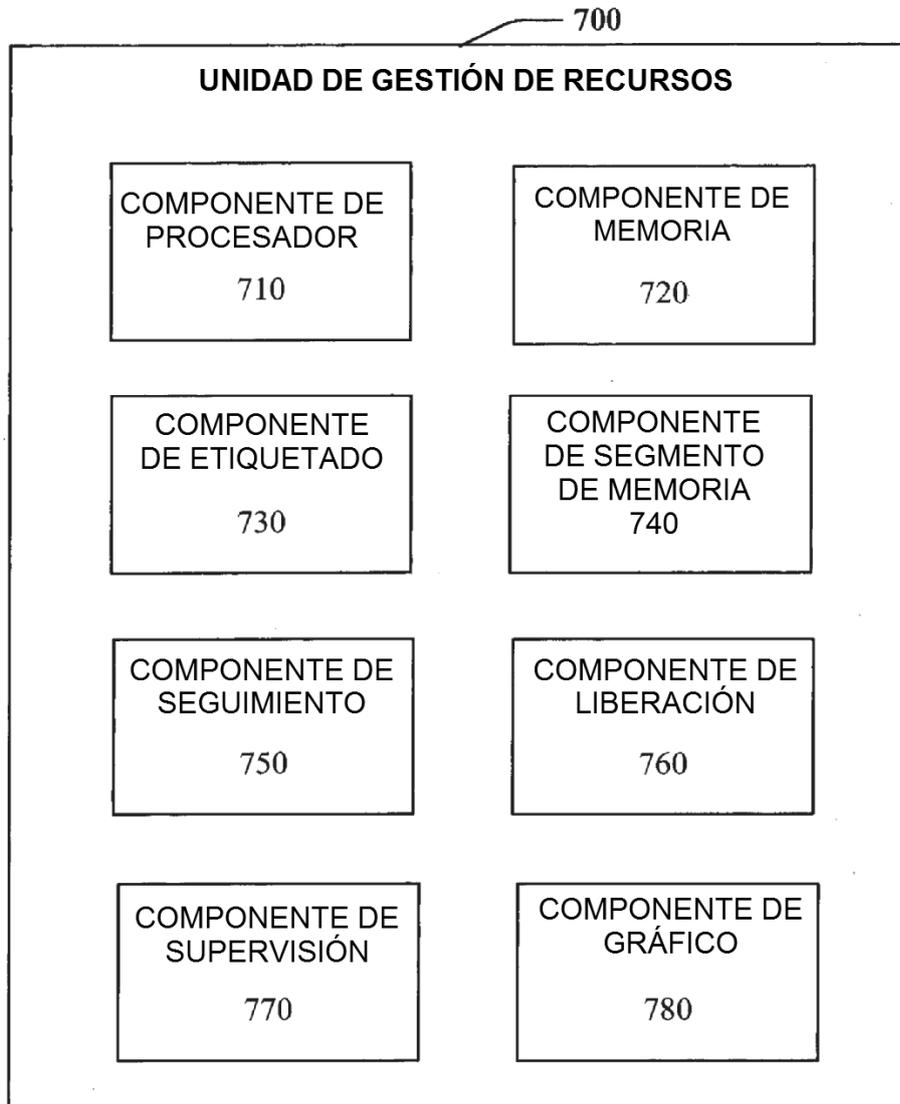
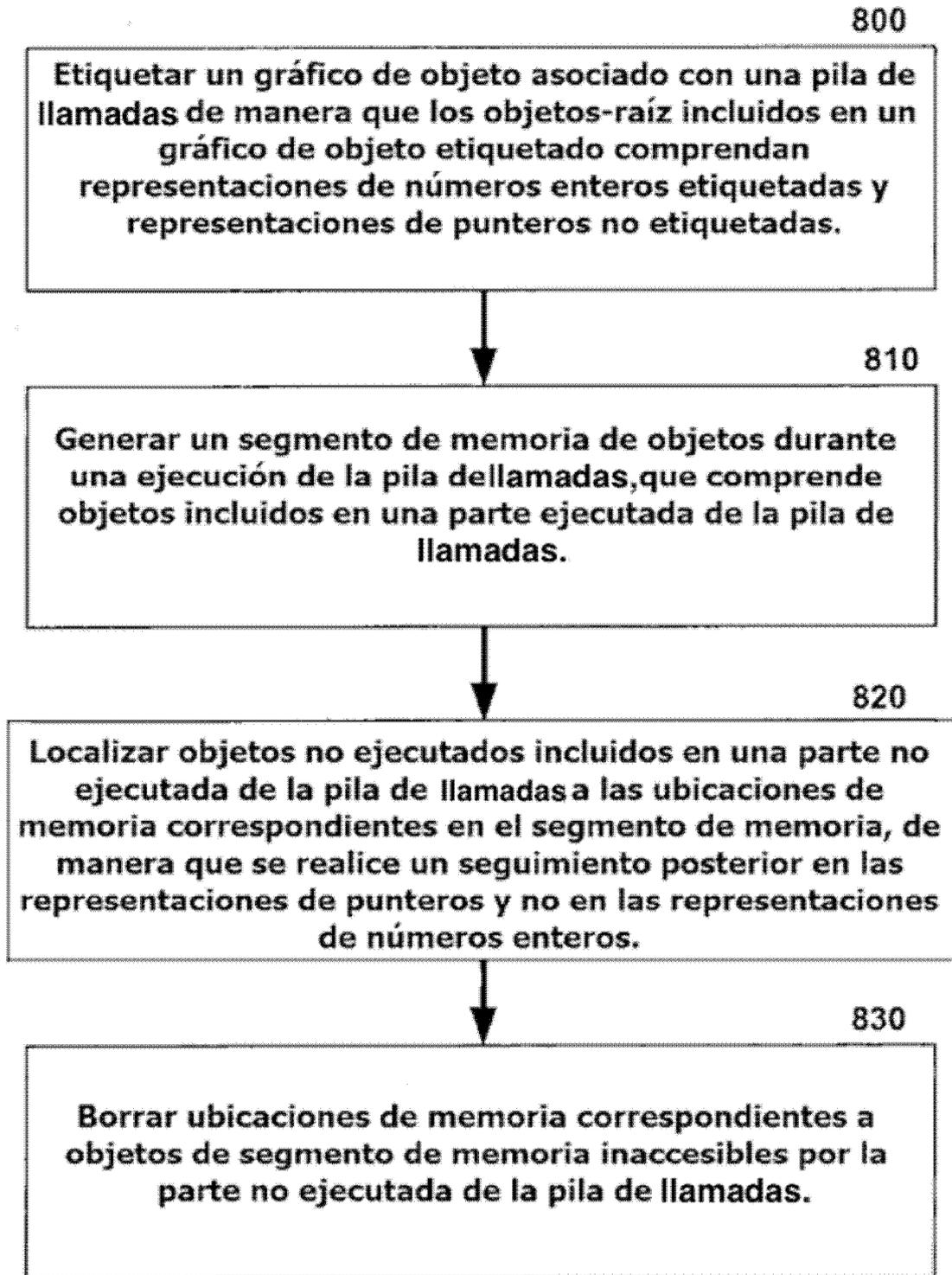


