

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 710 873**

51 Int. Cl.:

G06F 8/61 (2008.01)

G06F 8/71 (2008.01)

H04L 12/24 (2006.01)

H04L 29/08 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **22.12.2011 PCT/US2011/066688**

87 Fecha y número de publicación internacional: **28.06.2012 WO12088364**

96 Fecha de presentación y número de la solicitud europea: **22.12.2011 E 11851335 (7)**

97 Fecha y número de publicación de la concesión europea: **21.11.2018 EP 2656211**

54 Título: **Cumplimentación de dependencias de aplicaciones**

30 Prioridad:

23.12.2010 US 977095

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

29.04.2019

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)**

**One Microsoft Way
Redmond, WA 98052, US**

72 Inventor/es:

**BABEY, MATTHEW CHRISTOPHER;
JEWART, ERIC;
REIERSON, KRISTOFER HELLICK y
ANDERSON, ANGELA MELE**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 710 873 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

Cumplimentación de dependencias de aplicaciones

Antecedentes

5 Los programas de aplicaciones informáticas ("aplicaciones") típicamente tienen una serie de dependencias que son requeridas para desplegar y ejecutar la aplicación con éxito. Estas dependencias pueden incluir acceso a recursos, tales como bases de datos, aplicaciones, componentes de hardware, conexiones de red y similares. Mientras el dispositivo esté operando localmente dentro de la red corporativa de una empresa, muchas de estas dependencias existen debido a que están conectadas a la red corporativa. Sin embargo, una vez que el dispositivo ya no está conectado a la red corporativa, por ejemplo, cuando el dispositivo abandona las instalaciones de la corporación, es posible que el dispositivo ya no esté conectado a la red corporativa. Como resultado, muchas de las dependencias de la aplicación que estaban presentes en virtud de que el dispositivo estaba conectado a la red corporativa, ya no están presentes. En consecuencia, la aplicación no se puede desplegar o ejecutar en el dispositivo hasta que se restauren las dependencias.

15 Un mecanismo para restaurar las dependencias de la aplicación requiere que se establezca una conexión con la red corporativa. Por ejemplo, un usuario corporativo que trabaja desde su hogar puede necesitar conectarse a una red corporativa a través de una red privada virtual (VPN) con el fin de restablecer las dependencias. Sin embargo, utilizar una conexión de esta manera puede ser engorroso o puede exponer una red corporativa a amenazas de seguridad.

20 Es con respecto a estas y otras consideraciones que se presenta la divulgación realizada por la presente memoria descriptiva.

25 El documento US 2010/064284 A1 revela un procedimiento para satisfacer las dependencias del cargador en un dispositivo informático integrado en ejecución. Cuando una aplicación de software se carga en un dispositivo informático integrado, se identifican uno o más archivos binarios que deben ser cargados para ejecutar la aplicación de software. Se realiza una determinación respecto a si los archivos binarios identificados están almacenados en el dispositivo informático integrado. Si uno o más de los archivos binarios identificados faltan en el dispositivo informático integrado, se identifican uno o más dispositivos de almacenamiento de archivos que contienen los archivos binarios que faltan. Los archivos binarios que faltan se obtienen de uno o más dispositivos de almacenamiento de archivos durante el tiempo de ejecución y se utilizan para completar la carga de la aplicación de software.

30 El documento US 2008/072219 A1 revela un procedimiento y un sistema para desplegar componentes de software en un entorno de tiempo de ejecución. El procedimiento extrae metadatos a una memoria que es independiente de un sistema de archivos y verifica si los componentes pueden ser desplegados en base a los metadatos. El procedimiento despliega adicionalmente los componentes después de que se complete la verificación. Sobre la base de los metadatos extraídos, un procesador de archivos crea objetos de despliegue. Para desplegar los objetos desplegables, el procesador de archivos comprueba la compatibilidad de los objetos con la versión de la plataforma del entorno de tiempo de ejecución, con la versión de los objetos ya desplegados en el entorno y con las dependencias entre los objetos.

40 El documento US 2007/180433 A1 revela una generación de una biblioteca de dependencias personalizada para su uso por una aplicación. Se monitoriza la ejecución de la aplicación. Cada dependencia utilizada por la aplicación está identificada durante la ejecución de la aplicación. Se genera una biblioteca de dependencias personalizadas para la aplicación que contiene solo las dependencias utilizadas por la aplicación. Se monitoriza la ejecución de la aplicación. Cada clase utilizada por la aplicación es identificada durante la ejecución de la aplicación. Se genera una biblioteca específica para la aplicación que contiene solo las clases utilizadas por la aplicación.

