



(19)



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 297 914**

(51) Int. Cl.:
G06F 11/14 (2006.01)

(12)

TRADUCCIÓN DE PATENTE EUROPEA

T3

(86) Número de solicitud europea: **99307546 .4**

(86) Fecha de presentación : **23.09.1999**

(87) Número de publicación de la solicitud: **1014266**

(87) Fecha de publicación de la solicitud: **28.06.2000**

(54) Título: **Método, aparato y dispositivo de almacenamiento de programa para un cliente y un servidor de sincronización adaptativo y de transformación.**

(30) Prioridad: **23.10.1998 US 178100**

(45) Fecha de publicación de la mención BOPI:
01.05.2008

(45) Fecha de la publicación del folleto de la patente:
01.05.2008

(73) Titular/es:
**International Business Machines Corporation
New Orchard Road
Armonk, New York 10504, US**

(72) Inventor/es: **Huang, Yun Wu y
Yu, Philip Shi-Lung**

(74) Agente: **Elzaburu Márquez, Alberto**

ES 2 297 914 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, aparato y dispositivo de almacenamiento de programa para un cliente y un servidor de sincronización adaptativo y de transformación.

El presente invento se refiere en general a un proceso de sincronización y/o transformación para actualización entre datos en un cliente y réplicas de los datos en sistemas de gestión de información.

Con el rápido avance de las tecnologías de semiconductores, almacenamiento en memoria y presentación visual, los dispositivos portátiles o móviles, que solamente podrían conectarse ocasionalmente a la red, han llegado a ser cada vez más populares. Las modalidades en que se han usado tales dispositivos móviles, tal como el PALM PILOT™ de 3com han llegado a ser también muy versátiles. Por ejemplo, un usuario de PALM PILOT™ podría tener una aplicación de correo electrónico que sincronizarse con un sistema de correo electrónico Lotus NOTES™ en un escritorio de oficina. Podría existir también una aplicación de cotización de acciones en el PALM PILOT™ que consiga actualizarse mediante la sincronización con un sitio bien conocido de cotización de acciones situado en Internet. Podría existir también una aplicación de cotización de acciones en el PALM PILOT™ que tenga que sincronizarse con software financiero de ordenador personal (en adelante PC) local. Por tanto, es bastante posible que diversas aplicaciones dentro de un dispositivo necesiten sincronizar con aplicaciones diferentes en sistemas diversos de gestión de información (tales como NOTE™ de Lotus, EXCHANGE™ de Microsoft, sistemas de ficheros, bases de datos relacionales, bases de datos orientadas a objetos) que se ejecuten en ordenadores conectados a red. También es posible que una aplicación portátil (tal como un correo electrónico) tenga diferentes versiones para diferentes dispositivos portátiles (por ejemplo, PALM PILOT™, ZAURUS™ de Sharp, PSION™ de Psion PLC y otros varios dispositivos de bolsillo que ejecuten WINDOWS CE™ de Microsoft).

Como los dispositivos portátiles como PALM PILOT™ sólo se usan típicamente en ocasiones conectados a la red con la conexión que dura solamente un período corto de tiempo, es crítico que los procesos de sincronización entre las aplicaciones en los portátiles y sus contrapartidas en diversos ordenadores conectados a la red sean muy eficaces (de tal manera que el proceso de sincronización se pueda completar de forma satisfactoria mientras los portátiles están conectados a la red). Con los dispositivos portátiles que típicamente ofrecen menos potencia de CPU y menos memoria para computación, los procesos de sincronización usualmente no se realizan dentro de los dispositivos portátiles.

En este caso, a una unidad central de ordenador que centraliza las réplicas de datos de dispositivos portátiles se le denomina unidad central de réplica, o fuente de réplica. Como los dispositivos portátiles en general tienen menos potencia informática y memoria que una unidad central de réplica, los dispositivos portátiles típicamente realizan sincronización dentro de sus unidades centrales de réplica, que son PC de sobremesa. Típicamente, la sincronización se inicia en primer lugar apretando un pulsador del dispositivo portátil (por ejemplo, el pulsador HotSync en un PALM PILOT™). Inmediatamente después de apretar el pulsador, un software de gestor de sincronización de la unidad central de réplica (por ejemplo, el gestor HotSync de PILOT™ en el ordenador personal de mesa (en adelante PC) se hace cargo del proceso de sincronización. En el caso de un PALM PILOT™, el gestor de HotSync realiza la sincronización para una aplicación de PALM PILOT™ mediante la ejecución de un programa de aplicación específica (denominado Conducto en el PALM PILOT™) que sincroniza esta aplicación entre el PALM PILOT™ y un PC de mesa por medio de lógica de aplicación específica. Si es necesario sincronizar múltiples aplicaciones, el gestor de HotSync ejecuta cada uno de sus conductos correspondientes en orden secuencial. Basándose en este concepto, la sincronización se lleva a cabo siempre entre un PC y un dispositivo portátil. No se dispone de una sincronización simultánea entre múltiples dispositivos portátiles y un PC.

En general, la realización de los procesos de sincronización directamente en la unidad central de réplicas es muy limitativa. La razón estriba en que podrían existir muchas aplicaciones (tales como correo electrónico, calendario, agenda) de los que cada una puede ser compartida por muchos dispositivos portátiles diferentes (por ejemplo, PALM PILOT™, ZAURUS™, PSION™). Cada aplicación podría requerir una lógica de sincronización diferente para cada dispositivo diferente. Para procesar adecuadamente la sincronización para todas las aplicaciones y todos los dispositivos, una unidad central de réplicas podría necesitar instalar y gestionar muchas versiones diferentes de procesos de sincronización con el fin de manejar diferentes dispositivos portátiles. Por ello, la realización de procesos de sincronización en las unidades centrales de réplicas crea una carga pesada en la gestión de los diferentes dispositivos portátiles y de sus procesos de sincronización. Para muchas unidades centrales de réplicas, su tarea de una misión más crítica es no sincronizar con diversos portátiles, sino funcionar como un servidor de información tal como un servidor de base de datos, un servidor de grupo, o un dispositivo de ordenador personal de mesa. La ejecución de sincronización sobre ellos los detrae de sus tareas de misión crítica.

Una solución más eficaz a la sincronización de gestión entre múltiples dispositivos portátiles y unidades centrales de réplicas es el despliegue de proxies (filtros) de sincronización (o servidores de sincronización (en adelante sync) o servidores proxy de sincronización, o simplemente proxies sync) que mantiene conexiones persistentes de red, para realizar la sincronización en nombre de los portátiles y de las unidades centrales de réplicas (los dispositivos que centralizan las réplicas mantienen también una conexión persistente de red).

Para sincronizar, un portátil primero establece una conexión de red. Una vez conectado, en lugar de hacer una conexión directa a la unidad central de réplicas que es el objetivo para sincronización, hace una solicitud de sincronización a un proxy sync. Tras recibir dicha solicitud, el proxy sync establece una conexión a la unidad central de réplicas

objetivo y comienza el proceso de sincronización mediante la emisión de solicitudes de entrada/salida tanto al portátil como a la unidad central de réplicas. Un diseño claro del proxy de sincronización es para el proxy (filtro) mantener todos los programas de sincronización específicos de dispositivo y específicos de aplicación que puedan manejar todos los tipos de dispositivo portátil y todos los tipos de sistema de gestión de información centralizados por las unidades
 5 centrales de réplicas. Esto quiere decir que cada aplicación debe tener un único programa de sincronización desarrollado específicamente para cualquier combinación entre todos los tipos de dispositivo portátil y todos los sistemas de gestión de información que centralicen las réplicas de datos de portátiles. Por ejemplo, un proxy sync podría mantener cuatro programas de sincronización diferentes para una aplicación de correo electrónico con el fin de soportar ambos dispositivos portátiles basados en PALM PILOT™ y en WINDOWS CE™ y ambos sistemas de correo electrónico
 10 Microsoft EXCHAANGE™ y Lotus NOTES™ para PC. El primer programa de sincronización es para sincronizar entre PALM PILOT™ y EXCHANGE™, el segundo entre PALM™ y Lotus NOTES™, el tercero entre WINDOWS CE™ y EXCHANGE™, y el cuarto entre WINDOWS CE™ y Lotus NOTES™.

Otro aspecto de este diseño claro del proxy sync es permitir que la interfaz de entrada/salida de sincronización
 15 entre el proxy de sincronización y un sistema específico de gestión de información que centraliza réplicas de datos de portátil sea la misma para diferentes dispositivos portátiles. El proxy sync es el que maneja las diferencias entre dispositivos portátiles, no cada unidad central de réplicas. Los procesos reales de sincronización se llevan a cabo mediante los proxies de sincronización. El papel de los portátiles y de las unidades centrales de réplica durante la sincronización es simplemente responder a las llamadas de entrada/salida de sincronización realizadas por el proxy de sincroniza-
 20 ción. Estas llamadas están típicamente en la forma de información de adición, de recuperación, de eliminación, y de actualización. Por tanto, se ha reducido mucho la carga informática y la complejidad de la lógica de sincronización tanto para portátiles como para unidades centrales de réplica.

Una propuesta importante de esta solución clara de proxy de sincronización es que, con el fin de sincronizar ade-
 25 cuadamente entre cualesquiera aplicaciones para todos los tipos de dispositivo y cualesquiera sistemas de gestión de información para todas las unidades centrales de réplica, un proxy de sincronización debe ser capaz de invocar, durante una tarea de sincronización, una lógica de sincronización específica basada en la aplicación, dispositivo, y sistema de gestión de información requeridos como objetivos. La solución clara actual de proxy de sincronización cumple estos objetivos, en primer lugar mediante la restricción de su soporte a un pequeño número de aplicaciones, dispo-
 30 sitivos y sistemas de gestión de información para reducir el número total de variaciones de lógica de sincronización requerido para una sincronización adecuada; y mediante la pre-configuración de lógica de sincronización como unos códigos dinámicamente enlazables (durante el tiempo de ejecución de la sincronización) guardados en su memoria de disco. Una importante cuestión de dicha solución es la dificultad en la gestión de todas las diversas lógicas de sin-
 35 cronización. Si un organismo soporta más aplicaciones, dispositivos, y sistemas de gestión de información, el número de variaciones de lógica de sincronización aumenta drásticamente. Por tanto, cada proxy de sincronización debe ser capaz de enlazar la lógica de sincronización correcta. Además, en un organismo con múltiples proxies de sincroni-
 40 zación desplegados, cualquier cambio y actualización a una lógica de sincronización crea una difícil tarea de gestión de software de la propagación de estos cambios y actualizaciones a los ficheros correctos de todos los proxies de sincronización. El perfeccionamiento de la gestión de la lógica de sincronización es un importante objeto del presente invento.

Otra propuesta de la actual solución clara del proxy de sincronización es que las diversas lógicas de sincronización se pre-codifican por los realizadores de los proxies de sincronización, algunos de los cuales proporcionan también una
 45 plataforma de desarrollo para usuarios con el fin de aumentar o modificar la lógica de sincronización. Los realizadores de proxies de sincronización no crean en general las aplicaciones, los dispositivos, o los sistemas de gestión de información. Por tanto, no poseen el conocimiento experimentado de estos software y hardware y deben limitar el alcance de su apoyo. La escalabilidad de los proxies de sincronización actuales en cuanto a soportar más aplicaciones de portátiles, más dispositivos portátiles, y más sistemas de gestión de información es muy limitada. El presente invento atiende a esta limitación.

Además, el proxy claro de sincronización es un sistema autónomo cuya escalabilidad en relación con el proce-
 50 samiento de solicitudes simultáneas de sincronización está limitada por la potencia informática de la máquina en la que se instala el proxy de sincronización. No se dispone de colaboración entre proxies de sincronización en cuanto al equilibrado de carga en la solución clara de proxy de sincronización. Esta carencia de colaboración entre proxies de
 55 sincronización podría crear potencialmente una carga desequilibrada entre proxies que causa que algunas solicitudes de sincronización se retrasen o no se atiendan. El presente invento atiende esta necesidad.

El documento WO 97/04389 describe un método y un aparato para sincronizar transacciones en una red desco-
 60 nectable. Cada transacción incluye operaciones que se realizaron sobre una réplica de base de datos en un ordenador mientras este ordenador estaba desconectado de otro ordenador y por tanto de otra réplica de ordenador.

De acuerdo con las necesidades anteriormente mencionadas, el presente invento está dirigido a un método, aparato y dispositivo de almacenamiento de programa perfeccionados para un proxy de sincronización (o proxy de sync o servidor de sincronización), que sirve como un intermediario y una plataforma de cálculo en la realización de
 65 sincronización de actualización entre clientes, cuyos ejemplos incluyen, pero sin carácter limitativo: una aplicación para Internet; un dispositivo portátil o de bolsillo; un teléfono inteligente o un buscapersonas inteligente; y sistemas de gestión de información que centralizan las réplicas de datos de estos servicios. Los clientes podrían desconectarse de la red en algo o en la mayor parte del tiempo, mientras que los sistemas de gestión de información permanecen

conectados casi todo el tiempo. Los datos contenidos en los dispositivos de cliente y/o en sus réplicas de los sistemas de gestión de información se podrían actualizar con independencia entre sí.

Preferiblemente, la lógica de aceptador de sincronización comprende además medios para recibir del cliente la solicitud de sincronización incluyendo información para identificar la unidad central de réplicas y la lógica de sincronización distante; la lógica de sincronización comprende medios para una o más de las siguientes operaciones: resolver un conflicto, especificar una dirección de desbordamiento, y tomar ciertas medidas para satisfacer una necesidad específica de la aplicación; la solicitud de sincronización incluye una dirección de un servicio de directorio que guarda información para identificar la unidad central de réplica y la lógica de sincronización; y la lógica de manipulador de sincronización comprende medios para recuperar la información, en respuesta a la solicitud de sincronización

Preferiblemente, la solicitud de sincronización incluye información para identificar un código de transformación específico de dispositivo para convertir los datos entre el cliente y la unidad central distante durante la sincronización; y la lógica de manipulador de sincronización comprende medios para recuperar y convertir los datos entre el cliente y la unidad central distante basándose en el código de transformación específico de dispositivo; y en el que el código de transformación específico de dispositivo se puede situar en un punto distante en cualquier lugar de la red; y en el que la solicitud de sincronización incluye una dirección de un servicio de directorio que guarda información para identificar el código de transformación específico de dispositivo para transformar los datos entre el cliente y la unidad central distante; y la lógica de manipulador de sincronización comprende medios para recuperar el código de transformación distante basándose en la información y convertir los datos entre el cliente y la unidad central distante basándose en el código de transformación específica de dispositivo, y en el que el código de transformación comprende medios para una o más de las siguientes operaciones: filtrar; transformar; y cambiar los datos de tal manera que se puedan usar y procesar en diferentes dispositivos informáticos que tengan una o más de diferentes potencias de CPU, capacidades de memoria, y construcciones físicas.

