

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 808 924**

51 Int. Cl.:

H04L 29/08 (2006.01)

G06F 9/451 (2008.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **16.12.2016 PCT/CN2016/110259**

87 Fecha y número de publicación internacional: **06.07.2017 WO17114182**

96 Fecha de presentación y número de la solicitud europea: **16.12.2016 E 16880972 (1)**

97 Fecha y número de publicación de la concesión europea: **01.07.2020 EP 3399721**

54 Título: **Método y dispositivo de visualización de datos de interfaz**

30 Prioridad:

30.12.2015 CN 201511021196

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

02.03.2021

73 Titular/es:

**ADVANCED NEW TECHNOLOGIES CO., LTD.
(100.0%)
Cayman Corporate Centre, 27 Hospital Road
George Town, Grand Cayman KY1-9008, KY**

72 Inventor/es:

**HE, SANYUAN;
RAO, JUNXUE;
WANG, FEI;
WENG, YUEDONG;
ZHU, DELIANG;
SUN, XI;
WEN, XUELONG;
XU, WENTING;
CHEN, RUI;
XI, JIANGBO y
DENG, XUFENG**

74 Agente/Representante:

VIDAL GONZÁLEZ, Maria Ester

ES 2 808 924 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 y dispositivo de visualización de datos de interfaz

5 Campo técnico:

La presente solicitud se refiere a tecnologías de presentación de datos y, en particular, a un método y dispositivo de presentación de datos de interfaz.

10 Antecedentes

Una plataforma de interfaz de una aplicación cliente se escribe generalmente por códigos nativos. La plataforma de interfaz puede permanecer sin cambios después de establecerse de antemano en base a las necesidades del servicio. La plataforma y el estilo de cada interfaz se pueden adaptar a una estructura de datos de servicio correspondiente. Si es necesario cambiar la plataforma de interfaz en base a un cambio de servicio, o si se necesita agregar una nueva plataforma de interfaz para agregar nuevos datos de servicio, la aplicación cliente generalmente necesita ser recompilada, descargada y reinstalada. Si un usuario no actualiza la aplicación cliente, los cambios o adiciones de datos de servicio no se pueden presentar al usuario. El documento US 2006/0168536 describe la generación de interfaces gráficas de usuario uniformes independientes del dispositivo.

20 Resumen

La presente solicitud proporciona un método y dispositivo de presentación de datos de interfaz, para actualizar de manera dinámica la interfaz de una aplicación cliente.

25 De acuerdo con un primer aspecto de una implementación de la presente solicitud, se proporciona un método de presentación de datos de interfaz que se ejecuta en una aplicación cliente, donde el método incluye las siguientes etapas: analizar, para obtener un nodo de vista virtual, datos estructurados que son de una interfaz plantilla y que son entregados por un servidor, donde el servidor genera los datos estructurados de la plantilla de interfaz al analizar los datos de plantilla de interfaz escritos en un lenguaje de marcado de protocolo general, y los datos de plantilla de interfaz incluyen una etiqueta común para describir un widget; vincular datos de servicio al nodo de vista virtual; y calcular la información de ubicación del widget en base al nodo de vista virtual al que están vinculados los datos del servicio, y representar una vista basada en la información de ubicación del widget.

35 De acuerdo con un segundo aspecto de una implementación de la presente solicitud, se proporciona un método de presentación de datos de interfaz que se ejecuta en un servidor, donde el método incluye las siguientes etapas: analizar, para obtener datos estructurados, los datos de plantilla de interfaz escritos en un lenguaje de marcado de protocolo general, donde los datos de la plantilla de interfaz incluyen una etiqueta común para describir un widget; y entregar los datos estructurados a una aplicación cliente, de manera que la aplicación cliente muestre el widget en una vista.

40 De acuerdo con un tercer aspecto de una implementación de la presente solicitud, se proporciona un dispositivo de presentación de datos de interfaz ubicado en una aplicación cliente, e incluye: un módulo de motor de análisis, configurado para analizar, para obtener un nodo de vista virtual, datos estructurados que son de una plantilla de interfaz y que son entregados por un servidor, donde el servidor genera los datos estructurados de la plantilla de interfaz analizando los datos de plantilla de interfaz escritos en un lenguaje de marcado de protocolo general, y los datos de plantilla de interfaz incluyen una etiqueta común para describir un widget; un módulo de enlace, configurado para vincular datos de servicio al nodo de vista virtual; y un módulo de motor de diseño, configurado para calcular la información de ubicación del widget en base al nodo de vista virtual al que están vinculados los datos del servicio; y un módulo de motor de representación, configurado para representar una vista basada en la información de ubicación del widget.

45 De acuerdo con un cuarto aspecto de una implementación de la presente solicitud, se proporciona un dispositivo de presentación de datos de interfaz ubicado en un servidor, e incluye: un módulo de motor de análisis, configurado para analizar, para obtener datos estructurados, datos de plantilla de interfaz escritos en un lenguaje de marcado de protocolo general, donde los datos de la plantilla de interfaz incluyen una etiqueta común para describir un widget; y un módulo de comunicaciones, configurado para suministrar los datos estructurados a una aplicación cliente, de modo que la aplicación cliente represente el widget en una vista.

50 La presente solicitud implementa la plataforma de interfaz de la aplicación cliente mediante el uso de la tecnología de plantilla de interfaz en lugar del código nativo, y en consecuencia proporciona una solución para que la aplicación cliente muestre los datos del servicio mediante el uso de la plantilla de interfaz. Como tal, no solo la interfaz se puede actualizar sin actualizar la aplicación cliente, sino que también el servidor puede analizar los datos de la plantilla de interfaz para obtener los datos estructurados, de manera que los desarrolladores puedan diseñar la plantilla de interfaz en el lenguaje de marcado general. Por lo tanto, se puede eliminar el requisito de que los

desarrolladores aprendan protocolos patentados para que las soluciones de la presente solicitud sean más universales.