FIG. 7

**FIG. 8**

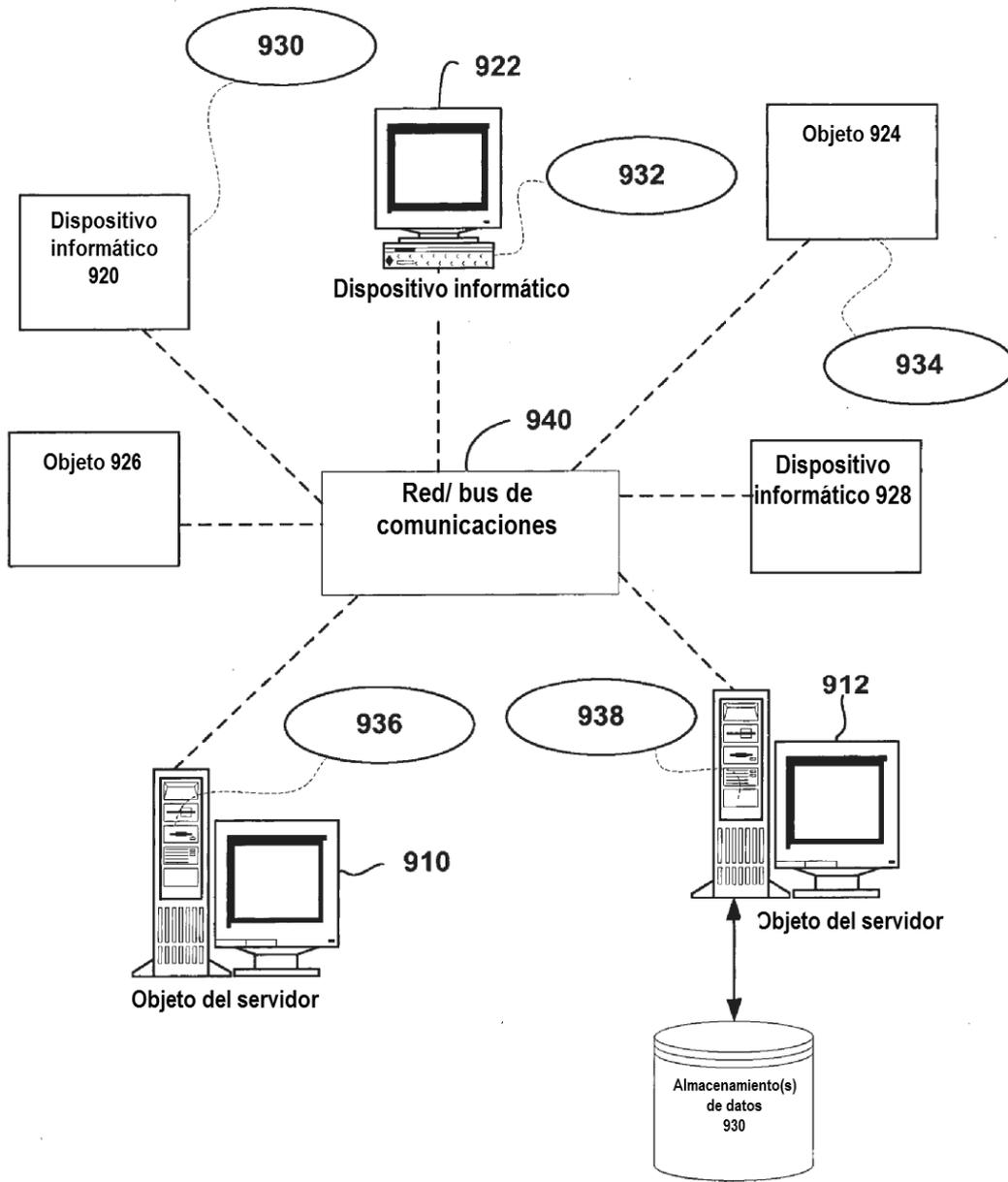


FIG. 9