El objeto de la presente invención es proporcionar un procedimiento y un sistema mejorados para satisfacer las dependencias de la aplicación.

45 Este objeto se consigue mediante el objeto de las reivindicaciones independientes.

Las realizaciones están definidas por las reivindicaciones dependientes.

Sumario

50 Se describen tecnologías en la presente memoria descriptiva para satisfacer las dependencias de las aplicaciones virtualizadas. A través de una implementación de los conceptos y tecnologías presentados en la presente memoria descriptiva, las dependencias de la aplicación se pueden identificar y restaurar en el tiempo de ejecución de la aplicación. En consecuencia, un dispositivo cliente puede desplegar y ejecutar con éxito una aplicación sin tener que establecer una conexión en segundo plano entre el dispositivo y la red corporativa. Como se usa en la presente memoria descriptiva, el término "dependencias" se refiere a cualquier recurso que una aplicación requiera para su

ejecución. Las dependencias pueden incluir, pero sin estar limitadas a, valores de configuración, componentes de hardware, conexiones de red, privilegios de acceso o cualquier otro tipo de recurso.

5 De acuerdo con un aspecto proporcionado en la presente memoria descriptiva, se proporciona un entorno de virtualización de aplicaciones que es capaz de ejecutar una aplicación virtualizada configurando un dispositivo cliente de tal manera que todas las dependencias de la aplicación virtualizada estén presentes. Se debe apreciar que el ámbito de la presente divulgación no pretende estar limitado al uso dentro de un entorno de virtualización de aplicaciones. Por el contrario, las realizaciones que se describen en la presente memoria descriptiva pueden utilizarse para satisfacer las dependencias de cualquier tipo de aplicación que se ejecute en un entorno virtualizado o no virtualizado.

10 De acuerdo con otro aspecto, cuando se recibe una solicitud para desplegar una aplicación virtualizada, un componente de cliente de virtualización de aplicaciones determina si las dependencias necesarias para desplegar la aplicación virtualizada están presentes. Si el componente de cliente de virtualización de aplicaciones determina que las dependencias no están presentes, el componente de cliente de virtualización de aplicaciones configura el dispositivo cliente de tal manera que las dependencias estén presentes. Si el componente de cliente de virtualización de aplicaciones determina que las dependencias están presentes, el componente de cliente de virtualización de aplicaciones despliega la aplicación virtualizada para su ejecución.

15 De acuerdo con otro aspecto, el componente de cliente de virtualización de aplicaciones también está configurado para proporcionar un límite de seguridad entre una aplicación virtualizada y una aplicación no virtualizada, de modo que la aplicación no virtualizada no pueda acceder a ciertos recursos. Por ejemplo, en una realización, el componente de cliente de virtualización de aplicaciones intercepta las llamadas a la interfaz de programación de aplicaciones (API) realizadas por la aplicación virtualizada para acceder a recursos, tales como solicitudes para acceder a recursos disponibles para el dispositivo cliente. El componente de cliente de virtualización de aplicaciones a continuación modifica entonces un espacio de nombres del recurso solicitado de manera que el recurso sea accesible en una nueva ubicación. En algunas realizaciones, el componente cliente de virtualización de aplicaciones también puede modificar una lista de control de acceso (ACL) asociada con el recurso y pasar la llamada a la API al recurso almacenado en la nueva ubicación. Las solicitudes para acceder al recurso desde aplicaciones no virtualizadas no pueden acceder al recurso en su nueva ubicación.

20 De acuerdo con otro aspecto, las dependencias de la aplicación se identifican antes del tiempo de ejecución al realizar un análisis estático en la aplicación. En una implementación, la aplicación virtualizada se instala en una estación de monitorización que proporciona un entorno en el que todas las dependencias de la aplicación virtualizada están presentes. Un motor de análisis de dependencias de la estación de monitorización monitoriza los cambios realizados en la estación de monitorización durante la instalación y ejecución de la aplicación virtualizada para identificar las dependencias. Al monitorizar la aplicación virtualizada, el motor de análisis de dependencias almacena los datos que identifican las dependencias en una base de datos de análisis de dependencias. La base de datos de análisis de dependencias se puede utilizar en el tiempo de ejecución de la aplicación para garantizar que todas las dependencias requeridas estén presentes.