Preferiblemente, el servidor comprende además un gestor de carga, acoplado a la lógica de aceptador, para monitorizar una carga informática de servidor y una capacidad informática de servidor y compartir la carga informática y la capacidad informática con servidores de sincronización que colaboren; y cuyo gestor de carga comprende una lógica para desviar la solicitud de sincronización a un servidor menos cargado, cuando la carga de computación de servidor excede de un valor umbral. Adicionalmente, la red incluye una o más de una Intranet e Internet en donde el código de transformación comprende además medios para filtrar una imagen o transformar una resolución de la imagen incluida en una página web enviada desde la unidad central distante al dispositivo portátil de tal manera que se pueda mostrar la página en una pantalla en el dispositivo portátil dentro de las restricciones de memoria del dispositivo portátil.

Preferiblemente, el servidor comprende además medios de gestor de caché (memoria asociada), acoplados al manipulador de sincronización, para recuperar y guardar la lógica de sincronización en una memoria caché, en el que dichos medios de gestor de caché comprenden además medios para preconfigurar y guardar la lógica de sincronización en la memoria caché.

Preferiblemente, la información de identificación comprende un localizador uniforme de recursos (en adelante URL). Además, la red es la www y el servidor de sincronización es un servidor proxy (filtro) y en el que el cliente se selecciona de un grupo que consiste en un dispositivo portátil; un teléfono inteligente, o un buscapersonas inteligente.

De acuerdo con un segundo aspecto, el invento provee un servidor de adaptación para transformar los datos entre un cliente y una réplica de unidad central distante, que guarda una réplica de datos en el cliente, según se reivindica en la reivindicación 2.

Preferiblemente, la lógica de aceptador comprende medios para recibir del cliente la solicitud que incluye información para identificar la unidad central de réplica y el código de transformación distante; en donde la solicitud incluye una dirección de un servicio de directorio que guarda información para identificar la unidad central de réplica y el código de transformación; y la lógica de manipulador comprende además medios para recuperar la información, en respuesta a la solicitud; en donde el código de transformación comprende medios para una o más de las siguientes operaciones: filtración; transformación; y cambio de datos para que se pueda usar y procesar en diferentes dispositivos informáticos que tengan una o más potencias de CPU diferentes; capacidades de memoria diferentes, y construcciones físicas diferentes; en donde el código de transformación comprende además medios para filtrar una imagen o transformar una resolución de la imagen incluida en una página web enviada desde la unidad central distante al dispositivo portátil, de tal manera que se pueda mostrar la página en una pantalla en el dispositivo portátil dentro de las restricciones de memoria del dispositivo portátil; en donde la solicitud incluye información para identificar una lógica que es una aplicación específica a un tipo de datos relacionados con el cliente y la unidad principal distante, en respuesta a una solicitud; y la lógica de manipulador que comprende medios para recuperar la lógica, conectar a la unidad central distante basándose en la información, y sincronizar los datos entre el cliente y la unidad principal distante.

Además, preferiblemente a lo antes expuesto, en donde la lógica comprende medios para una o más de las siguientes operaciones: resolver un conflicto; especificar una dirección de desbordamiento; y tomar determinadas medidas para satisfacer una necesidad específica de la aplicación; en donde la lógica se puede situar distante en cualquier lugar de la red, en donde la solicitud incluye una dirección de un servicio de directorio que guarda información para identificar la lógica; y la lógica de manipulador comprende medios para recuperar la lógica distante basándose en la información y sincronizar los datos entre el cliente y la unidad central distante basándose en la lógica.

Preferiblemente, el servidor comprende además un gestor de carga, acoplado a la lógica de aceptador, para monitorizar una carga informática de servidor y una capacidad informática de servidor y compartir la carga informática y la capacidad informática con servidores colaboradores; y cuyo gestor de carga comprende una lógica para desviar la solicitud a un servidor menos cargado, cuando la carga informática del servidor excede de un valor umbral.

Preferiblemente, el servidor comprende además unos medios de gestor de memoria caché, acoplados al manipulador, para recuperar y guardar la lógica en una memoria caché, en donde dichos medios de gestor de memoria caché comprenden además unos medios para preconfigurar y guardar la lógica en la memoria caché.

Preferiblemente, la información de identificación comprende un URL. Además, la red es la www y el servidor es un servidor proxy y en donde el cliente se selecciona de un grupo que consiste en un dispositivo portátil; un teléfono inteligente, o un buscapersonas inteligente.

De acuerdo con un tercer aspecto, el invento provee un dispositivo de cliente destinado a su conexión a un servidor de sincronización en donde se realiza la sincronización de los datos entre el cliente y una réplica de unidad central distante que guarda una réplica de los datos sobre el cliente, como se ha reivindicado en la reivindicación 3.

Preferiblemente, el cliente es un dispositivo portátil desconectable que se ha seleccionado de un grupo consistente en un ordenador portátil; un teléfono inteligente, o un buscapersonas inteligente, que comprende medios para establecer una conexión a la red, en donde la gestión de datos comprende una o más de las operaciones siguientes: extraer información de un elemento de datos; introducir información en un elemento de datos; actualizar un elemento de datos; y eliminar un elemento de datos, comprendiendo además medios para comunicar versión y actualizar información histórica entre el cliente y la unidad central de réplica distante, en donde la solicitud de sincronización comprende además información para identificar la unidad central de réplica y la lógica de sincronización, en donde la solicitud de sincronización incluye una dirección de un servicio de directorio que guarda información para identificar la unidad central de réplica y la lógica de sincronización, en donde la solicitud de sincronización incluye información para identificar un código de transformación específico de dispositivo para transformar datos entre el cliente y la unidad central distante, en donde el código de transformación específico de dispositivo se puede ubicar en un punto distante en cualquier lugar de la red, en donde la solicitud de sincronización incluye una dirección de un servicio de directorio que guarda información para identificar uno o más de la lógica de sincronización y del código de transformación específico de dispositivo para transformar los datos entre el cliente y la unidad central distante, en donde el cliente es un dispositivo portátil desconectable que está acoplado a una o más de una Intranet o de Internet en donde los datos incluyen una imagen incluida en una página web enviada desde la unidad central distante al dispositivo portátil transformados de tal manera que la página se pueda mostrar en una pantalla del dispositivo portátil dentro de las restricciones de memoria del dispositivo portátil.

Preferiblemente, la información de identificación comprende un URL.

Preferiblemente, la solicitud incluye una o más direcciones de red del servidor de sincronización guardadas en la memoria en las que una dirección de servidor de sincronización se preconfigura como una dirección por defecto de servidor de sincronización.

Preferiblemente, la solicitud incluye información de identificación sobre el cliente o un usuario seleccionado de un grupo que consiste en uno o más de los siguientes: un número de identificación (en adelante ID) de usuario; una contraseña cifrada; información secreta compartida para autenticación y autorización; un tipo de dispositivo; o un tipo de sistema en relación de asociación con el cliente.

Preferiblemente, el iniciador de sincronización está destinado a determinar una lista de aplicaciones que necesitan sincronizarse, cuyas aplicaciones se seleccionan de un grupo que consiste en una o más de las siguientes: una aplicación de agenda; una aplicación de bloc memorándum; una aplicación de calendario; y una aplicación de correo electrónico.

De acuerdo con un cuarto aspecto, el invento provee un dispositivo de cliente destinado a la conexión a un servidor de adaptación en donde la transformación de datos se realiza entre el cliente y una réplica de unidad central distante que guarda una réplica de los datos en el cliente, según se ha reivindicado en la reivindicación 6.

Preferiblemente, el cliente es un dispositivo portátil desconectable que se ha seleccionado de un grupo que consiste en: un ordenador portátil; un teléfono inteligente, o un buscapersonas inteligente, comprendiendo medios para establecer una conexión a la red, en donde la solicitud comprende además información para identificar la unidad central de réplica y el código de transformación, en donde la solicitud incluye una dirección de un servicio de directorio que guarda información para identificar la unidad central de réplica y el código de transformación, en donde la solicitud incluye información para identificar una lógica que es específica de aplicación a un tipo de datos en relación de asociación con el cliente y la unidad central distante, en donde la lógica se puede ubicar en cualquier lugar en una red distante al servidor y a la unidad central distante, en donde la solicitud incluye una dirección de un servicio de directorio que guarda información para identificar uno o más de la lógica y del código de transformación específica de dispositivo. Preferiblemente, el cliente es un dispositivo portátil desconectablemente acoplado a una o más de una intranet o Internet en donde los datos incluyen una imagen incluida en una página web enviada desde la unidad central distante al dispositivo portátil transformado de tal manera que la página se puede mostrar en una pantalla del dispositivo portátil dentro de las restricciones de memoria del dispositivo portátil.

Preferiblemente, la información de identificación comprende un URL.

Preferiblemente, la solicitud incluye una o más direcciones de red del servidor guardadas en la memoria, en donde una dirección de servidor está preconfigurada como una dirección por defecto de servidor.

El presente invento provee un servidor de sincronización para sincronizar datos entre un cliente y una réplica de unidad central distante, que guarda una réplica de datos en el cliente, como se ha reivindicado en la reivindicación 1.

El invento provee también un dispositivo de cliente, según se ha reivindicado en la reivindicación 2, destinado a su conexión al servidor de sincronización.

El invento provee también, en un servidor de sincronización, un método como el reivindicado en la reivindicación 4, y un programa de ordenador correspondiente como se ha reivindicado en la reivindicación 5.

De acuerdo con una realización preferida del presente invento, se ha provisto un método para que un cliente solicite servicios de sincronización de un proxy de sincronización. Por ejemplo, un dispositivo portátil emite una solicitud de sincronización a este proxy. Una solicitud de sincronización podría implicar una o más aplicaciones a sincronizar. Una vez que se ha aceptado la solicitud de sincronización, el proxy de sincronización procesa la sincronización para todas las aplicaciones solicitadas con el fin de sincronizarse para este dispositivo portátil de un modo secuencial. Para cada aplicación a sincronizar, el portátil podría proveer un identificador de sincronización que podría incluir el nombre de la aplicación a sincronizar, el número de identificación de la unidad central de réplica para esta aplicación, el ID del programa con lógica de sincronización específica de aplicación para esta aplicación (lógica sync), y el ID del programa con un método de transformación de datos específico de dispositivo para este dispositivo portátil (código de transformación). Los ID preferiblemente incluyen un identificador único (o un nombre único) y una dirección de red a partir de la que se puede recuperar esta información.

Alternativamente, durante el proceso de sincronización de una aplicación, en lugar de que el dispositivo portátil envíe directamente la información al proxy de sincronización, el dispositivo portátil podría enviar un identificador de sincronización que podría incluir solamente alguna información de identificación del usuario, el dispositivo portátil, y la aplicación a sincronizar, así como la dirección de un servicio de directorio en el que se guarden el nombre de la aplicación, y los tres fragmentos de información anteriormente mencionadas (es decir, el ID de la unidad central de réplica para esta aplicación, el ID de la lógica de sincronización para esta aplicación, y el ID del código de transformación para este dispositivo portátil).

Un ejemplo de un servidor de sincronización de adaptación para sincronizar datos entre un cliente y una réplica de unidad central distante, que guarda una réplica de datos en el cliente, y que tiene propuestas de una realización preferida del presente invento, incluye: lógica de aceptador de sincronización para identificar una unidad central de réplica y una lógica de sincronización que es específica de aplicación a un tipo de datos en relación de asociación con el cliente y la unidad central distante; en donde la lógica de sincronización se puede ubicar en cualquier lugar en una red distante del servidor de sincronización y de la unidad central adistante, en respuesta a una solicitud de sincronización; y lógica de manipulador de sincronización, acoplada a la lógica de aceptador de sincronización, para recuperar de la red la lógica de sincronización distante, conectar a la lógica de sincronización y sincronizar los datos entre el cliente y la unidad central distante usando lógica de sincronización recuperada. La solicitud de sincronización podría incluir también información para identificar un código de transformación específico de dispositivo para convertir datos entre el cliente y la unidad central distante durante la sincronización; en donde la lógica de manipulador de sincronización incluye medios para recuperar el código de transformación basándose en la información y convertir los datos entre el cliente y la unidad central distante basándose en el código de transformación específico de dispositivo. En estas condiciones, el código de transformación específico de dispositivo se puede ubicar localmente o distante en cualquier lugar de la red.

Otro ejemplo de un servidor de sincronización de adaptación para transformar datos entre un cliente, una réplica de unidad central distante que guarda una réplica de datos en el cliente, que tiene prestaciones de una realización preferida del presente invento, incluye: lógica de aceptador de sincronización para identificar la unidad central de réplica distante y un código de transformación específico de dispositivo para transformar los datos en la unidad central distante a los de un tipo de dispositivo en relación de asociación con el cliente, en donde el código de transformación se puede ubicar en cualquier lugar de una red distante del servidor de sincronización y de la unidad central distante, en respuesta a una solicitud; y lógica de manipulador de sincronización, acoplada a la lógica de aceptador de sincronización, para recuperar el código de transformación distante y transformar los datos entre el cliente y la unidad central distante basándose en el código de transformación específico de dispositivo. La solicitud puede incluir una solicitud de sincronización para identificar una lógica de sincronización, que es de específica aplicación a un tipo de datos en relación de asociación con el cliente y la unidad central distante, en respuesta a una solicitud de sincronización; y medios para recuperar la lógica de sincronización, conectar a la unidad central distante y sincronizar los datos entre el cliente y la unidad central distante durante la transformación.

De acuerdo con una realización preferida del presente invento, se ha provisto un método para que el proxy de sincronización reciba, acepte, y procese las solicitudes de sincronización realizadas desde los dispositivos portátiles. En la realización preferida, un proxy de sincronización -tras recibir una solicitud de sincronización de un dispositivo portátil- realiza una sincronización y/o una transformación para una o más aplicaciones portátiles. Para cada aplicación

portátil, el proxy podría recibir el nombre antes mencionado de la aplicación, el ID de unidad central de réplica, el ID de lógica de sincronización, y el ID del código de transformación bien directamente desde el dispositivo portátil solicitante o indirectamente de un servicio de directorio especificado por el dispositivo portátil solicitante. Para sincronizar una aplicación, el proxy de sincronización podría recuperar la lógica de sincronización en relación de asociación con esta aplicación de la dirección de red especificada en el ID de lógica de sincronización para esta aplicación enviado por el dispositivo portátil solicitante. Entonces, el proxy de sincronización establece una conexión de red con la unidad central de réplica en relación de asociación con esta aplicación, basándose en su ID de unidad central de réplica recibido del dispositivo portátil solicitante.

De acuerdo con una realización preferida del presente invento, para procesar sincronización para esta aplicación, el proxy de sincronización ejecuta la lógica recuperada de sincronización para esta aplicación. Durante la ejecución de la lógica de sincronización para esta aplicación, si se requiere transformación de datos, el proxy de sincronización hace una conexión a la dirección de red del código de transformación en relación de asociación con esta aplicación. La dirección de red del código de transformación es parte del ID de código de transformación enviado por el dispositivo portátil solicitante. Luego, el proxy de sincronización procesa la transformación de datos si es necesario, mediante la ejecución de este código de transformación durante el procesamiento de la lógica de sincronización mientras sincroniza esta aplicación. Aunque el presente invento no especifica con qué lenguaje de programación se ha elaborado la lógica de sincronización, la propuesta del presente invento en la que el proxy de sincronización recupera una lógica de sincronización de una unidad central distante y ejecuta esta lógica se puede considerar en términos de un procesamiento de web actual habilitada para Java donde un programa Java tal como una miniaplicación Java se pueda descargar mediante un navegador desde una unidad central distante y ejecutarse por el navegador localmente para el procesamiento de la web.