Breve descripción de los dibujos

5 La Figura 1 es un diagrama de flujo que ilustra un método de presentación de datos de interfaz, de acuerdo con una implementación de la presente solicitud;

la Figura 2 es un diagrama que ilustra una secuencia de temporización de interacción de información en un ejemplo de aplicación, de acuerdo con la presente solicitud;

10 la Figura 3 es un diagrama de arquitectura que ilustra el hardware de un dispositivo de presentación de datos de interfaz, de acuerdo con una implementación de la presente solicitud;

15 la Figura 4 es un diagrama de bloques lógico de software que ilustra un dispositivo de presentación de datos de interfaz, de acuerdo con una implementación de la presente solicitud; y

la Figura 5 es un diagrama de bloques lógico de software que ilustra un dispositivo de presentación de datos de interfaz, de acuerdo con otra implementación de la presente solicitud.

20 Descripción de las modalidades

Las implementaciones a modo de ejemplo se describen en detalle aquí y se presentan en los dibujos adjuntos. Cuando la siguiente descripción se refiere a los dibujos adjuntos, a menos que se especifique de cualquier otra manera, los mismos números en diferentes dibujos adjuntos representan los mismos elementos o elementos similares. Las implementaciones descritas en los siguientes ejemplos de implementación no representan todas las implementaciones de acuerdo con la presente solicitud. Por el contrario, son solo ejemplos de dispositivos y métodos que se describen en detalle en las reivindicaciones adjuntas y que están de acuerdo con algunos aspectos de la presente solicitud.

25 Los términos usados en la presente solicitud se usan simplemente con el fin de ilustrar las implementaciones, y no están destinados a limitar la presente solicitud. Los términos "un", "tal" y "el" de formas singulares usados en la presente solicitud y las reivindicaciones adjuntas también pretenden incluir formas plurales, a menos que se especifique de cualquier otra manera en el contexto. Se entenderá además que el término "y/o" usado en la presente descripción, indica e incluye cada una y cualquiera de las combinaciones posibles de uno o más de los artículos listados asociados.

30 Debe entenderse que, aunque los términos "primero", "segundo", "tercero", etc. pueden usarse en la presente solicitud para describir información diversa, la información no está limitada por los términos. Estos términos se usan simplemente para diferenciar información del mismo tipo. Por ejemplo, sin apartarse del alcance de la presente solicitud, la primera información también puede denominarse segunda información y, de manera similar, la segunda información también puede denominarse la primera información. En dependencia del contexto, por ejemplo, las palabras "si" usadas aquí pueden explicarse como "mientras" o "cuándo" o "en respuesta a la determinación".

35 En la presente solicitud, una plantilla de interfaz se puede separar de una aplicación cliente, para ayudar a la actualización dinámica de la plantilla de interfaz y permitir que un desarrollador de aplicaciones escriba la plantilla de interfaz en función de un lenguaje de marcado de protocolo general. La aplicación cliente se puede cargar en varios dispositivos inteligentes, como un terminal móvil, una tableta o un ordenador. Las soluciones no se limitan a un tipo específico de sistema operativo en un dispositivo inteligente. Por ejemplo, el sistema operativo puede ser un sistema iOS, un sistema Android, un sistema Windows, un sistema OS X, etc. El lenguaje de marcado de protocolo general puede ser el Lenguaje de marcado de hipertexto (HTML). Ciertamente, no se excluyen otros idiomas especificados en protocolos generales estándar adecuados para hacer una interfaz.

40 La plantilla de interfaz en la presente solicitud puede ser una plataforma de interfaz que incluye relación de diseño, estilo de visualización, lógica de control de operación de varios elementos comunes, pero no incluye datos de servicio. La plantilla de interfaz se puede implementar mediante el uso de HTML, hojas de estilo en cascada (CSS) y Javascript (JS). El elemento común puede incluir una etiqueta común para datos de plantilla de interfaz en código fuente HTML. Un widget se describe mediante el uso de la etiqueta común, por ejemplo, una etiqueta Body, una etiqueta Div, una etiqueta Title, una etiqueta Label y una etiqueta Img. El elemento común también puede incluir un elemento que presente un estilo de visualización. El estilo de visualización se puede implementar mediante el uso de CSS. El diseño se especifica mediante el uso de CSS para una interfaz escrita en HTML. Ciertamente, no se excluyen otros métodos para especificar el diseño de la interfaz. JS puede usarse para controlar el contenido de visualización, los efectos de visualización, el ajuste de la relación de diseño de las etiquetas comunes y para procesar las operaciones del usuario.

45 Después de configurar una plantilla de interfaz escrita por un desarrollador de aplicaciones en un servidor, el servidor convierte los datos de la plantilla de interfaz en datos estructurados de la plantilla de interfaz y envía los

datos estructurados de la plantilla de interfaz a una aplicación cliente. Con referencia a la Figura 1, después de que la aplicación cliente recibe los datos estructurados, un proceso de trabajo puede incluir las siguientes etapas:

- 5 S101. La aplicación cliente analiza los datos estructurados de la interfaz.
- S101. La aplicación cliente analiza los datos estructurados de la plantilla de interfaz enviados por el servidor para obtener un nodo de vista virtual.
- S102. Enlazar los datos del servicio al nodo de vista virtual.
- S103. Calcular la información de ubicación de un widget en base al nodo de vista virtual al que están vinculados los datos del servicio.
- 10 S104. Presentar una vista en base a la información de ubicación del widget.

Un proceso de trabajo del servidor puede incluir un proceso de conversión de datos de plantilla de interfaz, un proceso de distribución de datos estructurados de la plantilla de interfaz y un proceso de intercambio de datos de servicio con la aplicación cliente.

15 Para resolver el problema del bajo rendimiento de análisis de HTML, antes de que el servidor envíe los datos de la plantilla de interfaz a la aplicación cliente, el servidor primero analiza los datos de la plantilla de interfaz para obtener los datos estructurados, y luego envía los datos estructurados a la aplicación cliente, de esta manera, se puede evitar el proceso de análisis del HTML por parte de la aplicación cliente. Los datos estructurados seleccionados
 20 pueden tener una estructura de datos compacta que puede expresar directamente un modelo de objeto, como un formato JSON y un formato XML. El formato JSON se usa como ejemplo. El HTML se puede analizar para obtener un modelo de objeto de documento (DOM). Un nodo DOM almacena información relacionada de una etiqueta común HTML. El nodo DOM describe una ubicación relativa de un widget en la interfaz, descrito mediante el uso de la etiqueta común. Dado que un árbol de sintaxis del formato JSON
 25 es más simple que un árbol de sintaxis del HTML, la aplicación cliente puede analizar los datos en el formato JSON de manera más eficiente que los datos HTML.