Este Sumario no tiene la intención de identificar características clave o características esenciales de la materia reivindicada, ni se pretende que este Sumario se utilice para limitar el alcance de la materia reivindicada. Además, el objeto reivindicado no está limitado a las implementaciones que resuelven cualquiera o todas las desventajas señaladas en cualquier parte de esta divulgación.

40 **Breve descripción de los dibujos**

La figura 1 es un diagrama de arquitectura de red y de software que muestra un entorno operativo ilustrativo para las realizaciones que se describen en la presente memoria descriptiva;

la figura 2 es un diagrama esquemático que muestra un proceso para identificar las dependencias de una aplicación virtualizada, de acuerdo con una realización presentada en la presente memoria descriptiva;

45 la figura 3 es un diagrama de flujo que muestra aspectos de un proceso ilustrativo descrito en la presente memoria descriptiva para identificar las dependencias de una aplicación virtualizada en un modo de operación de aprendizaje, de acuerdo con una realización presentada en la presente memoria descriptiva;

la figura 4 es un diagrama de flujo que muestra aspectos de un proceso ilustrativo descrito en la presente memoria descriptiva para satisfacer las dependencias de una aplicación virtualizada, de acuerdo con una realización presentada en la presente memoria descriptiva;

50 la figura 5 es un diagrama esquemático que muestra un proceso para proporcionar un límite de seguridad para evitar que las aplicaciones externas no virtualizadas accedan a los recursos, de acuerdo con una realización presentada en la presente memoria descriptiva;

la figura 6 es un diagrama de flujo que muestra aspectos de un proceso ilustrativo descrito en la presente memoria descriptiva para proporcionar un límite de seguridad unidireccional para evitar que aplicaciones externas interactúen con recursos, de acuerdo con una realización presentada en la presente memoria descriptiva; y

- 5 la figura 7 es un diagrama de arquitectura de ordenador que muestra una arquitectura de software y hardware de ordenador ilustrativa para un sistema informático capaz de implementar las diversas realizaciones presentadas en la presente memoria descriptiva.

Descripción detallada

10 La siguiente descripción detallada está dirigida a las tecnologías para satisfacer las dependencias de las aplicaciones. Como se ha explicado brevemente más arriba, un dispositivo cliente configurado de la manera que se describe en la presente memoria descriptiva puede identificar las dependencias de la aplicación y hacer que esas dependencias estén disponibles en el tiempo de ejecución de la aplicación. De esta manera, el dispositivo cliente puede ser configurado para la ejecución de la aplicación sin requerir una acción manual para satisfacer las dependencias, como la conexión a una VPN.

15 Aunque el objeto que se describe en la presente memoria descriptiva se presenta en el contexto general de los módulos de programa que se ejecutan junto con la ejecución de un sistema operativo y programas de aplicación en un sistema informático, los expertos en la técnica reconocerán que otras implementaciones pueden realizarse en combinación con otros tipos de módulos de programa. En general, los módulos de programa incluyen rutinas, programas, componentes, estructuras de datos y otros tipos de estructuras que realicen tareas particulares o implementen tipos de datos abstractos particulares. Además, los expertos en la materia apreciarán que la materia que se describe en la presente memoria descriptiva se puede practicar con otras configuraciones de sistemas informáticos, incluidos dispositivos portátiles, sistemas multiprocesador, electrónica de consumo programable o basada en microprocesador, miniordenadores, ordenadores centrales y otros similares.

20 En la descripción detallada que sigue, se hacen referencias a los dibujos que se acompañan que forman parte de la misma, y que se muestran a modo de ilustración de realizaciones o ejemplos específicos. Haciendo referencia a continuación a los dibujos, en los que los números similares representan elementos similares a lo largo de varias figuras, se describirán los aspectos de un sistema informático y la metodología para satisfacer las dependencias de la aplicación.

25 La figura 1 es un diagrama de arquitectura de red y software que muestra un entorno operativo ilustrativo 100 para las realizaciones que se describen en la presente memoria descriptiva. El entorno operativo ilustrativo 100 que se muestra en la figura 1 incluye un dispositivo cliente 102 configurado para comunicarse con un servidor 104 a través de la red 106. El dispositivo cliente 102 es un dispositivo informático configurado para ejecutar un sistema operativo 108A y un componente cliente de virtualización de aplicaciones 110. El dispositivo cliente 102 puede ser un ordenador de escritorio o portátil, una tableta, un teléfono inteligente o cualquier otro tipo de dispositivo informático capaz de realizar las operaciones presentadas en la presente memoria descriptiva para satisfacer las dependencias de las aplicaciones virtualizadas. El dispositivo cliente 102 también podría ser un ordenador servidor configurado para proporcionar la funcionalidad que se describe en la presente memoria descriptiva.