En una realización del presente invento, el procedimiento de sincronización para una aplicación se divide en lógica de sincronización y código de transformación. La lógica de sincronización es el procedimiento específico de aplicación de las tareas de sincronización, mientras que el código de transformación es el proceso de conversión dependiente de dispositivos entre dos tipos de dispositivos. La separación de estos dos significa que el realizador de esta aplicación puede proveer y mantener una aplicación, mientras que el código de transformación se puede proveer y mantener por el fabricante del dispositivo portátil al que se aplica esta transformación. Los proxies de sincronización ya no tienen que guardar y mantener estos procedimientos de sincronización o transformación. Todo lo que necesitan hacer es recuperar los códigos apropiados de lógica de sincronización y/o transformación durante la sincronización. Esta solución perfecciona en gran escala el proceso de gestión de los procedimientos de sincronización y/o transformación dependientes de una aplicación y dependientes de un dispositivo por parte de los proxies de sincronización.

Dicha solución exime también a los realizadores de los proxies de sincronización de tener que desarrollar los procedimientos de sincronización que sean específicos para las aplicaciones y los dispositivos. Como resultado, los proxies de sincronización pueden ser más escalables en cuanto a soportar más aplicaciones, dispositivos, y sistemas de información de gestión, siempre que los procedimientos de sincronización dependientes de la aplicación y del servicio se hayan realizado y mantenido adecuadamente y se hayan hecho recuperables por sus respectivos realizadores.

Como una propuesta preferible de una realización preferida del presente invento, un proxy de sincronización podría desplegar una memoria caché para guardar temporalmente la lógica de sincronización o el código de transformación para una aplicación por anticipado para que estos datos ejecutables puedan usarse por parte de este proxy de sincronización en el futuro próximo. Más adelante se describe un ejemplo de la propuesta de memoria caché del presente invento. En primer lugar, el proxy de sincronización deja a un lado un bloque de memoria (ya sea memoria principal o espacio de disco) como la memoria caché. El proxy de sincronización podría adoptar un método de indicación para buscar y recuperar una lógica de sincronización o un código de transformación guardados en memoria caché (por ejemplo, basándose en sus nombres exclusivos). Para recuperar una información guardada en memoria caché (la lógica de sincronización o el código de transformación para una aplicación), la lógica de sincronización, antes de salir a la red para recuperarla, primero busca su memoria caché. Si no se encuentra la información en la memoria caché, entonces el proxy de sincronización va a la dirección de red en relación de asociación con el ID de esta información para recuperarlo y actualiza con él la memoria caché del proxy. Si se encuentra la información en la memoria caché, el proxy de sincronización recupera esta información de su memoria caché en lugar de recuperarla de la red. Luego, el proxy de sincronización ejecuta estos datos ejecutable (la lógica de sincronización o el código de transformación para una aplicación) cargando estos datos de la memoria caché para su ejecución.

Como otra propuesta preferible de una realización preferida del presente invento, un proxy de sincronización puede monitorizar continuamente su carga informática (por ejemplo, en función del número de solicitudes de sincronización que está procesando al mismo tiempo) contra su capacidad informática que se fija basándose en su potencia de CPU, en el tamaño de su RAM y de su memoria de almacenamiento, y en su capacidad de ancho de banda de red. Todos los proxies de sincronización que estén dentro de la red pueden participar en un compartimiento en tiempo real entre sí de su carga informática actual y de su capacidad informática actual preconfigurada. Tras recibir una solicitud de sincronización, un proxy de sincronización comprueba primero si la adición de esta tarea de sincronización a la carga informática actual excede de la capacidad informática de este proxy. Si es así, este proxy busca otros proxies de la misma red en cuanto a la información sobre su carga informática y capacidad informática. Este proxy, después de conseguir la información, puede luego desviar esta solicitud de sincronización que entra a otro proxy (de la misma red) cuya carga informática sea menor. Si todos los proxies de la misma red tienen una carga informática completa, entonces el proxy que reciba esta solicitud de sincronización puede devolver al dispositivo portátil solicitante un mensaje que

indique que la capacidad para todos los proxies de sincronización está completa. El proxy termina también la conexión con el dispositivo portátil solicitante. En este caso no se realizan procesos de sincronización. Con esta propuesta, el presente invento proporciona también una función de equilibrado de carga de la que no se puede disponer en la solución clara de proxy de sincronización

Los anteriores, y adicionalmente, los objetos, ventajas, y prestaciones del invento resultarán más evidentes a partir de la siguiente descripción detallada de una realización preferida y de los dibujos incluidos como apéndice, en los que:

La Figura 1 presenta un ejemplo de una arquitectura de la solución de proxy de sincronización;

La Figura 2 presenta un ejemplo de una arquitectura de un cliente implementado como un dispositivo portátil que tiene prestaciones del presente invento;

La Figura 3 presenta un ejemplo de una arquitectura de un proxy de sincronización del presente invento;

La Figura 4 presenta un ejemplo de una arquitectura de una propuesta de unidad central de réplica del presente invento;

La Figura 5 representa un ejemplo de una propuesta de aceptador portátil de sincronización del presente invento;

La Figura 6 representa un ejemplo de una propuesta de aceptador de sincronización de proxy del presente invento;

La Figura 7 representa un ejemplo de una propuesta de manipulador de sincronización portátil del presente invento;

La Figura 8 representa un ejemplo de una propuesta de manipulador de sincronización de proxy del presente invento;

La Figura 9 representa un ejemplo de una propuesta de gestor de memoria caché de proxy del presente invento;

La Figura 10 representa un ejemplo de un manipulador de sincronización de proxy que procesa una transformación de datos;

La Figura 12 representa un ejemplo de un gestor de réplica en una réplica de unidad central o un gestor de datos en un dispositivo portátil; y

La Figura 13 representa un ejemplo de una propuesta de gestor de carga de proxy del presente invento.

La Figura 1 representa un ejemplo de una arquitectura global de una red que despliega la solución de servidor de sincronización que tiene prestaciones del presente invento. Un cliente (101, 102) cuyos ejemplos incluyen, pero sin carácter limitativo, dispositivos portátiles o de bolsillo (también denominados un terminal portátil, un ordenador de bolsillo, o un aparato para Internet). Un dispositivo portátil se refiere en general a cualquier dispositivo basado en ordenador suficientemente pequeño para asirlo con una mano mientras se está haciéndole funcionar con la otra. Los dispositivos portátiles podrían tener presentaciones visuales de menor tamaño, porque la naturaleza del trabajo para el que se han diseñado no requiere que se muestre una gran cantidad de información a la vez. Los dispositivos portátiles contienen comúnmente equipos de comunicaciones que les permiten comunicar y/o sincronizar con un ordenador central. Ejemplos de dispositivos portátiles incluyen, pero sin carácter limitativo, PALM PILOT™ de 3Com, ZAURUS™ de Sharp, PSION™ de Psion PLC, y otros diversos dispositivos del tipo "palma" que ejecutan WINDOWS CE™, un teléfono inteligente, o un buscapersonas inteligente, etc. Por tanto, aunque la realización preferida se refiere a dispositivos portátiles, los expertos en la técnica apreciarán que el presente invento puede ser beneficioso para cualquier dispositivo de cliente que sincronice datos sobre una red.

Los clientes se pueden conectar intermitentemente a servidores (105-107) (también denominados proxy, proxy de sincronización, servidor de sincronización, o servidor proxy de sincronización). Ejemplos de los servidores (105, 106, 107) incluyen, pero sin carácter limitativo: un PC; un puesto de trabajo (tal como una IBM RS6000™); o un ordenador central. Una unidad central de réplica (109, 110, 111) puede ser cualquier ordenador que ejecute un sistema de gestión de información que mantenga réplicas de datos de los dispositivos portátiles. Una unidad central de réplica puede ser un PC, un puesto de trabajo, o un ordenador central, etc. Los proxies de sincronización se conectan con las unidades centrales de réplica a través de la red (108). Los dispositivos portátiles pueden estar sin conectar a la red durante una parte o la mayor parte del tiempo, pero tienen que permanecer conectados (103, 104) mientras estén realizando tareas de sincronización. Los expertos en la técnica se harán cargo de que un dispositivo portátil puede obtener una conexión a una red (tal como la Internet o una red de área local) llamando a un servidor de acceso distante a la red a través de un módem, o bien disponiendo de una conexión directa de puerto serie (por ejemplo, usando el gancho conmutador PALM PILOT™) a un ordenador (tal como un PC de mesa) que se conecta a la red. Como es convencional, un servidor de directorio (112) provee servicios que incluyen la preconfiguración y el almacenamiento de información para usuarios y proporciona un motor de búsqueda para recuperar dinámicamente la información a través de una red tal como Internet bajo solicitud. Ejemplos de estos servicios incluyen los servicios de directorio de Novell (en adelante NDS) de la casa Novell y la Directorio activa de Microsoft, de los que ambos proporcionan información de directorio a la que se puede acceder a través del protocolo ligero de acceso a directorios (en adelante LDAP) - un protocolo de Internet para tener acceso a información de directorio.

De acuerdo con el presente invento, la lógica de sincronización (120) y/o el código de transformación (130) se guardan a distancia. Un ejemplo de un servidor de sincronización con adaptación para sincronizar datos entre un cliente (101-102) y una réplica de unidad central distante (109-111), que guarda una réplica de los datos en el cliente, incluye: lógica de proxy de sincronización (Figura 3) para identificar la unidad central de réplica (109-111) y la lógica de sincronización (120), que es específica de aplicación a un tipo de datos en relación de asociación con el cliente y con la unidad central distante. La lógica de sincronización (120) se puede ubicar en cualquier lugar en una red distante al servidor de sincronización (105-107) y a la unidad central distante (109-111), en respuesta a una solicitud de sincronización. La lógica de proxy de sincronización está destinada a recuperar de la red (108) la lógica de sincronización distante; conectar a la unidad central distante basándose en la solicitud; y sincronizar los datos entre el cliente y la unidad central distante usando lógica de sincronización recuperada. La solicitud de sincronización puede incluir también información para identificar un código de transformación específico de dispositivo (130) para convertir datos entre el cliente y la unidad principal distante durante la sincronización, incluyendo medios para recuperar el código de transformación basándose en el código de transformación específico de dispositivo. En este caso, el código de transformación específico de dispositivo se puede ubicar local o distancientemente en cualquier lugar de la red.

Alternativamente, el servidor está destinado a transformar datos entre el cliente (101-102), y una réplica de unidad central distante (109-111) que guarda una réplica de datos en el cliente. En este caso, la solicitud se usa para identificar la unidad central de réplica distante y un código de transformación (130) aplicables para transformar los datos en la unidad central distante a los de un tipo de dispositivo en relación de asociación con el cliente. El código de transformación se puede ubicar en cualquier lugar en una red a distancia del servidor de sincronización y de la unidad central distante. El servidor recupera el código de transformación distante (130) basándose en la solicitud y transforma los datos entre el cliente y la unidad central distante basándose en el código de transformación específico de dispositivo. La solicitud puede incluir una solicitud de sincronización con información para identificar la lógica de sincronización (120), que es específico de aplicación a un tipo de datos en relación de asociación con el cliente y la unidad central distante, en respuesta a la solicitud de sincronización. El servidor recupera la lógica de sincronización; conecta a la unidad central distante basándose en la información; y sincroniza los datos entre el cliente y la unidad central distante durante la transformación.

La Figura 2 representa un ejemplo de una arquitectura global de un cliente implementada como un dispositivo portátil capaz de realizar sincronización de acuerdo con el presente invento. Como se ha representado, el dispositivo portátil incluye una CPU (201), una memoria principal (203) tal como una RAM volátil, y memoria para almacenamiento que use (202) por ejemplo RAM no volátil, una ROM, o un disco para guardar sistemas (tales como el sistema operativo), aplicaciones (tales como el correo electrónico y el software de calendario), y los datos (tales como el contenido de una agenda o de un bloc memorándum). La mayoría de los dispositivos portátiles guardan toda la información en RAM y ROM sin ningún disco. La memoria principal (203) guarda la lógica (204) de dispositivo portátil del presente invento que preferiblemente se ha realizado como un código ejecutable de ordenador que se podría cargar desde la memoria de almacenamiento (202) a la memoria principal (203). En este caso, la lógica (204) de dispositivo portátil del presente invento (204) incluye un iniciador (205) de sincronización (con detalles representados en la Figura 5), un manipulador (206) de sincronización (con detalles representados en la Figura 7), y un gestor (207) de datos (con detalles representados en la Figura 11). La lógica (204) de dispositivo portátil del presente invento se guarda inicialmente en la memoria de almacenamiento (202). Cuando se ha iniciado la tarea de sincronización, la CPU (201) carga la lógica de sincronización (en todo o en parte en una modalidad bajo demanda) en la memoria principal (202) y comienza a ejecutar esta lógica para sincronización.

La Figura 3 representa un ejemplo de una arquitectura de un dispositivo informático configurado como un proxy de sincronización del presente invento. Los ejemplos incluyen, pero sin carácter limitativo: un PC; un puesto de trabajo; un servidor; o un ordenador central. El proxy de sincronización puede incluir una CPU (301), dispositivos de almacenamiento (302) tales como discos, y una memoria principal (303) tal como una RAM. En este caso, la memoria principal (303) guarda la lógica (304) de proxy de sincronización, que preferiblemente se ha realizado como un código ejecutable de ordenador que se podría cargar desde los discos (302) a la memoria principal (303). En este ejemplo, la lógica (304) de proxy de sincronización incluye un aceptador (305) de sincronización (con detalles representados en la Figura 6), una lógica (306) de manipulador de sincronización (con detalles representados en la Figura 8), un gestor (307) de memoria caché (con detalles representados en la Figura 9), y un gestor (308) de carga (con detalles representados en la Figura 13).

La Figura 4 representa un ejemplo de una arquitectura de un dispositivo informático configurada como una unidad central de réplica del presente invento. Son ejemplos de una unidad central de réplica, pero sin carácter limitativo: un PC; un puesto de trabajo; un servidor; o un ordenador central. Según se ha representado, la unidad central de réplica puede incluir una CPU (401), dispositivos de almacenamiento (402) tales como discos, y memoria principal (403) tal como una RAM. La memoria principal (403) guarda la lógica (404) de unidad principal de réplica, que preferiblemente se ha realizado como un código ejecutable por ordenador que se podría cargar desde discos (402) a la memoria principal (403). La lógica (404) de unidad central de réplica incluye un aceptador (405) de réplica (con detalles representados en la Figura 12) y un gestor (406) de réplica (con detalles representados en la Figura 11).