Cada plantilla de interfaz tiene un identificador de interfaz único para distinguir entre diferentes plantillas de interfaz. Después de analizar los datos de la plantilla de interfaz para obtener los datos estructurados, el servidor distribuye
 30 los datos estructurados a la aplicación cliente, de modo que la aplicación cliente muestra subsecuentemente los datos del servicio mediante el uso de los datos estructurados. Los datos estructurados de la plantilla de interfaz incluyen un identificador de interfaz.

Un proceso de distribución de una plantilla de interfaz actualizada a la aplicación cliente puede variar con diferentes escenarios de servicio. Por ejemplo, después de actualizar la plantilla de interfaz, el servidor puede distribuir
 35 activamente, a la aplicación cliente, nuevos datos estructurados obtenidos al analizar la plantilla de interfaz, o puede distribuir los datos estructurados a la aplicación cliente según sea necesario, etc.

Un proceso de trabajo de la aplicación cliente puede incluir un proceso de análisis y almacenamiento de los datos estructurados de la plantilla de interfaz, un proceso de intercambio de la plantilla y un proceso de representación de
 40 una vista.

Después de que el servidor envía los datos estructurados de la plantilla de interfaz a la aplicación cliente, la aplicación cliente analiza los datos estructurados de la plantilla de interfaz para generar nodos de vista virtual de
 45 estructura de árbol para el almacenamiento. Cada nodo de vista virtual puede almacenar información básica de un nodo DOM en el HTML, como información relacionada de la etiqueta común, y puede incluir además información de tipo en la aplicación cliente correspondiente, información sobre un evento de respuesta en la aplicación cliente, etc. La información de ubicación del widget descrita mediante el uso de la etiqueta común HTML se puede calcular utilizando el nodo de vista virtual.

50 Después de que se desencadena un procedimiento de intercambio de datos de servicio, la aplicación cliente obtiene los datos del servicio del servidor, y el servidor puede suministrar un identificador de plantilla que corresponde a los datos del servicio a la aplicación cliente al suministrar los datos del servicio.

55 A manera de ejemplo, el proceso de vincular los datos de servicio a la plantilla de interfaz se puede implementar mediante el uso de un script JS en una plantilla de interfaz impulsada por el motor JS. Los datos del servicio son procesados por el motor JS. El motor JS ejecuta el script JS para insertar los datos del servicio en un nodo de vista virtual correspondiente. Por ejemplo, se puede hacer referencia al siguiente proceso: 1. Si los atributos y métodos en el código nativo (denominado en lo sucesivo Native) de un sistema operativo del dispositivo (por ejemplo, un sistema
 60 iOS o un sistema Android) no existen en el JS de la plantilla de interfaz, se puede extender una interfaz del motor JS, para vincular los atributos y métodos nativos al motor JS. La interfaz extendida del motor JS puede incluir, por ejemplo, una función de entrada JS para recibir y procesar los datos del servicio, o una interfaz de un motor de representación para llamar a Native. 2. El motor JS transmite los datos del servicio a la función de entrada JS. 3. Los datos del servicio se insertan en el nodo de vista virtual correspondiente mediante el uso de la función de entrada JS.

En un ejemplo, se puede instruir al motor de representación, mediante el uso del motor JS y la interfaz del motor de representación para llamar a Native, para activar el proceso de representación de la vista. En el proceso, la aplicación cliente puede representar la vista mediante el uso de una pluralidad de métodos. Por ejemplo, después de calcular la información de ubicación de cada widget, el widget se puede representar en la ubicación correspondiente de la vista. En este caso, un efecto de presentación de la interfaz es que los widgets se presentan al usuario uno por uno. En otro ejemplo, después de que se calcula en una memoria la información de ubicación de los widgets en una vista completa, la información de ubicación de los widgets se puede obtener de la memoria, los widgets se representan juntos en la vista y la vista completa se presenta al usuario, para que el tiempo de representación de la visualización se reduzca considerablemente.

Puede haber un proceso de actualización de los datos estructurados de la plantilla de interfaz obtenida por la aplicación cliente del servidor. Se pueden diseñar diferentes procedimientos de implementación para enviar datos estructurados de la plantilla de interfaz actualizada a la aplicación cliente en función de diferentes escenarios de servicio. Por ejemplo, después de actualizar la plantilla de la interfaz, el servidor puede enviar activamente los datos estructurados a la aplicación cliente. Alternativamente, después de activar un procedimiento de servicio, la aplicación cliente envía una solicitud al servidor cuando los datos del servicio deben presentarse mediante el uso de una nueva plantilla de interfaz. Por lo tanto, como ejemplo, un proceso de solicitud de datos estructurados de la nueva plantilla de interfaz del servidor puede incluir: Después de obtener los datos de servicio del servidor, la aplicación cliente puede buscar la plantilla de interfaz que corresponde a los datos de servicio en base a un identificador de interfaz en los datos de servicio; y si no se identifica una plantilla de interfaz, solicitar al servidor los datos estructurados de la plantilla de interfaz que corresponden al identificador de interfaz.

Después de recibir los datos estructurados actualizados, la aplicación cliente analiza los datos estructurados para obtener un nodo de vista virtual y actualiza la información relacionada almacenada en el nodo de vista virtual. El nodo de vista virtual está vinculado a los datos del servicio en base a la información actualizada, y el nodo de vista virtual al que están vinculados los datos del servicio se pasa al motor de representación mediante el uso de la interfaz del motor de representación para llamar a Native, para volver a dibujar la vista.