30 El servidor 104 es un sistema informático configurado para ejecutar un sistema operativo 108B y el componente del servidor de virtualización de aplicaciones 120. Se debe apreciar que el servidor 104 puede ser un ordenador servidor real configurado para ejecutar el componente del servidor de virtualización de aplicaciones 110 o puede comprender otro tipo del sistema informático configurado para realizar la funcionalidad que se describe en la presente memoria descriptiva como realizada por el servidor 104.

35 La red 106 ilustrada en la figura 1 puede comprender una red de área amplia o una red de área local. Por ejemplo, la red 106 puede ser una red de área local corporativa, una red de área amplia tal como Internet, o una combinación de múltiples redes de área amplia y área local. Se debe apreciar que si bien se ha ilustrado una sola red 106 en la figura 1, muchas otras redes pueden ser utilizadas. También debería apreciarse que mientras que un único dispositivo cliente 102 y servidor 104 se han ilustrado en la figura 1, muchos de dispositivos de este tipo pueden ser utilizados por las realizaciones que se describen en la presente memoria descriptiva.

40 Como se ha explicado brevemente más arriba, el dispositivo cliente 102 está configurado para ejecutar un componente de cliente de virtualización de aplicaciones 110. El componente de cliente de virtualización de aplicaciones 110 es un componente de software configurado para proporcionar un entorno de virtualización de aplicaciones. En este sentido, el componente de cliente de virtualización de aplicaciones 110 está configurado para desplegar y ejecutar una aplicación virtualizada 112.

45 El componente de cliente de virtualización de aplicaciones 110 proporciona funcionalidad para encapsular la ejecución de la aplicación virtualizada 112 del sistema operativo 108A. El componente de cliente de virtualización de apli-

caciones 110 también podría proporcionar funcionalidad para encapsular la ejecución de la aplicación virtualizada 112 de otros programas de aplicación y recursos del sistema del dispositivo cliente 102. Por ejemplo, el componente de cliente de virtualización de aplicaciones 110 podría virtualizar los recursos del sistema operativo 108A o del dispositivo cliente 102. Cuando la aplicación virtualizada 112 intenta acceder a los recursos físicos, el componente de cliente de virtualización de aplicaciones 110 presenta un recurso virtualizado a la aplicación 112. De esta manera, la aplicación virtualizada 112 se puede ejecutar de una manera que no afecte a los recursos reales expuestos por el sistema operativo 108A o por el dispositivo cliente 102.

De acuerdo con otros aspectos, el componente de cliente de virtualización de aplicaciones 110 también proporciona funcionalidad para cargar porciones de la aplicación virtualizada 112 bajo demanda. En particular, el componente de cliente de virtualización de aplicaciones 110 puede operar en conjunto con el componente del servidor de virtualización de aplicaciones 120 para transmitir en continuo porciones necesarias de la aplicación virtualizada 112 desde el servidor 104 al dispositivo cliente 102. De esta manera, se puede acceder a la aplicación virtualizada 112. en el dispositivo cliente 102 bajo demanda. Además, debido a que solo las partes necesarias de la aplicación virtualizada 112 pueden ser transmitidas en continuo desde el servidor 104 al dispositivo cliente 102, se puede proporcionar acceso a la aplicación virtualizada 112 sin transmitir en continuo la aplicación completa 112 desde el servidor 104 al dispositivo cliente 102.

Detalles adicionales con respecto a las funciones proporcionadas por el componente de cliente de virtualización de aplicaciones 110 para encapsular la ejecución de la aplicación virtualizada 112 y para transmitir la aplicación virtualizada 112 desde el servidor 104 al dispositivo cliente 102 se pueden encontrar en la patente de EE. UU número 7.225.264 presentada el 29 de mayo de 2007, titulada "Sistemas y procedimientos para entregar contenido a través de una red de ordenadores", la patente de EE. UU número 7.200.632 presentada el 3 de abril de 2007, titulada "Procedimiento y sistema para servir aplicaciones de software a ordenadores cliente", la patente de EE. UU número 7.451.451 presentada el 11 de noviembre de 2008, titulada "Capa de protección y abstracción del sistema operativo", y la patente de EE. UU número 7.797.372 presentada el 14 de septiembre de 2010 titulada "Servir aplicaciones de software desde servidores ordenadores cliente".