Entorno informático 1000

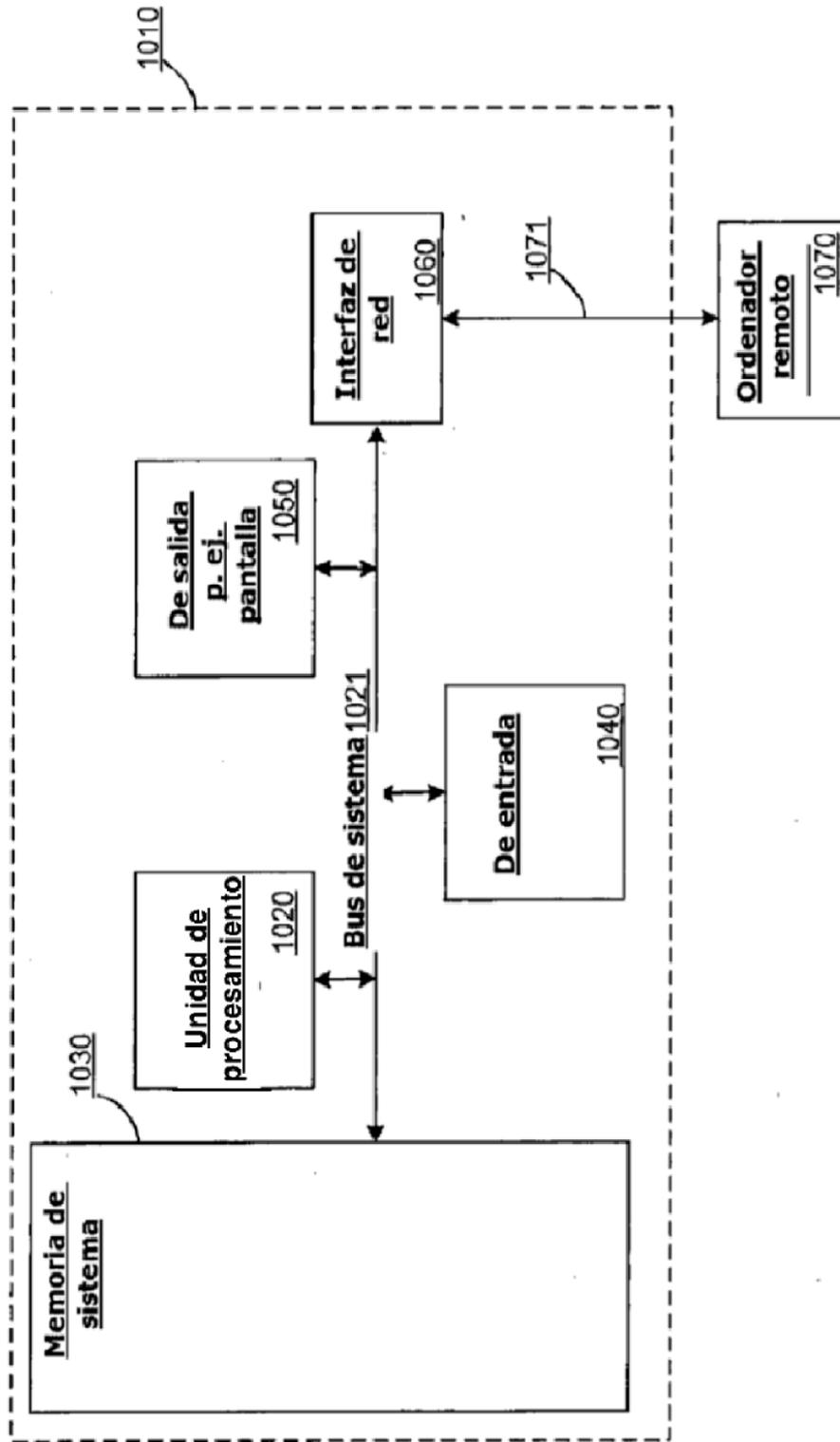


FIG. 10