El siguiente es un ejemplo de implementación de la solución en un escenario de aplicación. En el ejemplo de aplicación, en base a diferentes funciones, las aplicaciones cliente se clasifican en una aplicación cliente de servicio y una aplicación cliente de interfaz, y los servidores se clasifican en un servidor de servicio y un servidor de interfaz. La aplicación cliente de servicio y la aplicación cliente de interfaz se pueden integrar y cargar en una misma App en un terminal móvil. El servidor de interfaz incluye un motor de análisis HTML y un módulo de administración de plantillas de interfaz. La aplicación cliente de interfaz incluye un motor de análisis de datos estructurados de plantilla de interfaz, un motor JS, un motor de diseño y un motor de representación.

Con referencia a la Figura 2, en la etapa 201, el servidor procesa previamente los datos de la plantilla de interfaz.

El servidor utiliza el motor de análisis HTML para analizar los datos de la plantilla de interfaz para obtener datos estructurados basados en un formato JSON. Una etiqueta común HTML en los datos de la plantilla de interfaz, un atributo CSS asociado con la etiqueta común y un script JS se analizan y luego se almacenan en un mismo archivo JSON del módulo de administración de plantilla de interfaz en forma de un nodo de árbol DOM. El módulo de administración de plantillas de interfaz realiza la adición, eliminación o modificación en estos archivos JSON.

En la etapa 202, la aplicación cliente de interfaz analiza los datos JSON de la plantilla de interfaz.

Después de recibir los datos JSON de la plantilla de interfaz, la aplicación cliente de interfaz utiliza el motor de análisis de datos estructurados de plantilla de interfaz para analizar los datos JSON de la plantilla de interfaz para obtener el nodo de vista virtual. La información como la información básica del nodo DOM de la plantilla de interfaz, la información de tipo en una aplicación y la información sobre un evento de respuesta en la aplicación se almacenan en el nodo de vista virtual.

En la etapa 203, la aplicación cliente de servicio intercambia datos de servicio con el servidor de servicio.

La aplicación cliente de servicio envía una solicitud al servidor de servicio para obtener los datos del servicio. El servidor de servicio encapsula los datos de servicio necesarios en un formato predeterminado y agrega una ID de la plantilla de interfaz para enviar los datos de servicio en forma de datos JSON a la aplicación cliente de servicio.

En la etapa 204, los datos del servicio están vinculados al nodo de vista virtual.

Después de recibir los datos del servicio, la aplicación cliente de servicio reenvía los datos del servicio al motor JS de la aplicación cliente de interfaz para su procesamiento. El motor JS de la aplicación cliente de interfaz busca los datos JSON de la plantilla de interfaz correspondiente en base al ID de la plantilla de interfaz, y transmite los datos de servicio a una función de entrada JS de la plantilla de interfaz para su procesamiento. La función de entrada JS divide los datos del servicio y luego inserta los datos del servicio en el nodo de vista virtual correspondiente para el enlace. Por ejemplo, los datos de servicio relacionados con una etiqueta Body están vinculados a un nodo de vista

virtual que almacena la etiqueta Body, y los datos de servicio relacionados con una etiqueta Div están vinculados a un nodo de vista virtual que almacena la etiqueta Div.

5 En la etapa actual, si no se identifican datos JSON de una plantilla de interfaz correspondiente en base a la ID de la plantilla de interfaz, la aplicación cliente de la interfaz necesita solicitar nuevamente los datos JSON de la plantilla de interfaz que corresponden a la ID de la plantilla de interfaz del servidor de interfaz.

En la etapa 205, el motor de diseño calcula la ubicación de un widget.

10 El motor de diseño calcula la ubicación de cada widget mediante el uso de la información almacenada en cada nodo de vista virtual y almacena la información de ubicación de cada widget en una memoria.

En la etapa 206, se representa una vista.

15 En este ejemplo, después de calcular las ubicaciones de los widgets, cuando se presenta la vista, se instruye al motor de representación, mediante el uso del motor JS, para realizar la representación de la vista. El motor de representación de vistas obtiene de la memoria la ubicación calculada de cada widget en la vista completa y presenta la vista completa.

20 En correspondencia a la implementación del método de presentación de datos de interfaz, la presente solicitud proporciona además una implementación de un dispositivo de presentación de datos de interfaz.

25 La implementación del dispositivo de presentación de datos de interfaz en la presente solicitud se puede aplicar a un terminal inteligente y un servidor. La implementación del dispositivo se puede implementar mediante el uso de software, o se puede implementar mediante el uso de hardware o una combinación de software y hardware. La implementación de software se usa como ejemplo. Como dispositivo lógico, el dispositivo se forma leyendo una instrucción de programa de ordenador correspondiente en una memoria no volátil a una memoria por un procesador del terminal inteligente o el servidor donde se encuentra el dispositivo. Desde una perspectiva de hardware, como se muestra en la Figura 3, la Figura 3 es un diagrama estructural que ilustra el hardware donde se encuentra un dispositivo de presentación de datos en la presente solicitud. Además del procesador, la memoria, la interfaz de red y la memoria no volátil que se muestran en la Figura 3, el terminal inteligente o el servidor donde se encuentra el dispositivo en esta implementación pueden incluir otro hardware basado en las funciones reales del terminal inteligente y el servidor. Los detalles no se describen aquí para evitar la duplicación.

35 Con referencia a la Figura 4, un dispositivo de presentación de datos de interfaz 400 se puede ubicar en una aplicación cliente en una implementación, e incluye un módulo de motor de análisis 401, un módulo de enlace 402, un módulo de motor de diseño 403 y un módulo de motor de representación 404.

40 El módulo del motor de análisis 401 se puede configurar para analizar, para obtener un nodo de vista virtual, datos estructurados de una plantilla de interfaz enviada por un servidor. El servidor genera los datos estructurados de la plantilla de interfaz analizando los datos de la plantilla de interfaz escritos en un lenguaje de marcado de protocolo general. Los datos de la plantilla de interfaz incluyen una etiqueta común para describir un widget.

45 El módulo de enlace 402 se puede configurar para enlazar datos de servicio al nodo de vista virtual.

El módulo del motor de diseño 403 se puede configurar para calcular la información de ubicación del widget en base al nodo de vista virtual al que están vinculados los datos del servicio.