Como se describirá con mayor detalle en la presente memoria descriptiva, el componente de cliente de virtualización de aplicaciones 110 también puede configurarse para garantizar que una o más dependencias 114 de la aplicación virtualizada 112 se cumplan en un tiempo de ejecución de la aplicación 112. Tal como se usa en la presente memoria descriptiva, el término "dependencias" se refiere a cualquier recurso que la aplicación 112 requiera para su ejecución. Las dependencias pueden incluir, pero sin estar limitadas a disposiciones de configuración, componentes de hardware, conexiones de red, privilegios de acceso, la presencia de archivos específicos en ubicaciones específicas, recursos de la aplicación tales como la instalación de un software antivirus, componentes de hardware tales como un monitor u otro dispositivo, requisitos de memoria, capacidades de procesamiento y otros.

Como se explicará con mayor detalle más adelante, las dependencias 114 pueden ser identificadas durante la instalación de la aplicación 112, antes del tiempo de ejecución de la aplicación 112, o en el tiempo de ejecución de la aplicación 112. Por ejemplo, cuando se recibe una solicitud para ejecutar la aplicación virtualizada 112, el componente de cliente de virtualización de aplicaciones 110 determina si están presentes los requisitos de las dependencias 114 para desplegar y / o ejecutar la aplicación virtualizada 112. Si el componente de cliente de virtualización de aplicaciones 110 determina que las dependencias 114 no están presentes, el componente de cliente de virtualización de aplicaciones 110 configura el dispositivo cliente 102 de tal manera que todas las dependencias 114 estén presentes. Una vez que todas las dependencias 114 están presentes o satisfechas, la aplicación virtualizada 112 puede desplegarse. Tal como se utiliza en la presente memoria descriptiva, el término "despliegue" significa hacer que la aplicación 112 se encuentre disponible para su ejecución. A continuación se proporcionan detalles adicionales con respecto a la configuración del dispositivo cliente 102, de manera que todas las dependencias 114 están presentes y están proporcionadas con respecto a las figuras 2 - 4.

Se debe apreciar que el dispositivo cliente 102 también puede incluir una aplicación externa no virtualizada 118 que puede ejecutarse en el sistema operativo 108A. Como se describirá con mayor detalle más adelante, el componente de cliente de virtualización de aplicaciones 110 proporciona funcionalidad en una realización para garantizar que la aplicación externa 118 no pueda acceder a los recursos que utiliza la aplicación virtualizada 112. Por ejemplo, el componente de cliente de virtualización de aplicaciones 110 podría redirigir una solicitud de acceso a un recurso a una ubicación en la que el recurso no existe. De esta manera, el sistema operativo puede responder a la solicitud indicando que el recurso solicitado no existe o no se puede ubicar. Los detalles adicionales con respecto a este proceso se proporcionarán a continuación con respecto a las figuras 5 - 6.

Como se ha descrito más arriba, el servidor 104 podría incluir la aplicación virtualizada 112, el componente del servidor de virtualización de aplicaciones 110 y el sistema operativo 108B. Además, el servidor 104 puede incluir un motor de análisis de dependencias 122 y una base de datos de análisis de dependencias 124 para almacenar los datos de dependencias 126 que identifican las dependencias 114. Se debe apreciar que la base de datos de análisis de dependencias 124 que contiene datos de dependencias 126 también se puede almacenar y / o hacerse accesible al componente de cliente de virtualización de aplicaciones 110 del dispositivo cliente 102. De acuerdo con las reali-

zaciones, el servidor 104 puede operar como una estación de monitorización que está configurada para instalar y ejecutar la aplicación virtualizada 112 y para identificar las dependencias 114 de la aplicación virtualizada 112. De acuerdo con diversas realizaciones, el motor de análisis de dependencias 122 puede identificar las dependencias 114 de la aplicación virtualizada 112 y almacenar los datos de dependencias 126 que identifican las dependencias 114 en la base de datos de análisis de dependencias 124. También se puede proporcionar una interfaz de usuario 130 para ver y editar el contenido de la base de datos 124. Detalles adicionales sobre diversos procesos proporcionados en la presente memoria descriptiva para identificar las dependencias 114 se proporcionarán a continuación con respecto a las figuras 2 y 3.