Como es convencional, el dispositivo portátil necesita en primer lugar establecer una conexión de red antes de que puedan realizarse las tareas de sincronización. Hay muchas tecnologías disponibles en la conexión de un dispositivo portátil a la red tal como la Internet o una red de área local. Por ejemplo, una PALM PILOT™ tiene un software de protocolo de control de transmisión/protocolo de Internet (en adelante TCP/IP) que viene con el dispositivo. Cuando

un PALM PILOT™ está unido a un módem, puede ejecutar este software de TCP/IP para llamar a un proveedor de servicios de Internet (en adelante ISP) a través de una línea telefónica para obtener una conexión a Internet. Alternativamente, se puede conectar el dispositivo a través de un puerto serie a un PC de mesa que esté conectado a la red. Hay programas disponibles (como el servicio de acceso distante de Microsoft (en adelante RAS)) para habilitar un PC como un servidor de acceso de red y proveer una conexión a la red a cualquier dispositivo que se conecta a este puerto serie de PC. Aunque no constituye una parte del presente invento, el proceso de conexión a la red del dispositivo del cliente es un requisito previo de los métodos de sincronización realizados en el presente invento.

Después que el dispositivo establece una conexión a la red, este dispositivo puede realizar ahora las tareas de sincronización empezando el proceso de iniciador de sincronización representado por el iniciador (205) de sincronización en la Figura 2, y con más detalle en la Figura 5. Para comenzar el proceso de sincronización, el iniciador de sincronización envía primero una solicitud de sincronización a un proxy (501) de sincronización. Los expertos en la técnica observarán que se pueden conocer de antemano las direcciones de red de los proxies de sincronización mediante los dispositivos portátiles, y también que se puede preconfigurar por defecto una dirección de proxy de sincronización mediante los dispositivos portátiles. La información que el dispositivo portátil envía al proxy para solicitar sincronización puede incluir un código entendido mutuamente para indicar que se trata de una solicitud de sincronización, así como alguna información de identificación sobre el usuario (tal como el ID de usuario y la contraseña codificada o secreto compartido para autenticación y autorización) y el dispositivo y tipo de sistema del portátil. Aunque el método para autenticación y autorización no forma parte del presente invento, los expertos en la técnica observarán que se pueden incorporar tecnologías actuales de autenticación y autorización sobre la red al procedimiento de colaboración entre el iniciador de sincronización del portátil (501 en la Figura 5) y el aceptador de sincronización del proxy de sincronización (305 en la Figura 3).

Si se rechaza la solicitud de sincronización, el proceso de iniciador de sincronización termina la tarea (504) de sincronización. De lo contrario, el iniciador de sincronización determina la lista de aplicaciones que es necesario sincronizar (502). Las aplicaciones a sincronizar podrían incluir, por ejemplo, una aplicación de agenda, una aplicación de bloc memorándum, una aplicación de calendario, una aplicación de correo electrónico, o cualesquiera otras en el dispositivo portátil. Los expertos en la técnica pueden entender que la determinación de una lista de aplicaciones a sincronizar podría implicar una entrada de pantalla del usuario (tal como comprobación de las aplicaciones a sincronizar desde una interfaz de usuario), una recuperación de un conjunto por defecto de aplicaciones, o un software para seleccionar dinámicamente las aplicaciones a sincronizar. Una vez que se ha determinado la lista de aplicaciones a sincronizar, el proceso de iniciador de sincronización comienza un proceso (503) de manipulador de sincronización (con detalles representados en la Figura 7) para cada aplicación de la lista. La primera tarea del proxy de sincronización es iniciar el aceptador de sincronización (305 en la Figura 3 y descrito con más detalle con referencia a la Figura 4).

La Figura 6 representa un ejemplo de una prestación del aceptador de sincronización de proxy del presente invento. Como se ha representado, el aceptador de sincronización puede ser un bucle infinito en el que el proxy de sincronización recibe continuamente solicitudes de sincronización de los dispositivos portátiles (601). Tras recibir una solicitud de sincronización, el aceptador de sincronización podría en primer lugar comprobar si su propia carga informática ha alcanzado su capacidad (602). Si es así, el aceptador de sincronización inicia al gestor (603) de carga para que expida esta solicitud a otro proxy de sincronización de la misma red (detalles representado en la Figura 13). Si el proxy de sincronización no está sobrecargado, el aceptador de sincronización puede realizar un procedimiento convencional de autenticación y autorización basándose en la información de identificación contenida en los mensajes de solicitud recibidos de los dispositivos portátiles solicitantes. Si falla el procedimiento de autenticación y autorización (por ejemplo, el usuario de portátil no ha podido autenticar o no está autorizado para sincronización por este proxy de sincronización), el proxy envía un mensaje de rechazo de vuelta al portátil (605), termina la conexión (606) y vuelve a esperar a la siguiente solicitud de sincronización (601). Si el procedimiento de autenticación y autorización se ha realizado de forma satisfactoria, el proxy envía entonces un mensaje de aceptación de vuelta al dispositivo portátil solicitante (607). A continuación, el proxy inicia al manipulador de sincronización de proxy con el fin de sincronizar para esta tarea de sincronización (608).

Los expertos en la técnica se harán cargo de que el proceso de aceptador de sincronización se implementa preferiblemente con un concepto de flujo múltiple de control de programa que usa tecnología actual de software. Esto significa que, tras recibir una solicitud desde un portátil (601), el aceptador de sincronización comienza un nuevo flujo de control de programa para procesar las etapas restantes (602-608) para esta solicitud. Las tres líneas con puntas de flecha, desde 603 hasta 601, desde 606 hasta 601, y desde 608 hasta 601, simplemente indican una terminación de este flujo de control. Los expertos en la técnica observarán también que en un sistema operativo que no soporte flujos múltiples de control de programa, la creación de un nuevo flujo de control (601) se puede considerar como la creación de un proceso nuevo. Por tanto, la terminación de un flujo de control es equivalente a la terminación de un proceso.

Volviendo ahora a los dispositivos portátiles, con referencia también a la Figura 5, el manipulador de sincronización para cada una de las aplicaciones a sincronizar se inicia secuencialmente (503) una vez que se ha recibido del proxy (605) el mensaje de aceptación. La Figura 7 representa un ejemplo más detallado de la lógica de manipulador de sincronización (para una aplicación)

Como se ha representado en la Figura 7, el manipulador de sincronización para una aplicación específica en primer lugar recupera información en relación de asociación con esta aplicación (702). La información incluye preferible-

mente el ID de la unidad central de réplica, el ID de la lógica de sincronización, y el ID del código de transformación para esta aplicación (701).

Una unidad central de réplica para una aplicación es el dispositivo de ordenador que guarda la réplica de esta aplicación. El dispositivo puede ser un PC, un puesto de trabajo, un servidor o un ordenador central, y debe estar conectada a la red. El ID de la unidad central de réplica puede incluir la dirección de red de la unidad central de réplica y la ubicación de la réplica de esta aplicación dentro de la unidad central de réplica. Los expertos en la técnica observarán que, basándose en la tecnología actual de Internet, un localizador universal de recursos (en adelante URL) es un modo razonable de implementar el ID de la unidad central de réplica.

La lógica de sincronización para una aplicación, preferiblemente realizada como código ejecutable, se puede usar para resolver conflictos, para especificar dirección de desbordamiento, o para tomar ciertas medidas para satisfacer las necesidades específicas para esta aplicación. Por ejemplo, la lógica de sincronización para una aplicación de mantener gastos podría especificar que el portátil siempre domina a la unidad central de réplica (tal como un servidor de base de datos corporativos) para ciertos archivos (como los gastos de hotel) puesto que los archivos corporativos representan precios sugeridos, mientras que los archivos del portátil especifican las facturaciones reales. El ID de la lógica de sincronización puede incluir un nombre exclusivo de la lógica de sincronización, la dirección de red y la ubicación dentro de esta dirección donde se guarda esta lógica de sincronización. Los expertos en la técnica observarán que, usando la tecnología actual de Internet, de nuevo un URL es una forma razonable para implementar el ID de lógica de sincronización.

El código de transformación para un dispositivo y una combinación de aplicaciones específicos, preferiblemente realizado como un código ejecutable, se puede usar para filtrar, transformar y cambiar datos de tal manera que se pueda usar cierta información de un modo factible y procesar en dispositivos informáticos que tengan diferentes potencias de CPU, capacidades de memoria, y construcciones físicas. Por ejemplo, una página web con imágenes que se envíen desde una unidad central de réplica de PC a un cliente podría ejecutar un código de transformación para que todas las imágenes o bien se eliminasen (función de filtro) o bien se redujesen a una resolución muy poco fina (función de transformación) para que esta página se pueda mostrar en la pantalla del cliente sin absorber demasiada memoria. El ID del código de transformación para una aplicación puede incluir un nombre exclusivo del código de transformación, la dirección de red y la ubicación dentro de esta dirección donde se guarda el código de transformación. Los expertos en la técnica observarán también que, usando la tecnología actual de Internet, aún resulta de nuevo razonable usar un URL para implementar el ID de código de transformación.

De acuerdo con otra prestación del presente invento, los tres números de identificación (ID) antes mencionados para cada aplicación se preconfiguran y guardan en los dispositivos portátiles o en un servicio de directorio para el usuario de este dispositivo portátil. Si estos ID para la aplicación se guardan en el dispositivo portátil, el manipulador de sincronización envía el identificador de sincronización, que preferiblemente incluye el nombre de la aplicación a sincronizar junto con los tres ID del proxy (701) de sincronización y luego entra en un bucle (702). Si estos ID se guardan por medio de un servicio de directorio, el manipulador de sincronización envía al proxy (701) de sincronización el identificador de sincronización que incluye la identificación del usuario, el dispositivo, la aplicación, y la dirección del servidor de directorio (que el proxy de sincronización usará para acceder al servicio de directorio).

Los expertos en la técnica observarán que se puede obviar la necesidad de enviar uno cualquiera de los tres ID mediante el uso de las configuraciones por defecto. Por ejemplo, si un dispositivo portátil configura con un proxy de sincronización una aplicación por defecto a sincronizar, entonces este proxy de sincronización no necesita recuperar el nombre de la aplicación a sincronizar durante el procesamiento de sincronización para este dispositivo portátil.

La falta de un nombre de aplicación indica en este caso la aplicación por defecto a sincronizar.

Similarmente, el usuario de un dispositivo portátil puede configurar también con un proxy de sincronización una unidad central de réplica o dispositivo por defecto. En estos casos, el proxy de sincronización no necesita recuperar esta información ni desde el dispositivo portátil ni del directorio. En su lugar, puede usar su propio reglaje por defecto preconfigurado para este dispositivo portátil para obtener esta información.

Aunque el ID de información tal como la unidad central de réplica, la lógica de sincronización, o el código de transformación se han descrito anteriormente como que incluyen una dirección de red y el nombre y ubicación exactos con sus respectivas direcciones, una implementación alternativa de un ID no requiere la especificación de su correspondiente dirección de red. En este caso, es de la responsabilidad del proxy de sincronización encontrar la dirección exacta de red de un ID particular basándose en el nombre incluido en este ID. Un ejemplo es que un dispositivo portátil hace una solicitud de sincronización para una aplicación en donde el ID para el código de transformación no incluye la dirección de red del código de transformación apropiado para este dispositivo. El proxy de sincronización, tras recibir el ID del código de transformación, determina la ubicación apropiada de red donde reside el código de transformación apropiado basándose en el tipo del dispositivo que realiza la solicitud de sincronización.

Refiriéndose de nuevo a la Figura 7, dentro del bucle (702), el manipulador de sincronización en primer lugar espera por una llamada de interfaz de programador de aplicaciones (en adelante API) procedente del proxy (702) de sincronización. Si la llamada del proxy indica que se ha completado el proceso de sincronización para esta aplicación el manipulador de sincronización termina, y regresa al iniciador de sincronización para procesar la siguiente aplicación.

a sincronizar. Si la llamada de API es una función de gestión de datos (tal como abrir una base de datos, extraer información de un archivo, etc.), el manipulador de sincronización inicia al gestor de datos para procesar esta API (703) (con descripciones más detalladas en la Figura 11).

El aceptador de sincronización (Figura 6) instalado en el proxy de sincronización - después que se ha completado el procedimiento de autenticación y autorización y se ha aceptado la solicitud de sincronización - podría enviar un mensaje de aceptación al portátil solicitante (605 en la Figura 6) y luego arrancar el manipulador de sincronización (606 en la Figura 6) de proxy para procesar la sincronización.

La Figura 8 representa un ejemplo de una ilustración detallada de un manipulador de sincronización de proxy. Como se ha representado, el manipulador de sincronización de proxy podría recibir en primer lugar del dispositivo portátil solicitante el nombre de la aplicación a sincronizar y los tres ID respectivamente para la unidad central de réplica, la lógica de sincronización, y el código de transformación (801). La información se puede recibir directamente del dispositivo portátil que hace la solicitud de sincronización, o a través de un servidor de directorio según se ha descrito anteriormente. Si no hay más aplicaciones a sincronizar, el portátil podría enviar un mensaje de terminación para así indicarlo. Si se recibe uno de dichos mensajes de terminación en lugar de los tres ID, el manipulador de sincronización termina el proceso de sincronización. De lo contrario, el manipulador de sincronización podría solicitar al gestor de memoria caché que recupere la lógica de sincronización basándose en el ID de lógica de sincronización (802). En la Figura 9 se describe un ejemplo de la gestión de memoria caché. Después que se ha recuperado la lógica de sincronización, el manipulador de sincronización podría entonces establecer una conexión a la réplica de esta aplicación basado en el ID de la unidad central de réplica (803). Una vez que se ha establecido la conexión, el manipulador de sincronización puede empezar a ejecutar la lógica de sincronización para realizar la tarea de sincronización para esta aplicación (804). Con referencia a la Figura 10 se describirá un ejemplo más detallado de la ejecución de la lógica de sincronización.

La Figura 9 representa un ejemplo del gestor de memoria caché desplegado por el proxy de sincronización para gestionar la introducción en memoria caché y la recuperación de objetos tales como la lógica de sincronización y los códigos de transformación para aplicaciones de portátil. Tras recibir una solicitud para un objeto (901), el gestor de memoria caché comprueba si este objeto está en la memoria caché. La memoria caché puede estar o bien en la memoria principal (303) o bien en discos (302).

Si el objeto solicitado no está en la memoria caché, el gestor de memoria caché recupera este objeto de la red basándose en el ID de este objeto (902). Una vez que se ha recuperado de la red, el objeto se introduce en la memoria caché (903) y se devuelve al solicitante (904). Si el objeto se encuentra en la memoria caché, el gestor de memoria caché comprueba si el objeto contenido en la memoria caché todavía es actual. Si no lo es, el gestor de memoria caché lo recupera de la red usando su ID (902). Después que se ha recuperado de la red el objeto actualizado, sustituye al antiguo en la memoria caché (903) y se devuelve al solicitante (904). Si el objeto está actualizado, se devuelve al solicitante (904).