50 El módulo de motor de representación 404 se puede configurar para representar una vista basada en la información de ubicación del widget. El lenguaje de marcado de protocolo general puede ser HTML, y los datos estructurados pueden ser datos JSON o datos XML.

55 La información almacenada en el nodo de vista virtual puede incluir información relacionada de la etiqueta común y/o el tipo de la aplicación cliente, y/o información sobre un evento de respuesta en la aplicación cliente.

El módulo del motor de análisis 401 se puede configurar adicionalmente para indicar al nodo de vista virtual que actualice la información relacionada almacenada después de que la aplicación cliente recibe datos estructurados de una plantilla de interfaz actualizada.

60 Además, se puede incluir un módulo de datos de servicio 405, y se configura para recibir los datos de servicio suministrados por el servidor y enviar los datos de servicio al módulo de motor de análisis 401, y solicitar los datos de servicio del servidor en base a una notificación del módulo de motor de análisis 401.

65 El módulo del motor de análisis 401 se puede configurar adicionalmente para enviar una notificación al módulo de datos de servicio 405 cuando no se identifican datos estructurados de una plantilla de interfaz que corresponda a los datos de servicio.

Que el módulo del motor de representación 404 representa una vista basada en la información de ubicación del widget que incluye: después de que el módulo del motor de diseño 403 calcula la información de ubicación de los widgets, obtener la información de ubicación de los widgets y representar los widgets en la vista.

5 La Figura 5 es una implementación de un dispositivo de presentación de datos de interfaz 500. El dispositivo 500 está ubicado en el lado del servidor e incluye: un módulo de motor de análisis 501, configurado para analizar, para obtener datos estructurados, datos de plantilla de interfaz escritos en un lenguaje de marcado de protocolo general, donde los datos de plantilla de interfaz incluyen una etiqueta común para describir un widget y un módulo de comunicaciones 502, configurado para suministrar los datos estructurados a una aplicación cliente, de manera que
10 la aplicación cliente presente el widget en una vista.

El lenguaje de marcado de protocolo general puede ser HTML, los datos estructurados pueden ser datos JSON y la etiqueta común se puede almacenar en un nodo DOM para los datos JSON.

15 Para un proceso de implementación de funciones y roles de cada unidad en el dispositivo, se puede hacer referencia a un proceso de implementación de una etapa correspondiente en el método anterior. Los detalles no se describen aquí por simplicidad.

20 La implementación del aparato puede corresponder a la implementación del método, y por lo tanto se describe brevemente por simplicidad. Para partes relacionadas, se puede hacer referencia a descripciones en la implementación del método. La implementación del aparato descrito es simplemente un ejemplo. Las unidades descritas como partes separadas pueden o no estar físicamente separadas, y las partes mostradas como unidades pueden o no ser unidades físicas, pueden ubicarse en una posición o pueden distribuirse en una pluralidad de unidades de red. Una parte o la totalidad de los módulos se pueden seleccionar en función de las necesidades
25 reales para lograr los objetivos de las soluciones de la presente solicitud. Una persona de habilidad ordinaria en la técnica puede comprender e implementar las implementaciones de la presente solicitud sin esfuerzos creativos.

REIVINDICACIONES

1. Un método para representar datos en una interfaz de aplicación, el método que comprende:
 - 5 analizar (S101), mediante una aplicación cliente, datos estructurados de una plantilla de interfaz, en donde la plantilla de interfaz es suministrada por un servidor a la aplicación cliente para obtener un nodo de modelo de objeto de documento, en donde los datos estructurados de la plantilla de interfaz son generados por el servidor analizando los datos de la plantilla de la interfaz escritos en un lenguaje de marcado de protocolo general, y los datos de la plantilla de la interfaz comprenden una etiqueta común para describir un widget, en donde el nodo del modelo del objeto del documento almacena información relacionada con la etiqueta común, y en donde el nodo del modelo del objeto del documento comprende información de ubicación del widget en la interfaz descrita mediante el uso de la etiqueta común, y en donde la plantilla de interfaz comprende un primer identificador de interfaz único para distinguir entre plantillas de interfaz;
 - 10 basado en los datos estructurados analizados, generar, mediante la aplicación cliente, un nodo de vista virtual como parte de una estructura de árbol de nodos de vista virtual para almacenar información para nodos de modelo de objeto de documento, que incluye información relacionada con la etiqueta común;
 - vincular (S 102), mediante la aplicación cliente, datos de servicio al nodo de vista virtual desde la estructura de árbol de los nodos de vista virtual, en donde la aplicación cliente busca una plantilla de interfaz que corresponda a los datos de servicio basados en un identificador de interfaz en los datos de servicio que corresponden al primer identificador único de interfaz;
 - 20 calcular (S103) información de ubicación del widget en función de la información almacenada en el nodo de vista virtual al que están vinculados los datos del servicio;
 - representar (S104), mediante la aplicación cliente, el widget en una vista de una interfaz de la aplicación cliente en una ubicación correspondiente de la vista basada en la información de ubicación del widget, en donde los datos de servicio se insertan en el widget en la vista (S104); y
 - 25 activar mediante la aplicación cliente un procedimiento de servicio para presentar los datos del servicio; después de obtener los datos de servicio del servidor, buscar, mediante la aplicación cliente, una plantilla de interfaz que corresponda a los datos de servicio basados en un identificador de interfaz en los datos de servicio; y
 - 30 si no se identifica una plantilla de interfaz, solicitar, mediante la aplicación cliente, datos estructurados actualizados de la nueva plantilla de interfaz que corresponde al identificador de interfaz del servidor.
- 35 2. El método de acuerdo con la reivindicación 1, en donde el lenguaje de marcado de protocolo general es HTML, y los datos estructurados son datos JSON o datos XML.
- 40 3. El método de acuerdo con cualquiera de las reivindicaciones 1 a 2, en donde el nodo de vista virtual almacena información relacionada con la etiqueta común, y/o el tipo de la aplicación cliente, y/o un evento de respuesta en la aplicación cliente.
- 45 4. El método de acuerdo con cualquiera de las reivindicaciones 1 a 3, en donde el método comprende, además: después de recibir datos estructurados actualizados de una plantilla de interfaz actualizada, actualizar, mediante la aplicación cliente, la información relacionada almacenada en el nodo de vista virtual.
- 50 5. El método de acuerdo con la reivindicación 1, que comprende, además:
 - después de recibir los datos de estructura actualizados del servidor, analizar mediante la aplicación cliente, los datos de estructura actualizados para obtener un nodo de vista virtual actualizado;
 - 55 actualizar la información relacionada almacenada en el nodo de vista virtual, en donde el nodo de vista virtual está vinculado a los datos de servicio en base a la información actualizada; y
 - pasar el nodo de vista virtual a un motor de representación para volver a representar el nodo de vista virtual.
- 60 6. El método de acuerdo con cualquiera de las reivindicaciones 1 a 5, en donde representar el widget en el nodo de vista virtual basado en la información de ubicación del widget comprende: después de calcular la información de ubicación de los widgets, obtener la información de ubicación de los widgets y mostrar los widgets en la vista.
- 65 7. El método de acuerdo con cualquiera de las reivindicaciones 1 a 6, que comprende, además: suministrar los datos estructurados a la aplicación cliente, de modo que la aplicación cliente presente el widget en el nodo de vista virtual.
8. El método de acuerdo con cualquiera de las reivindicaciones 1 a 7, en donde los datos estructurados son datos JSON y la etiqueta común se almacena en el nodo del modelo de objeto de documento para los datos JSON.