Se debe apreciar que las realizaciones que se describen en la presente memoria descriptiva son meramente con fines ilustrativos y no pretenden limitar el ámbito de la aplicación a las realizaciones que se describen en la presente memoria descriptiva. Los expertos en la materia deberían apreciar que las tecnologías que se describen en la presente memoria descriptiva pueden aplicarse junto con una variedad de tipos de aplicaciones y que las realizaciones que se describen en la presente memoria descriptiva no se limitan a satisfacer las dependencias de una aplicación virtualizada 112. También se debe entender que los conceptos descritos en la presente memoria descriptiva pueden aplicarse a cualquier tipo de aplicación, incluida pero no limitada a las aplicaciones virtualizadas y no virtualizadas que se ejecutan en entornos virtualizados y no virtualizados.

La figura 2 es un diagrama esquemático que muestra un proceso para identificar las dependencias de la aplicación virtualizada 112 mientras la aplicación virtualizada 112 se está ejecutando en un entorno, de acuerdo con una realización presentada en la presente memoria descriptiva. En particular, la figura 2 ilustra aspectos de un modo de aprendizaje de operación utilizado para identificar las dependencias 114. En el modo de aprendizaje de operación, un interceptor de llamadas a la API 202 intercepta las llamadas a la API realizadas por la aplicación 112 para acceder a los recursos. Como se explicará más adelante, las llamadas a la API pueden realizarse durante el tiempo de ejecución de la aplicación 112 o en el momento en que se instala la aplicación 112.

Cuando la aplicación virtualizada 112 está instalada en un dispositivo, como el dispositivo cliente 102, el instalador de la aplicación 112 podría realizar cambios en el dispositivo 102. Por ejemplo, varios archivos pertenecientes a la aplicación virtualizada 112 pueden almacenarse o los archivos de registro pueden ser creados o modificados. El motor de análisis de dependencias 122 monitoriza la instalación de la aplicación 112 para detectar cualquier cambio de configuración realizado como resultado de la instalación de la aplicación virtualizada 112. El motor de análisis de dependencias 122 también puede comparar la configuración del ordenador en el que se está ejecutando antes de que se instale la aplicación virtualizada 112 con la configuración después de que se instale la aplicación virtualizada 112. El motor de análisis de dependencias 122 puede utilizar los cambios detectados para identificar las dependencias 114. Los datos de dependencias 126 que identifican las dependencias 114 se almacenan en la base de datos de análisis de dependencias 124.

Como parte del modo de operación de aprendizaje, el motor de análisis de dependencias 122 puede monitorizar aún más la aplicación virtualizada 112 durante el tiempo de ejecución. A medida que se ejecuta la aplicación virtualizada 112, la aplicación virtualizada 112 puede realizar llamadas a la API para acceder a los recursos. El interceptor de llamadas a la API 202 enlaza las llamadas a la API realizadas por la aplicación virtualizada 112 y pasa las llamadas al motor de análisis de dependencias 122, en el que se inspeccionan para recopilar información sobre las dependencias adicionales 114 de la aplicación virtualizada 112. Las llamadas a la API pueden incluir solicitudes para acceder a recursos particulares, tales como archivos, bases de datos y similares. Además, las llamadas a la API pueden incluir solicitudes para acceder a aplicaciones particulares.

El motor de análisis de dependencias 122 inspecciona las llamadas a la API interceptadas para identificar las dependencias de tiempo de ejecución 114 de la aplicación virtualizada 112. El motor de análisis de dependencias 122 puede almacenar datos que identifican las dependencias de tiempo de ejecución 114 en la base de datos de análisis de dependencias 124. Además, los datos pueden almacenarse en la base de datos 124 que indica si las dependencias identificadas 114 son proporcionadas por un sistema operativo 108. En realizaciones alternativas, los datos que identifican las dependencias en tiempo de ejecución 114 pueden transmitirse desde el servidor 104 al dispositivo cliente 102. En tales realizaciones, los datos que identifican las dependencias pueden empaquetarse en un paquete de aplicaciones que puede configurarse para ser transmitido al dispositivo cliente 102. Como se describirá con mayor detalle a continuación, los datos generados durante el modo de aprendizaje de operación se utilizan en el tiempo de ejecución de la aplicación 112 para garantizar que todas las dependencias 114 estén disponibles a la aplicación 112. Por ejemplo, los datos de dependencias 126 almacenados en la base de datos 124 podrían utilizarse para determinar si la ejecución de una aplicación depende de cualquier recurso que no sea proporcionado por un sistema operativo. Si es así, los recursos podrían estar disponibles para la aplicación.