Aunque la validación de actualidad del objeto contenido en la memoria caché no es una prestación del presente invento, los expertos en la técnica se harán cargo de que se pueden adoptar muchas de tales técnicas actuales mediante la prestación de gestor de memoria caché del presente invento. Por ejemplo, un gestor de memoria caché podría adoptar una política tal que no se considerase que un objeto estuviese actualizado si está en la memoria caché durante un período fijo de tiempo. Otro ejemplo es que un gestor de memoria caché podría coincidir con el proveedor original del objeto de tal manera que, cada vez en que un objeto original se actualice (o se cree o se elimine), el proveedor podría enviar un mensaje al gestor de memoria caché sobre el cambio de este objeto. El gestor de memoria caché, tras recibir tal mensaje, podría entonces o bien eliminar este objeto de la memoria caché, o bien marcarlo como inválido, o recuperar el actualizado y sustituir el antiguo en la memoria caché.

En la Figura 10 se ha ilustrado con más detalle un ejemplo de la ejecución de la lógica de sincronización (904) mediante la prestación de manipulador de sincronización de proxy (Figura 8). Según se ha representado, la lógica de sincronización implica la extracción de datos de - y la introducción de datos en - el dispositivo portátil (1003) que hace la solicitud de sincronización, y la unidad central de réplica (1004) que centraliza la réplica de la aplicación de esta lógica de sincronización para el dispositivo portátil solicitante. Podrían existir muchas diferencias entre los sistemas de gestión de datos de un dispositivo portátil y el sistema de gestión de información desplegado en la unidad central de réplica de este dispositivo portátil (por ejemplo, correo electrónico PALM PILOT™ frente a correo electrónico Lotus NOTES™). Por ejemplo, los formatos de datos podrían ser diferentes. Las capacidades de memoria entre el dispositivo portátil y la unidad central de réplica podrían ser diferentes. (Por ejemplo, PALM PILOT™ frente a puesto de trabajo). Los dispositivos de presentación visual entre ellos podrían también ser diferentes (por ejemplo, pantalla PALM PILOT™ en blanco y negro de baja resolución frente a un monitor de PC de alta resolución). La necesidad de transformar apropiadamente la información procedente de un extremo (por ejemplo, una unidad de réplica de puesto de trabajo) hasta el otro (por ejemplo, un PALM PILOT™) se satisface en el presente invento mediante la ejecución del código de transformación.

Durante la ejecución de la lógica de sincronización para una aplicación (Figura 10), la lógica de sincronización determina la dirección de actualización para cada elemento de datos de esta aplicación en el dispositivo portátil y en la unidad central de réplica. Por ejemplo, para dos elementos de datos del mismo ID exclusivo del dispositivo portátil y de la unidad central de réplica respectivamente, la lógica de sincronización podría determinar que el elemento de

datos del portátil domine al de unidad central de réplica, y viceversa. La lógica de sincronización podría determinar también que estos dos elementos de datos son iguales y no es necesario movimiento de datos. Es posible también que la lógica de sincronización pueda averiguar que estos dos elementos plantean un conflicto (por ejemplo, ambos se han actualizado con independencia) y decida duplicar cada elemento mientras que al mismo tiempo retenga su propia versión.

En la prestación de manipulador de sincronización de proxy del presente invento, antes de que a un elemento de datos se le introduzca información de un dispositivo a otro (por ejemplo, desde la unidad central de réplica al dispositivo portátil, o viceversa), el manipulador de sincronización de proxy comprueba si es aplicable una transformación de datos. Si lo es, el manipulador de sincronización de proxy puede solicitar el código de transformación (especificado por el ID enviado desde el dispositivo portátil que hizo la solicitud de sincronización) del gestor (1003) de memoria caché. El gestor de memoria caché podría procesar los procedimientos ilustrados en la Figura 9 para hacer disponible el objeto solicitado. Una vez que está disponible el código de transformación para la aplicación que se está sincronizando, el manipulador de sincronización de proxy ejecuta el código para el elemento de datos a introducir al dispositivo (1003).

La Figura 11 representa un ejemplo de lógica para un gestor (207) de datos o un gestor (406) de réplica. El manipulador de sincronización de proxy (306 y Figura 8) puede usar técnicas convencionales para extraer datos (1103) de - e introducir datos (1104) a - los dispositivos portátiles y a sus unidades centrales de réplica por medio de sus API. Depende del gestor (207) de datos y del gestor (406) de réplica la interpretación de las llamadas de API recibidas del proxy de sincronización y la realización de las funciones apropiadas de gestión de datos o de réplica. Los expertos en la técnica apreciarán que estas llamadas de API son funciones típicas de gestión de datos tales como una extracción de información de un elemento de datos (1103), una introducción de información en un elemento de datos (1104), una actualización de un elemento de datos (1105), una eliminación de un elemento de datos, etc. Con el fin de procesar la sincronización de un modo más eficaz, se puede introducir información de la lógica de sincronización para aprovechar cualquier versión e información histórica de actualización disponible de los dispositivos portátiles y sus unidades centrales de réplica con objeto de acelerar el proceso de sincronización. Los expertos en la técnica apreciarán que las API antes mencionadas incluyen funciones que recuperan (1106) y/o configuran (1106) información histórica de versión y actualización de los dispositivos portátiles y de la unidades centrales de réplica.

La Figura 12 representa un ejemplo de la prestación de aceptador de réplica del presente invento (405).

El aceptador de réplica puede ser un proceso con bucles en el que en primer lugar se espera por una llamada de API (1201). Cuando llega una llamada de API del proxy de sincronización, el aceptador de réplica podría iniciar un proceso de autenticación y autorización para verificar la identidad del proxy de sincronización solicitante (1202). Si falla el proceso de autenticación y autorización, el aceptador de réplica rechaza la llamada de API (1203).

De lo contrario, el aceptador de réplica inicia al gestor (1202) de réplica para procesar la función apropiada de gestión de réplica.

La Figura 13 representa un ejemplo de la prestación de gestor de carga de un proxy de sincronización del presente invento (308). Cuando el aceptador de sincronización de proxy (Figura 6) recibe una solicitud de sincronización y detecta que este proxy de sincronización está sobrecargado, puede arrancar el gestor de carga para expedir esta solicitud de sincronización a otro proxy de sincronización (603). El gestor de carga, tras recibir una petición para expedir una solicitud de sincronización (1301), determina si hay otro proxy en la misma red que no esté sobrecargado (1302). Si se encuentra uno, el gestor de carga expide la solicitud de sincronización a este proxy de sincronización (1304). Si todos los proxies de la misma red tienen una carga completa, el gestor de carga rechaza esta solicitud de sincronización (1303). Los expertos en la técnica podrían apreciar que los gestores de carga de todos los proxies de sincronización podrían participar en un protocolo para intercambiar información entre sí. Basándose en esta información de carga procedente de otros proxies de la misma red, un gestor de carga podría adoptar también una política en la elección del proxy a expedir (por ejemplo, el gestor de carga podría elegir el proxy que tenga la mínima carga informática actual) para obtener un equilibrado global de carga sobre la red.

El manipulador de sincronización de proxy podría decidir la expedición de una solicitud de sincronización basándose no sólo en su estado de carga según se ha ilustrado en (602). Es posible que en una red que despliegue múltiples proxies de sincronización, cada proxy de sincronización esté especializado en la sincronización de datos para un sistema de gestión de información o dispositivo portátil específicos. Por ejemplo, en una red que soporte sincronización de datos para dispositivos portátiles que incluya un dispositivo basado en PALM PILOT™ y en WINDOWS CE™, se despliegan dos proxies de sincronización con el proxy 1 que es especialista en sincronización de datos para PALM PILOT™ y el proxy 2 que se especializa en sincronización de datos para Windows CE. De este modo, basándose en esta configuración, es probable que más lógicas de sincronización y códigos de transformación en relación de asociación con los dispositivos con PALM PILOT sean guardados en la memoria caché por el proxy 1 y más lógicas de sincronización y códigos de transformación en relación de asociación con los dispositivos de Windows CE™ sean guardados en la memoria caché por el proxy 2 a medida que pasa el tiempo. Tras la recepción de una solicitud de sincronización (601), un proxy de sincronización, en lugar de sólo comprobar la carga (602), puede comprobar también el tipo de dispositivo y, si no está especializado en este tipo de dispositivo, pueda expedir la solicitud al proxy que esté especializado en este tipo de dispositivo. Por ejemplo, el proxy 1, tras recibir una solicitud de sincronización de un dispositivo con Windows CETM, podría expedir esta solicitud al proxy 2 que está especializado en la sincroniza-

ción de datos para Windows CETM y por tanto es más probable que tenga guardados en la memoria caché lógicas de sincronización actualizadas y códigos de transformación actualizados que sean necesarios para cumplimentar esta solicitud de sincronización.

En lugar de un esquema preconfigurado de división de especializaciones como se ha descrito anteriormente, los proxies de sincronización podrían entre ellos mismos emitir dinámicamente situación de memorias caché, de un modo similar a la condición de carga que comparten entre sí. De este modo, cada proxy de sincronización puede expedir dinámicamente una solicitud de sincronización a otro proxy de sincronización en el que el estado de su memoria caché sea más favorable para procesar esta solicitud.

Una realización preferida del presente invento incluye prestaciones que se pueden implementar como software realizado tangiblemente en un producto de programa de ordenador o dispositivo de almacenamiento de programa para ejecución en una CPU (201, 301) provista con el cliente (101, 102) y servidor (105, 107). Por ejemplo, un software implementado en un código popular ejecutable de ordenador orientado a objetos tal como JAVA™ de Sun proporciona portabilidad a través de plataformas diferentes. Los expertos en la técnica apreciarán que se pueden emplear también otros entornos de programa orientados a procedimiento y orientados a objeto (en adelante OO) incluyendo, sin carácter limitativo, C++ y Smalltalk.

Los expertos en la técnica se harán cargo también de que se podrían implementar métodos del presente invento como software para ejecución en un ordenador o en otro dispositivo basado en procesador. El software podría realizarse sobre un dispositivo de almacenamiento de datos magnético, eléctrico, óptico, u otro dispositivo de programa persistente, incluyendo, sin carácter limitativo: discos magnéticos, unidades de acceso directo compartido (en adelante DASD), memoria de burbujas; cinta; discos ópticos tales como CD-ROM y DVD (discos vídeo-digitales); y otro dispositivos persistentes de almacenamiento (también denominados no volátiles) tales como núcleo, ROM, PROM, memoria flash, o RAM reforzada con batería. Los expertos en la técnica se harán cargo de que se podría tener acceso y mantener directamente a uno o más de los componentes ejemplificados en la memoria (203) del cliente (101, 102) o servidor (105-107) por medio de disco (302), memoria de almacenamiento (202), la red (108), otro servidor, o se podrían distribuir entre una pluralidad de servidores.

Un proxy de sincronización o lógica de servidor de sincronización, proceso y aparato que emplee lógicas de sincronización y/o códigos de transformación distribuidos en la red para aumentar la escalabilidad y perfeccionar la capacidad de gestión de sincronización entre un cliente tal como: un aparato con Internet; un dispositivo portátil o de bolsillo; un teléfono inteligente o buscapersonas inteligente; y sus fuentes de réplica distantes. Un ejemplo de un servidor de sincronización con adaptación para sincronizar datos entre un cliente y una réplica distante de unidad central, que guarda una réplica de datos en el cliente, incluye lógica de aceptador de sincronización para identificar una unidad central de réplica y una lógica de sincronización que es específica de cliente para un tipo de datos en relación de asociación con el cliente y la unidad principal distante; en donde la lógica de sincronización puede ubicarse en cualquier lugar de una red distante al servidor de sincronización y a la unidad central distante, en respuesta a una solicitud de sincronización; y una lógica de manipulador de sincronización, acoplada a la lógica de aceptador de sincronización, para recuperar de la red la lógica de sincronización distante, conectar a la unidad central distante, y sincronizar los datos entre el cliente y la unidad central distante usando la lógica de sincronización recuperada. La solicitud de sincronización podría identificar explícitamente (o por defecto) la unidad central de réplica y la lógica de sincronización, o bien podría incluir una dirección de un servicio de directorio que guarde información para identificar la unidad central de réplica y la lógica de sincronización. La solicitud de sincronización podría también o incluir alternativamente información para identificar un código de transformación específico de dispositivo (local o distante) para convertir datos entre el cliente y la unidad central distante durante la sincronización; y medios para recuperar el código de transformación basándose en la información, y convertir los datos entre el cliente y la unidad central distante basándose en el código de transformación específico de dispositivo.

REIVINDICACIONES

1. Un servidor (107) de sincronización para sincronizar datos entre un cliente (102) y una réplica (110) de unidad central distante, que guarda una réplica de datos en el cliente, que comprende:

una lógica (305) de aceptador de sincronización para recibir del cliente una solicitud de sincronización, cuya solicitud incluye un primer identificador para identificar una unidad central de réplica, un segundo identificador para identificar una lógica (120) de sincronización que es específica de aplicación a un tipo de datos en relación de asociación con el cliente y la unidad central distante, y un tercer identificador para identificar un código (130) de transformación específico de dispositivo para transformar los datos en la unidad central distante a los de un tipo de dispositivo en relación de asociación con el cliente ; y

una lógica de manipulador (306), acoplada a la lógica de aceptador de sincronización, para recuperar de la red la lógica de sincronización, recuperar de la red el código de transformación, conectar a la réplica de unidad central distante, sincronizar los datos entre el cliente y la réplica de unidad central distante usando la lógica de sincronización recuperada, y transformar los datos entre el cliente y la réplica de unidad central distante basándose en el código de transformación específico de dispositivo;

en donde la lógica (120) de sincronización y el código (130) de transformación están ubicados de un modo distante entre sí de tal manera que la lógica de manipulador los recupere de diferentes ubicaciones usando los identificadores respectivos.

2. Un dispositivo de cliente destinado a la conexión al servidor de sincronización de la reivindicación 1, cuyo dispositivo de cliente comprende:

una unidad central de tratamiento (en adelante CPU);

una memoria, acoplada a la CPU, que guarda un código ejecutable para su ejecución en la CPU, cuyo código comprende:

un iniciador (205) de sincronización para enviar la solicitud de sincronización mencionada en la reivindicación 1 al servidor de sincronización de la reivindicación 1.

3. El cliente de la reivindicación 2, en donde el cliente es un dispositivo portátil desconectable seleccionado de un grupo que consiste en un ordenador portátil; un teléfono inteligente, o un buscapersonas inteligente, que comprende medios para establecer una conexión a la red.

4. En un servidor de sincronización, un método para sincronizar de forma adaptadora datos entre un cliente y una réplica de unidad central distante que guarda una réplica de datos en el cliente, cuyo método comprende las etapas de:

recibir del cliente una solicitud de sincronización, incluyendo la solicitud un primer identificador para identificar una unidad central de réplica, un segundo identificador para identificar una lógica (120) de sincronización que es específica de aplicación a un tipo de datos en relación de asociación con el cliente y la unidad central distante, y un tercer identificador para identificar un código (130) de transformación específico de dispositivo para transformar los datos en la unidad central distante a los de un tipo de dispositivo en relación de asociación con el cliente;

recuperar de la red la lógica de sincronización;

recuperar de la red el código de transformación;

conectar a la réplica de unidad central distante;

sincronizar los datos entre el cliente y la réplica de unidad central distante usando la lógica recuperada de sincronización; y

transformar los datos entre el cliente y la réplica de unidad central distante basándose en el código de transformación específico de dispositivo;

en donde la lógica (120) de sincronización y el código (130) de transformación están ubicados en lugares distantes entre sí de tal manera que la lógica de manipulador los recupere de diferentes ubicaciones usando los identificadores respectivos.