9. El método de acuerdo con cualquiera de las reivindicaciones 1 a 8, en donde los datos estructurados comprenden un primer árbol de sintaxis con un formato más simple que un segundo árbol de sintaxis de la plantilla de interfaz.
- 5 10. El método de acuerdo con cualquiera de las reivindicaciones 1 a 9, en donde vincular los datos de servicio al nodo de vista virtual comprende insertar los datos de servicio en el nodo de vista virtual mediante el uso de una función de entrada.
- 10 11. El método de acuerdo con cualquiera de las reivindicaciones 1 a 10, que comprende, además:
después de activar un procedimiento de servicio, enviar mediante la aplicación cliente al servidor una solicitud cuando los datos de servicio deban ser presentados en la interfaz de la aplicación cliente por la nueva plantilla de interfaz;
15 buscar, mediante la aplicación cliente, la nueva plantilla de interfaz que corresponda a los datos del servicio en base a un identificador de interfaz en los datos del servicio; y
si no se identifica una plantilla de interfaz en base a la búsqueda, solicitar datos de estructura de la plantilla de interfaz correspondiente a la interfaz identificada desde el servidor.
- 20 12. Un dispositivo de presentación de datos de interfaz, el dispositivo implementado por software, hardware o una combinación de software y hardware, el dispositivo configurado para realizar el método de cualquiera de las reivindicaciones 1 a 11.