Se debe apreciar que no todos los recursos pueden estar disponibles para la aplicación. En tales situaciones, es posible que la aplicación no pueda desplegarse en el dispositivo hasta que los recursos estén disponibles y todas las dependencias de la aplicación estén debidamente cumplimentadas. Por ejemplo, una corporación puede tener documentos a los que solo se puede acceder desde un edificio corporativo. Cuando un usuario fuera del edificio corporativo ejecuta una aplicación que interactúa con esos documentos, el componente de cliente de virtualización de

aplicaciones 110 puede impedir que la aplicación acceda a los documentos. En este caso, la dependencia puede ser la presencia de una dirección IP corporativa que indique que el usuario está dentro del edificio corporativo. Si el usuario no está dentro del edificio corporativo, la aplicación virtualizada 112 puede no desplegarse ya que no se cumple la dependencia correspondiente a la presencia de una dirección IP corporativa.

5 La figura 3 es un diagrama de flujo que muestra aspectos de un proceso ilustrativo revelado en la presente memoria descriptiva para identificar las dependencias 114 de una aplicación virtualizada 112 en el modo de aprendizaje que se ha descrito más arriba. Se debe apreciar que las operaciones lógicas que se describen en la presente memoria descriptiva con respecto a las figuras 3, 4 y 6 se implementan (1) como una secuencia de actos implementados por ordenador o módulos de programa que se ejecutan en un sistema informático y / o (2) como circuitos lógicos de máquinas interconectados o módulos de circuito dentro del sistema informático. La implementación es una cuestión de elección que depende del rendimiento y de otros requisitos del sistema informático. En consecuencia, las operaciones lógicas que se describen en la presente memoria descriptiva se denominan de diversas maneras como operaciones, dispositivos estructurales, actos o módulos. Estas operaciones, dispositivos estructurales, actos y módulos pueden ser implementados en software, en firmware, en lógica digital para propósitos especiales y en cualquier combinación de los mismos. También se debe tener en cuenta que se pueden realizar más o menos operaciones de las que se muestran en las figuras y se describen en la presente memoria descriptiva. Estas operaciones también pueden realizarse en un orden diferente a los que se han descritos en la presente memoria descriptiva.

La figura 3 muestra una rutina 300 que ilustra las operaciones realizadas en el modo de operación de aprendizaje que se muestra en la figura 2 y que se ha descrito más arriba. La rutina 300 comienza en la operación 302, en la que la aplicación virtualizada 112 se instala en un sistema de monitorización, tal como el servidor 104. El servidor 104 incluye el motor de análisis de dependencias 122 que proporciona la funcionalidad que se ha descrito más arriba para identificar las dependencias 114 de la aplicación virtualizada 112. El servidor 104 también puede incluir todas las dependencias 114 requeridas para la instalación y ejecución adecuadas de la aplicación virtualizada 112. De esta manera, la aplicación virtualizada 112 puede operar en el servidor 104 en un modo de aprendizaje. Se debe apreciar que otro ordenador que no sea el servidor 104 podría ser utilizado como la estación de monitorización para realizar el proceso de aprendizaje que se ha descrito en la presente memoria descriptiva.

De acuerdo con una implementación, cuando el servidor 104 está funcionando en el modo de aprendizaje, el motor de análisis de dependencias 122 está configurado para monitorizar los cambios realizados en el servidor 104 que identifican las dependencias 114 de la aplicación virtualizada 112 y almacenan los datos de dependencias 126 que identifican las dependencias 114 en la base de datos de análisis de dependencias 124. Las dependencias 114 pueden incluir dependencias que son identificadas por el análisis de la instalación y la configuración de la aplicación 112, así como las dependencias identificadas por la interceptación de llamadas a la API realizadas por la aplicación virtualizada 112 en tiempo de ejecución.

Desde la operación 302, la rutina 300 procede a la operación 304, en la que el motor de análisis de dependencias 122 monitoriza los cambios realizados durante la instalación de la aplicación virtualizada 112. El motor de análisis de dependencias 122 puede monitorizar los cambios comparando los ajustes de configuración del sistema después de que la aplicación virtualizada 112 sea instalada en los ajustes de configuración del sistema antes de la instalación de la aplicación virtualizada 112. Además, el motor de análisis de dependencias 122 puede monitorizar varios recursos del sistema para identificar las dependencias 114, tales como los cambios realizados en un registro o la creación de archivos y carpetas adicionales relacionados con la instalación de la aplicación virtualizada 112.