5. Un programa de ordenador que comprende instrucciones para realizar todas las etapas del método de la reivindicación 4, cuando dicho programa de ordenador se ejecuta en un sistema de ordenador.

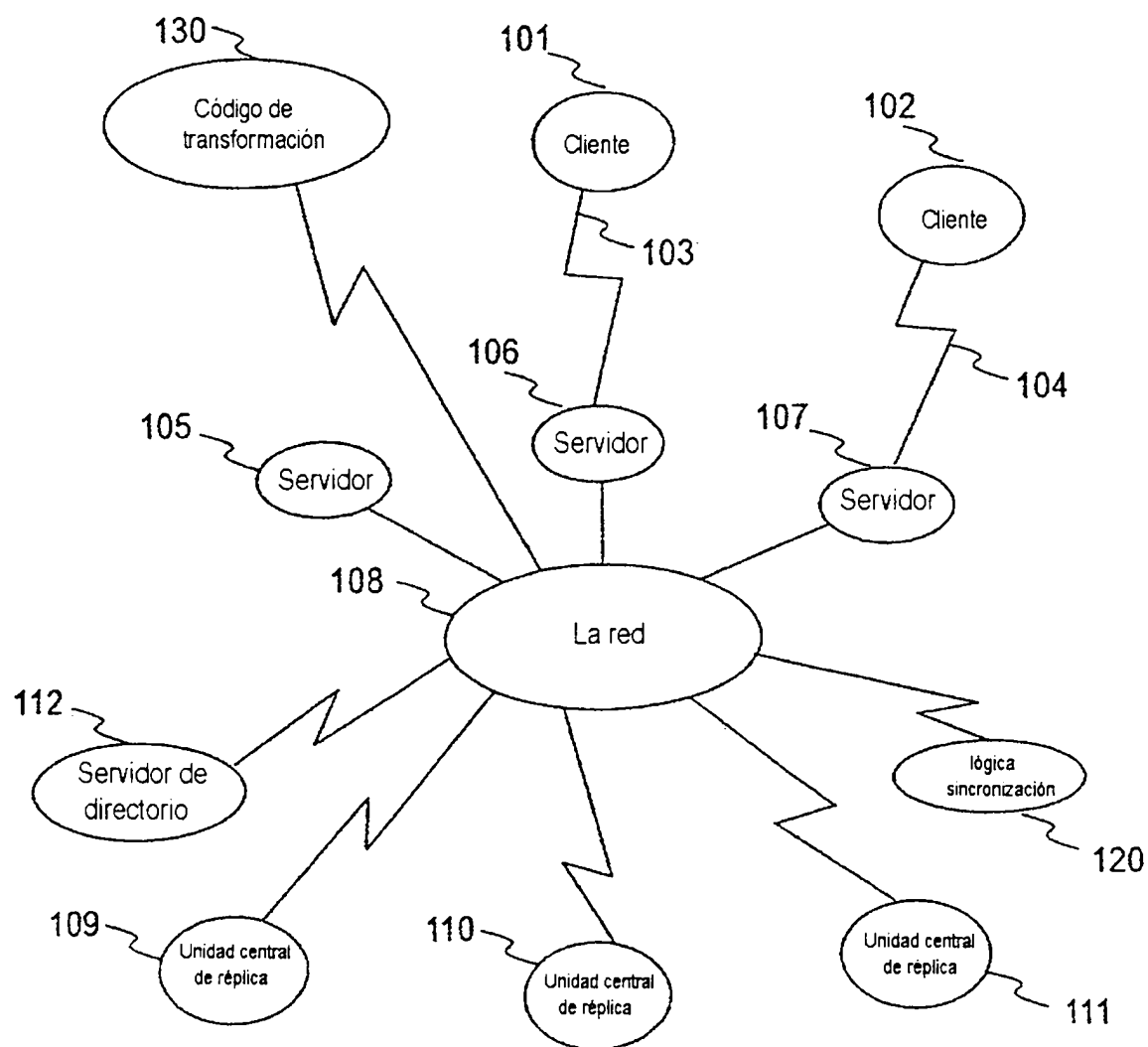


Fig.1

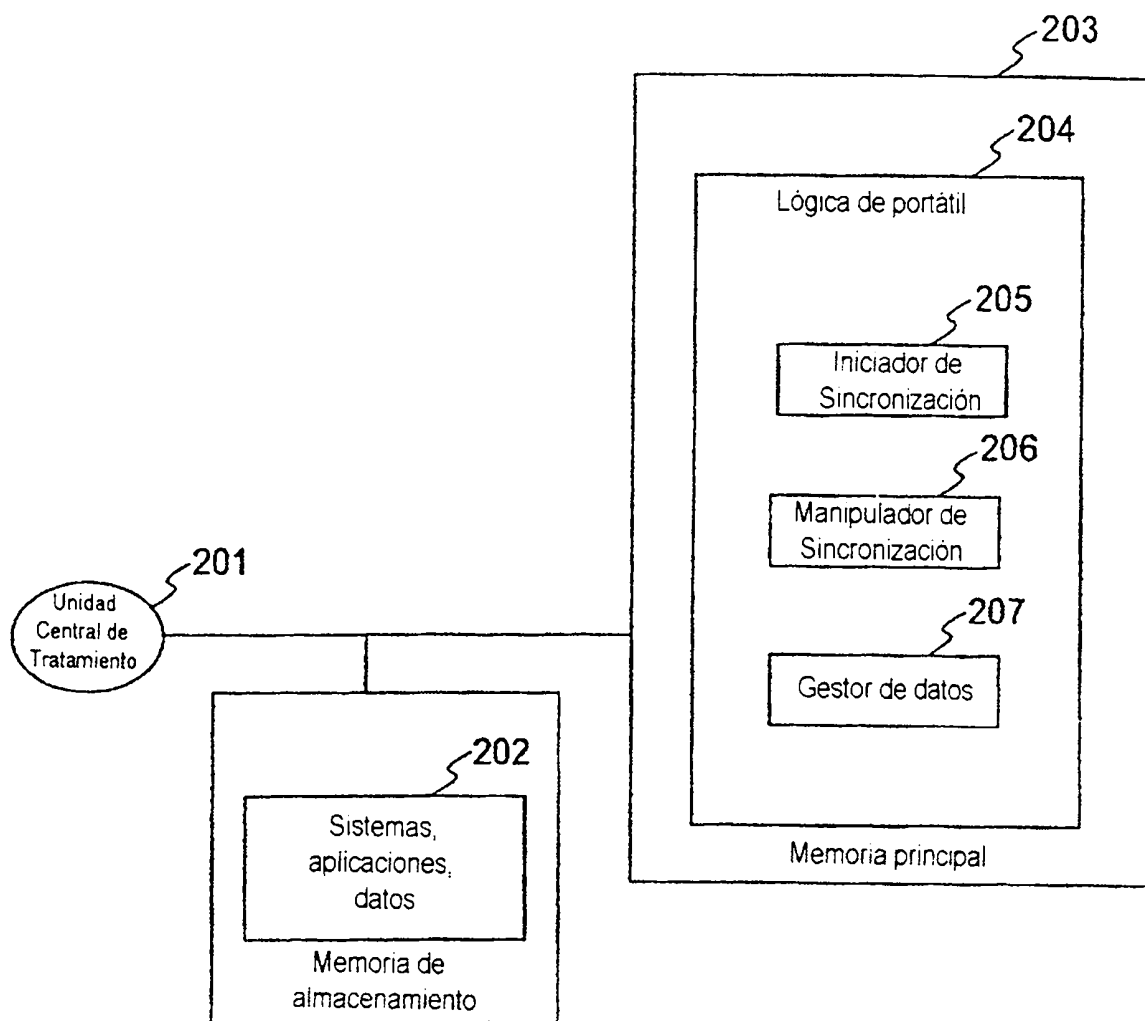