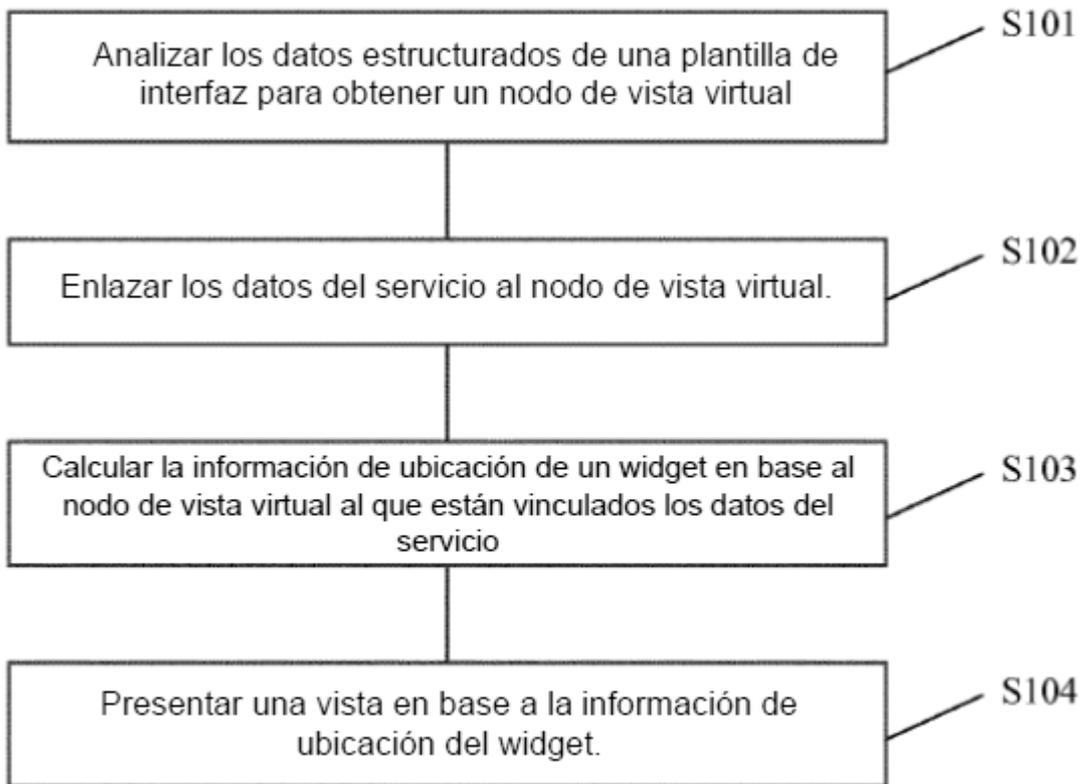


FIGURA 1

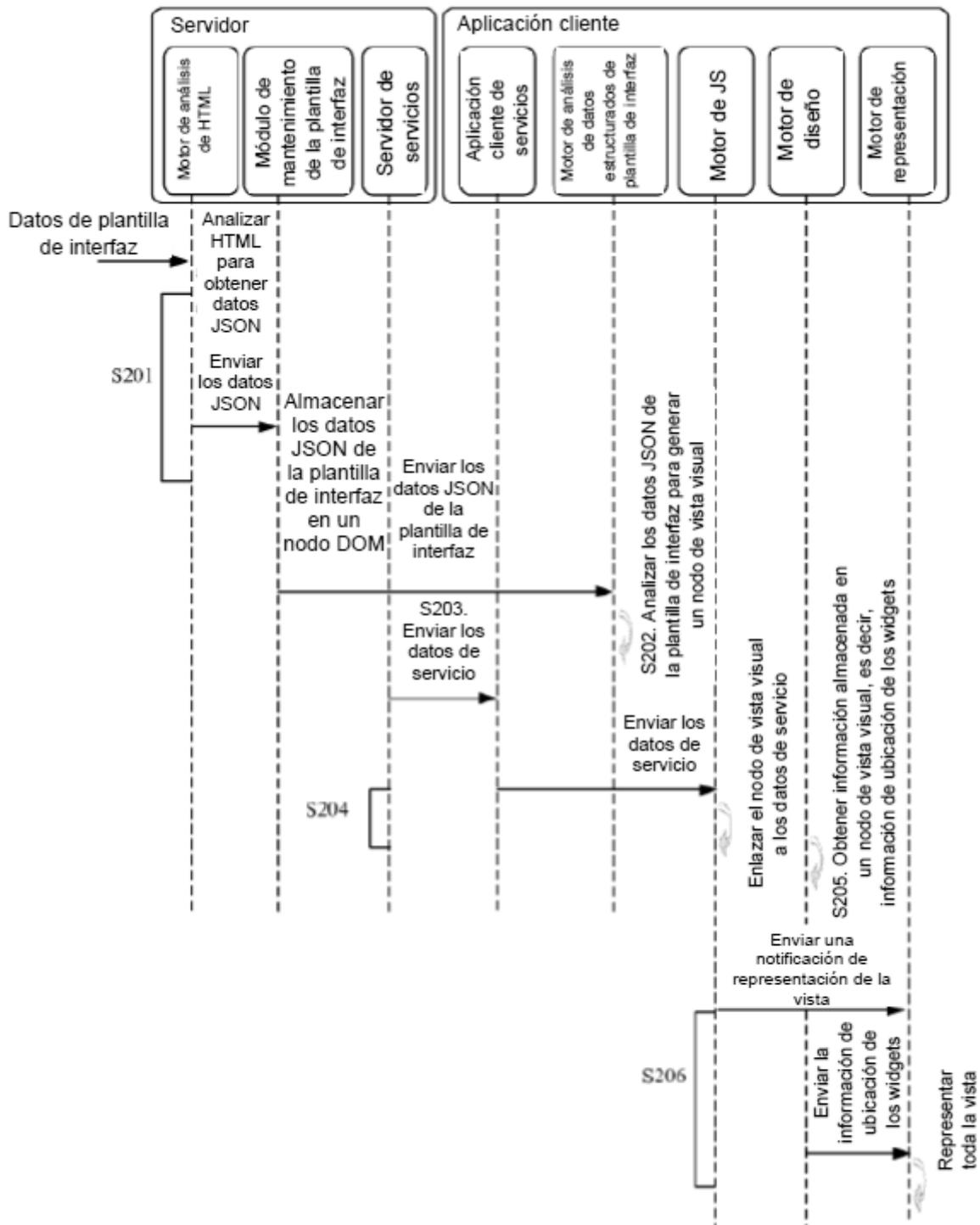


FIGURA 2

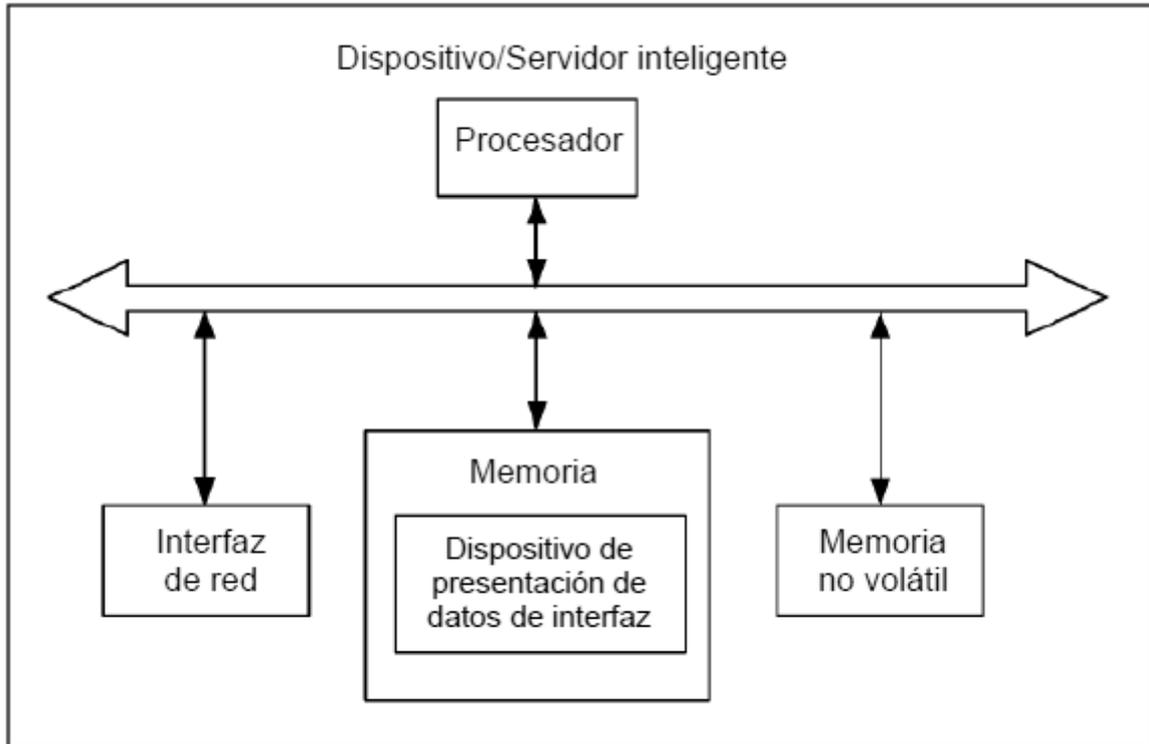


FIGURA 3

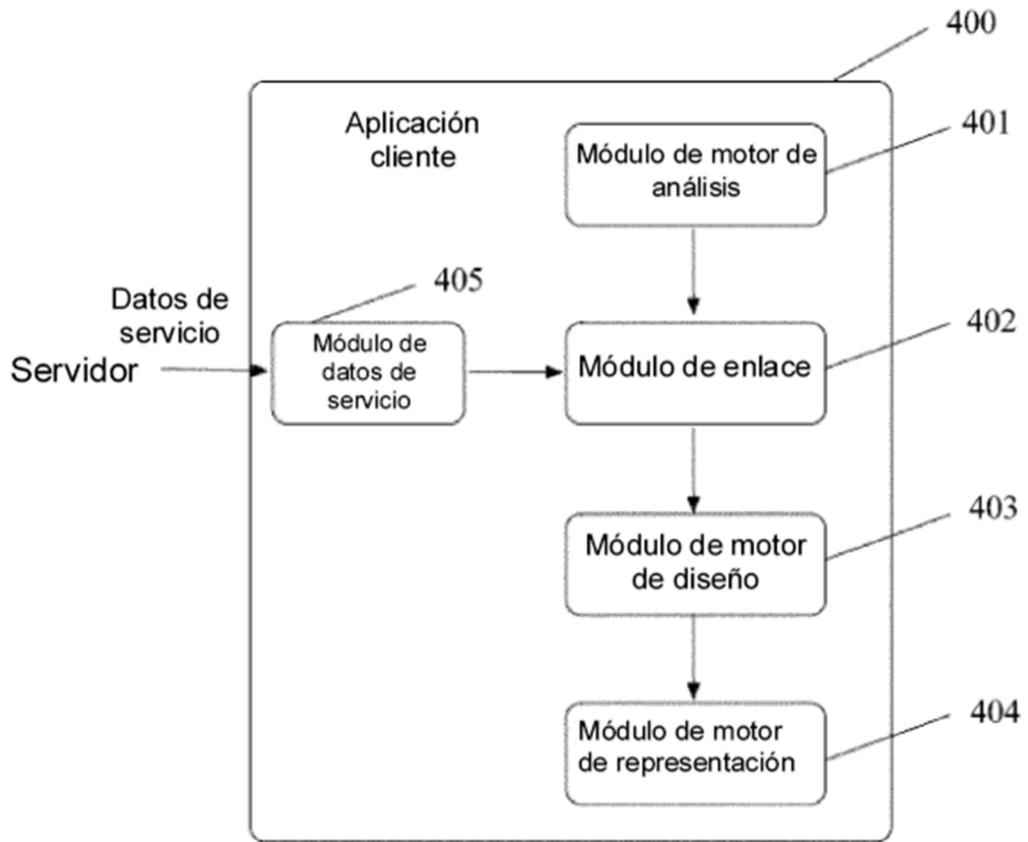


FIGURA 4

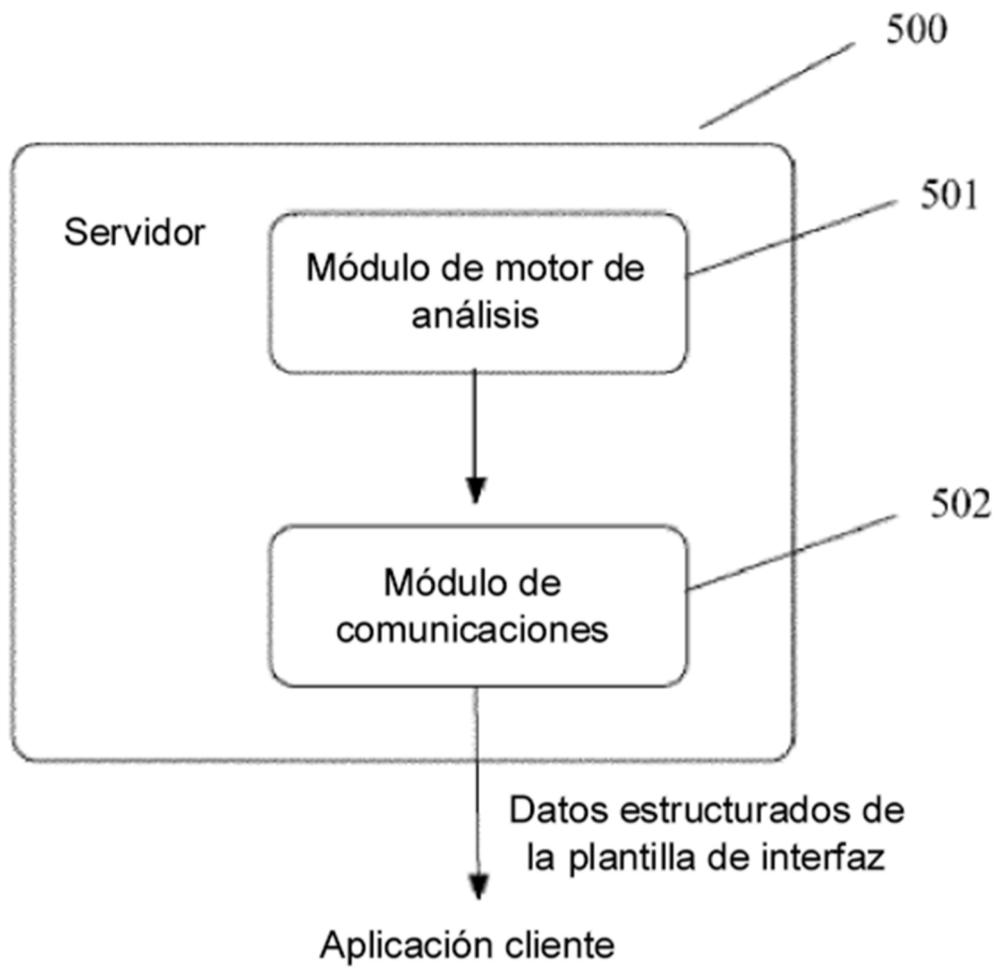


FIGURA 5