Fig.2

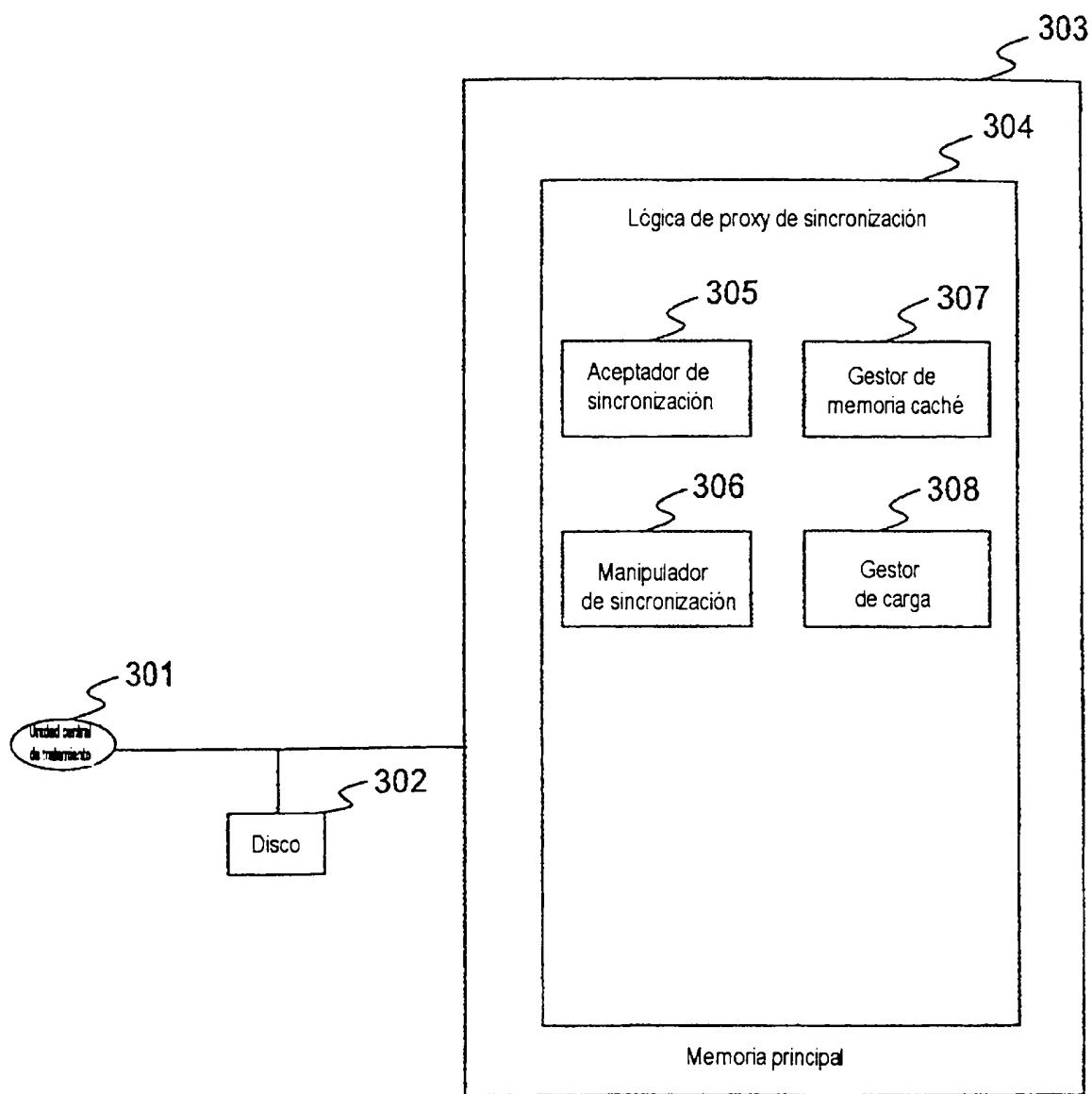


Fig.3

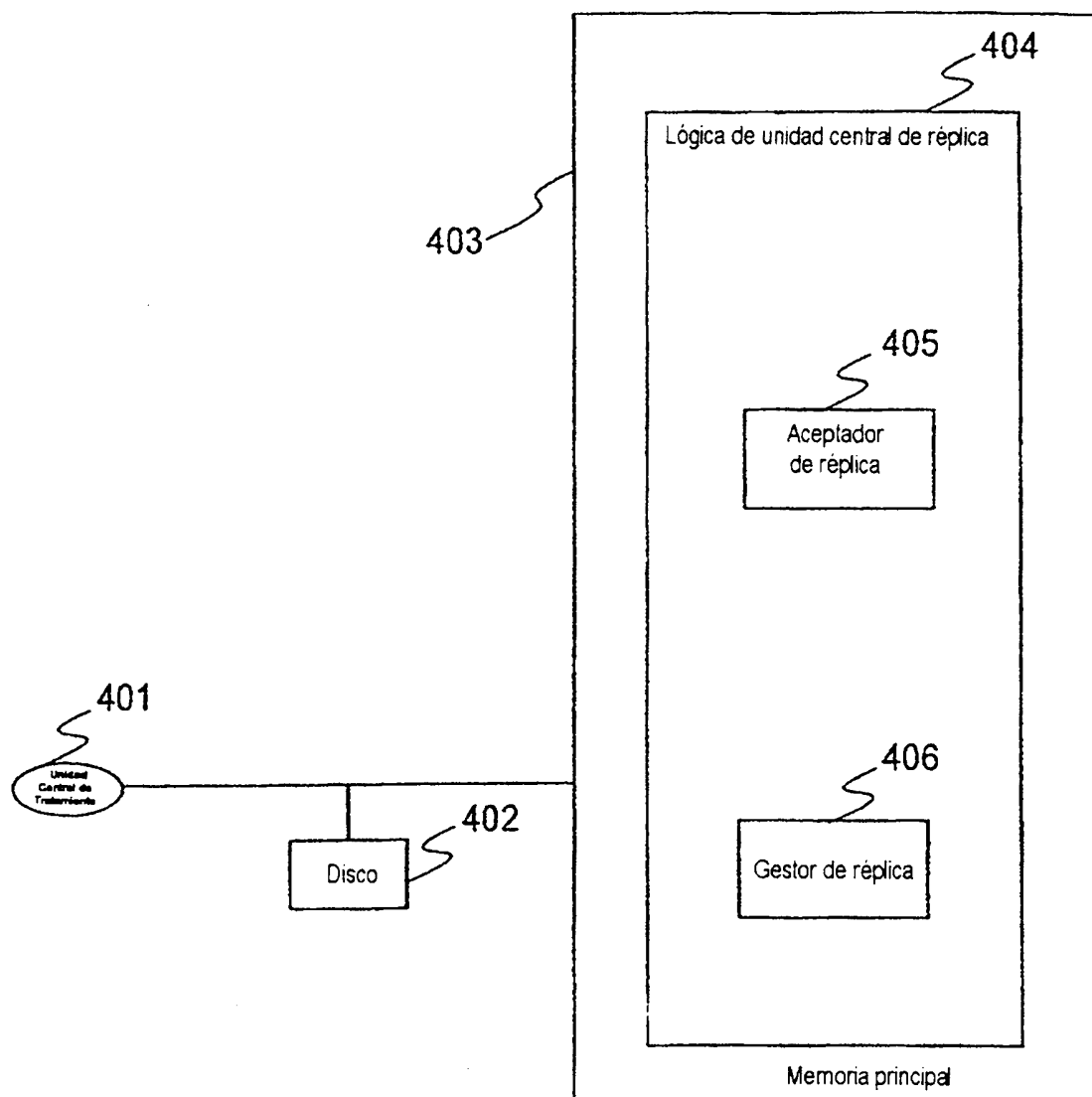


Fig.4

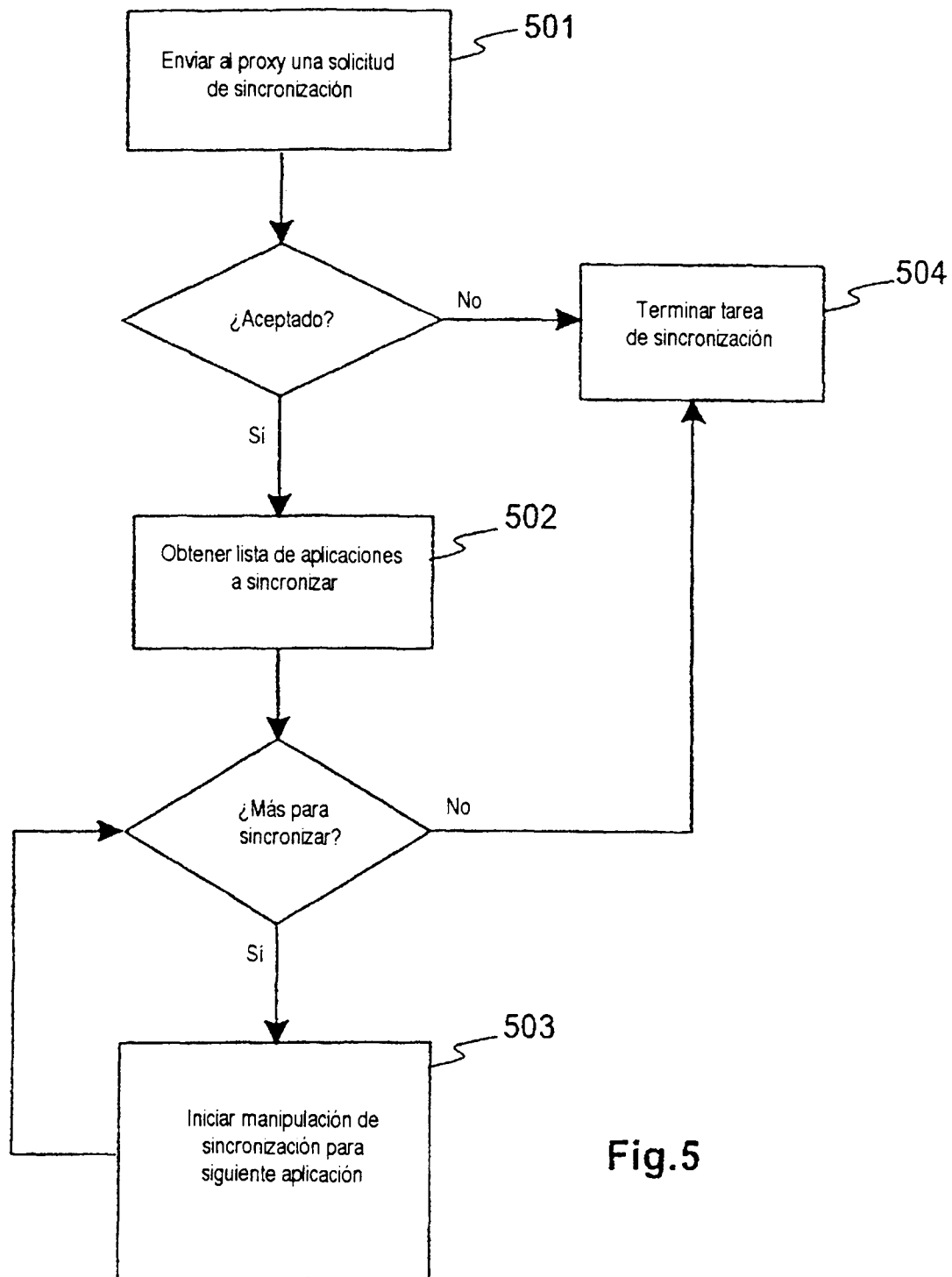


Fig.5

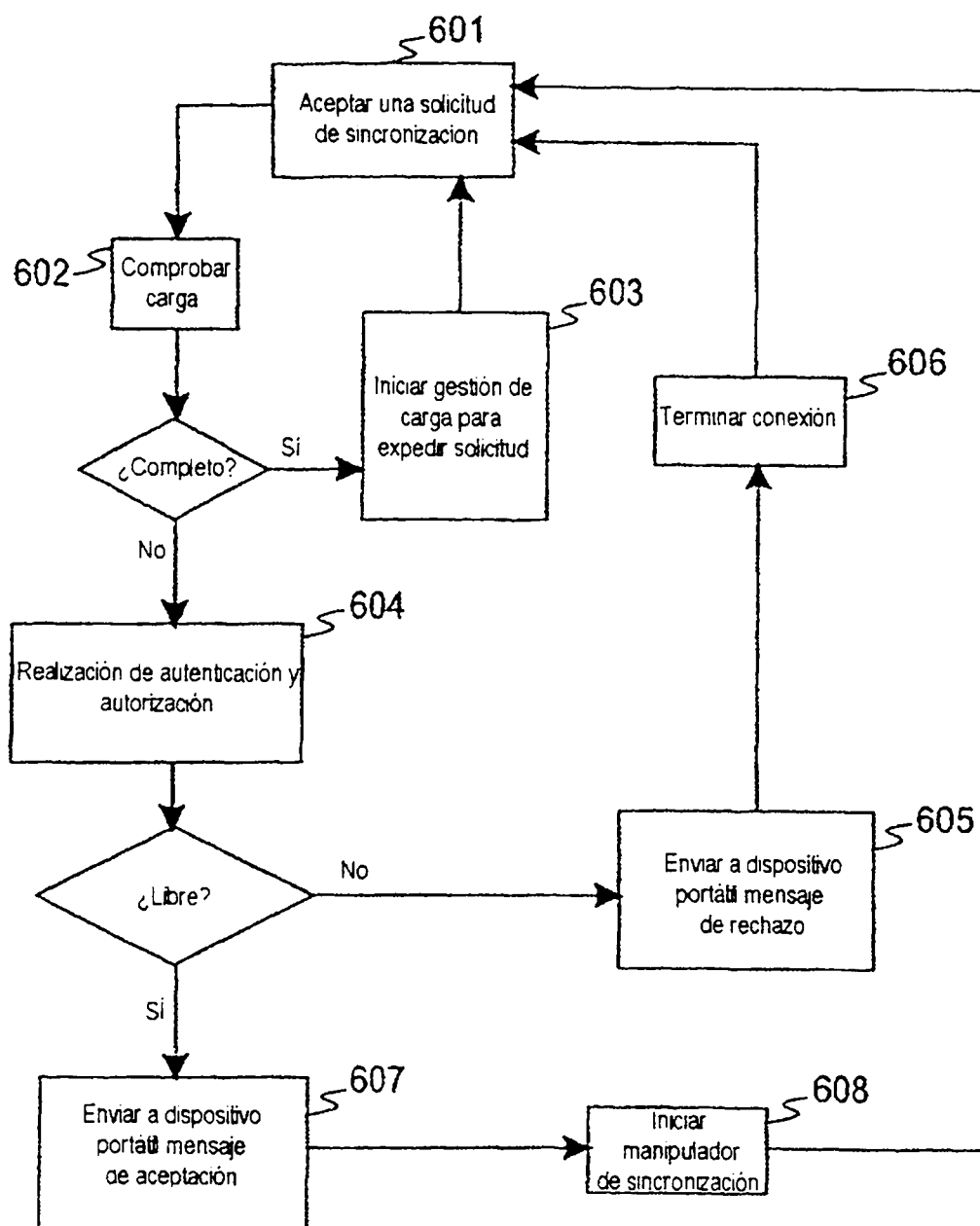


Fig.6

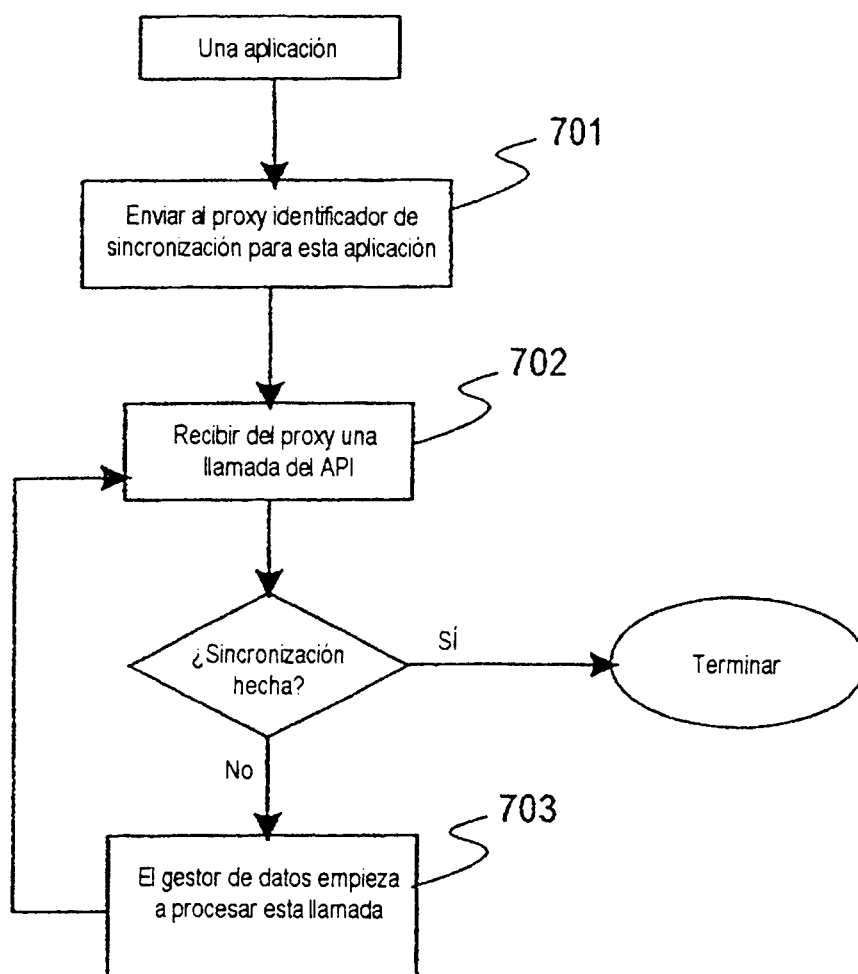


Fig. 7

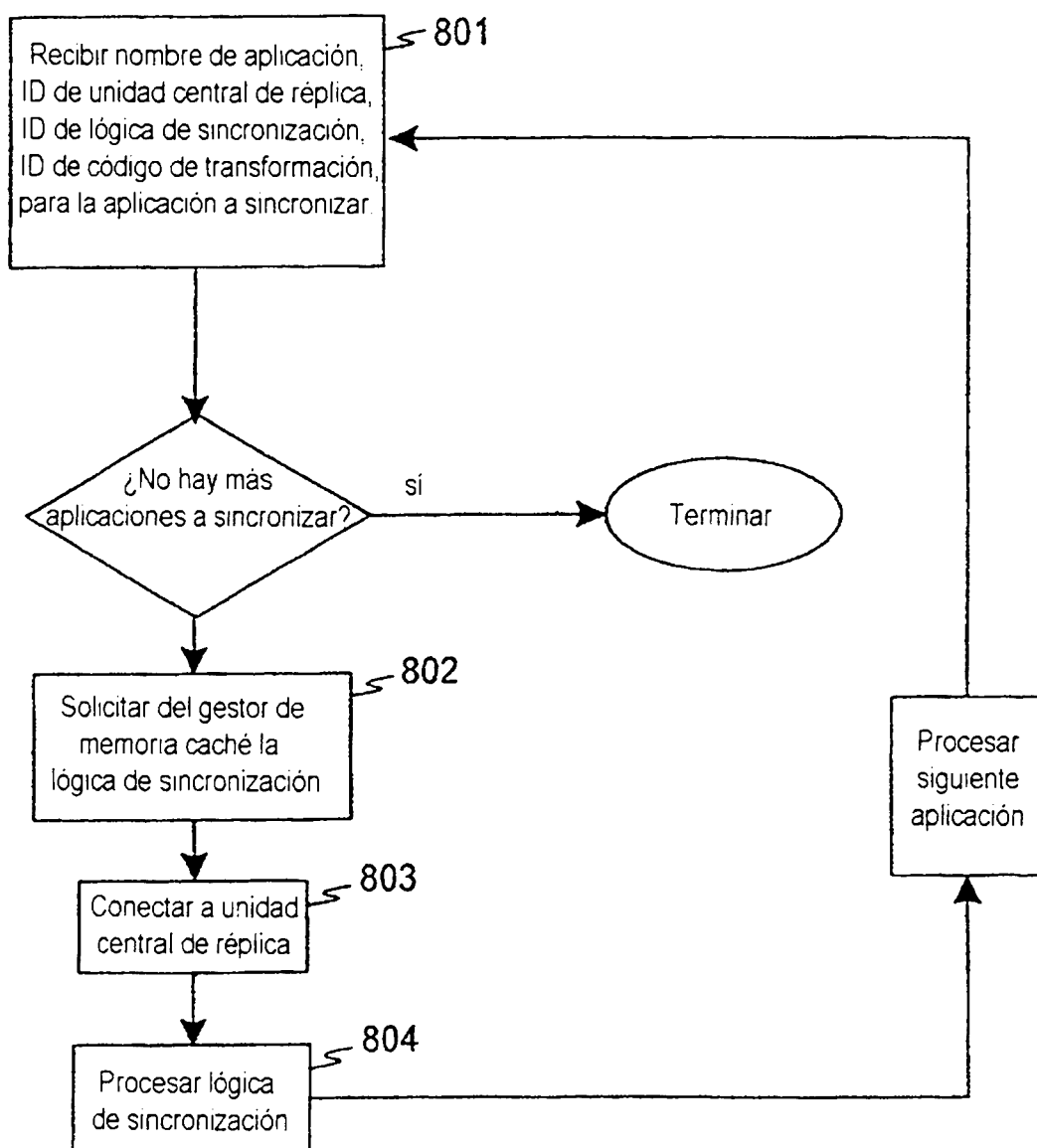


Fig.8

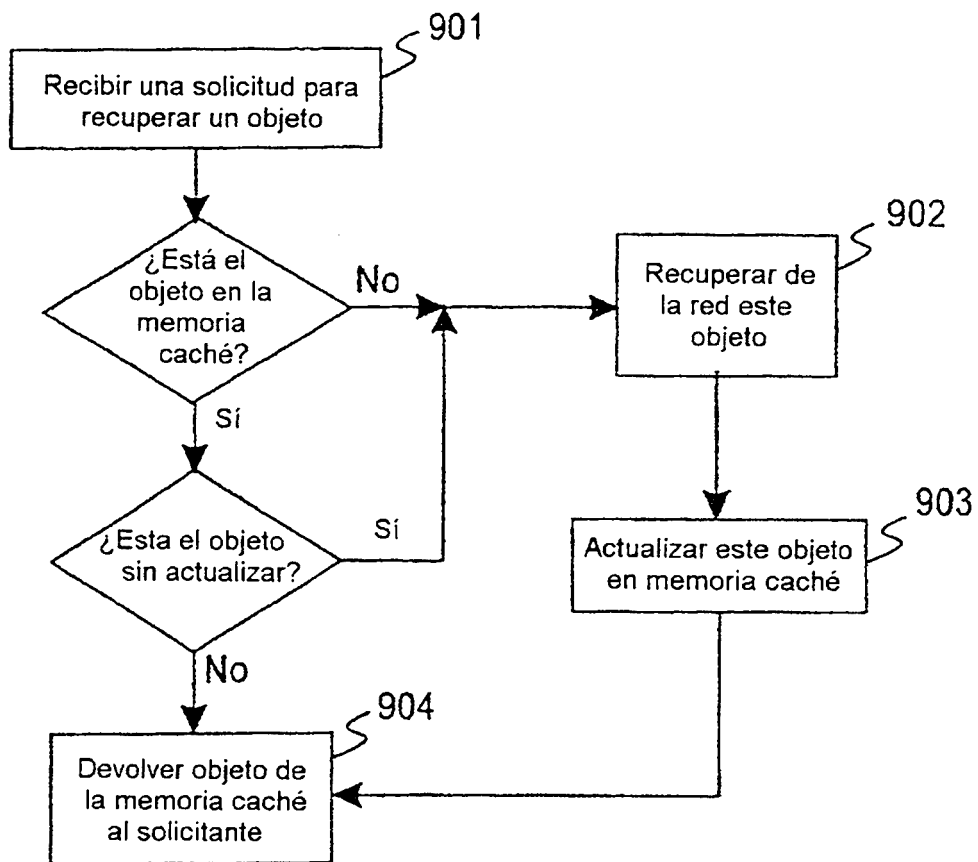


Fig.9

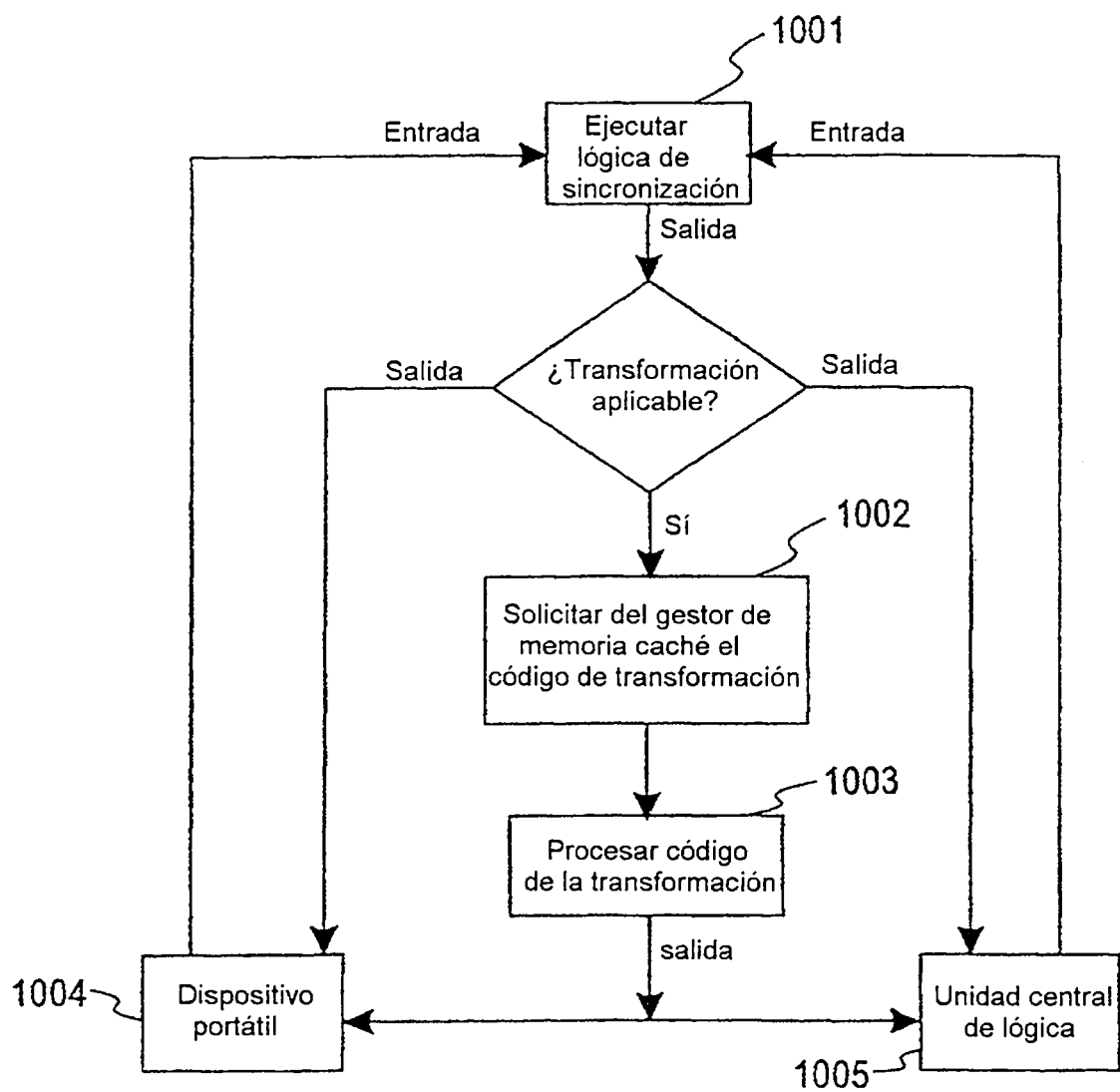


Fig.10

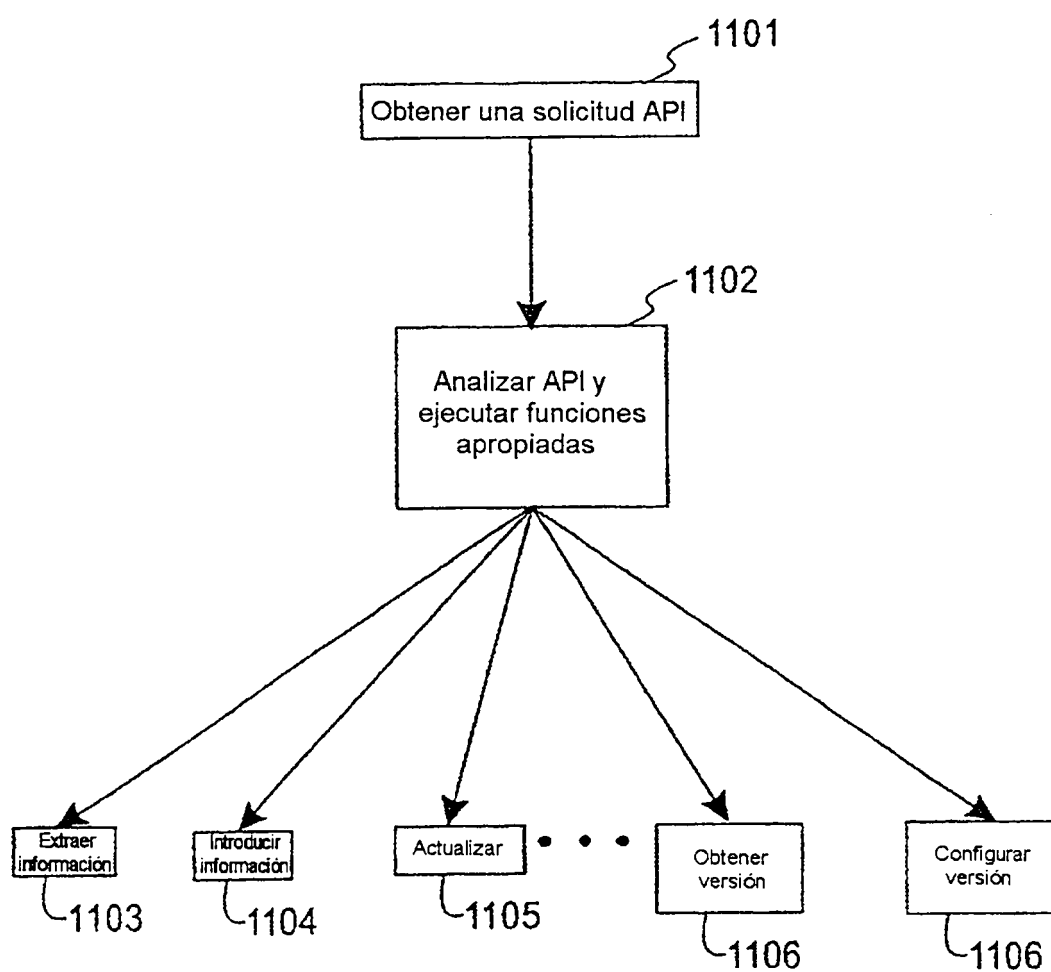


Fig.11

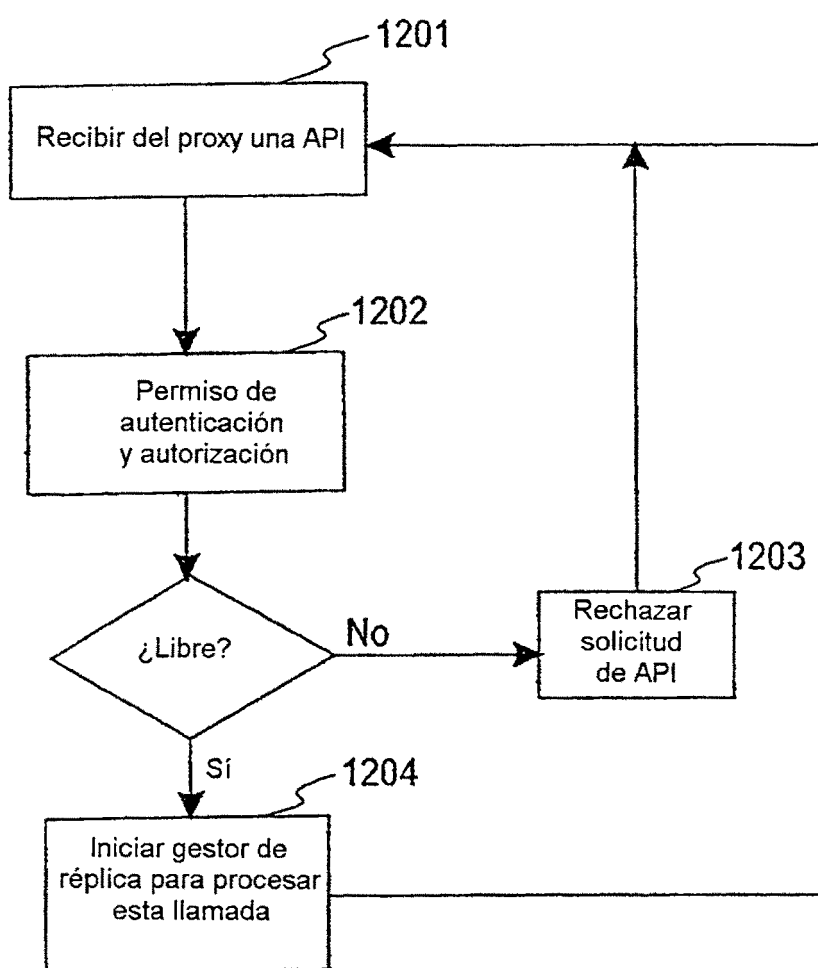


Fig.12

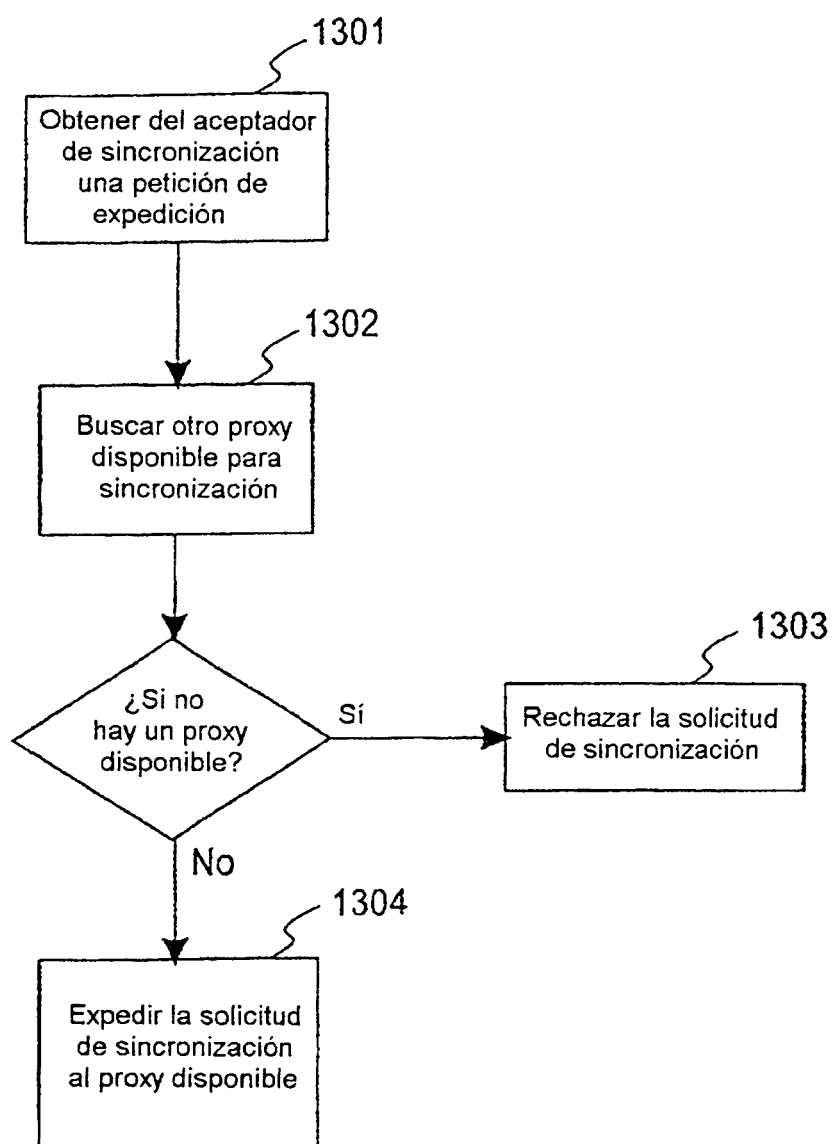


Fig.13