Desde la operación 304, la rutina 300 procede a la operación 306, en la que se ejecuta la aplicación virtualizada 112. Durante la ejecución, el motor de análisis de dependencias 122 monitoriza continuamente la aplicación virtualizada 112 para las llamadas a la API realizadas por la aplicación virtualizada 112 para acceder a los recursos, así como cualquier cambio de configuración realizado como resultado de la ejecución de la aplicación virtualizada 112.

Desde la operación 306, la rutina 300 procede a la operación 308, en la que el interceptor 202 de llamadas a la API intercepta las llamadas a la API realizadas por la aplicación virtualizada 112. El interceptor 202 de llamadas a la API puede proporcionar estas llamadas a la API al motor de análisis de dependencias 122. Desde la operación 308, la rutina 300 procede a la operación 310, en la que el motor de análisis de dependencias 122 inspecciona los parámetros de las llamadas a la API interceptadas para identificar las dependencias 114 de la aplicación virtualizada 112. Como se ha descrito más arriba, las llamadas a la API pueden incluir solicitudes para acceder a un recurso en particular, tal como un archivo, una aplicación o un componente de hardware, tal como un altavoz o una pantalla. De esta manera, el motor de análisis de dependencias 122 es capaz de determinar las dependencias de tiempo de ejecución de la aplicación virtualizada 112 mediante la inspección del formato y el contenido de las llamadas a la API interceptadas.

Desde la operación 310, la rutina 300 procede a la operación 312, en la que el motor de análisis de dependencias 122 almacena los datos de dependencias 126 recopilados durante la instalación y la ejecución de la aplicación virtualizada 112 que identifica las dependencias 114 en la base de datos de análisis de dependencias 124. De acuerdo con una implementación, la base de datos de análisis de dependencias 124 se puede almacenar en el servidor 104

o en cualquier otra ubicación a la que pueda acceder el dispositivo cliente 102 o el servidor 104 a través de la red 106.

5 Desde de la operación 312, la rutina 300 procede a la operación 314, en la que se determina si cada una de las dependencias 114 de la aplicación virtualizada 112 se agrega o configura durante la instalación o despliegue de la aplicación virtualizada 112. Esto se puede lograr, por ejemplo, mediante el examen de un registro proporcionado por un sistema operativo. Alternativamente, esto se logra al examinar una tabla de asignación que asigna las llamadas a la API a las dependencias. La tabla de asignación puede incluir información que indique si cada dependencias es proporcionada por un sistema operativo.

10 A partir de la operación 314, la rutina 300 procede a la operación 316 si la dependencia se agrega o configura durante la instalación o despliegue de la aplicación. En la operación 316, los datos se almacenan en la base de datos de análisis de dependencias 124, lo que indica que las dependencias 114 que se agregan o configuran durante la instalación o despliegue de la aplicación virtualizada 112 deberán estar presentes en el dispositivo cliente 102 en tiempo de ejecución para desplegar con éxito la aplicación virtualizada 112 en el dispositivo cliente 102. Si en la operación 314 se determina que las dependencias 114 no se agregan ni configuran durante la instalación o despliegue de la aplicación virtualizada 112, la rutina 300 continúa con la operación 318, en la que la rutina 300 finaliza. De manera similar, a partir de la operación 316, la rutina 300 también procede a la operación 318, en la que la rutina 300 finaliza.

20 Una vez que las dependencias 114 de la aplicación virtualizada 112 se han identificado y los datos de dependencias 126 que identifican las dependencias 114 se almacenan en la base de datos de análisis de dependencias 124, la aplicación virtualizada 112 puede ser desplegada en el dispositivo cliente 102. Sin embargo, antes de desplegar la aplicación virtualizada 112 en el dispositivo cliente 102, las dependencias 114 requeridas para desplegar la aplicación virtualizada 112 deben ser accesibles por el dispositivo cliente 102. En una implementación, los datos de dependencias 126 que identifican las dependencias 114 se almacenan en el dispositivo cliente 102 y son utilizadas por el componente de cliente de virtualización de aplicaciones 110 para garantizar que las dependencias 114 de la aplicación virtualizada 112 estén presentes en el tiempo de ejecución del despliegue de la aplicación virtualizada 112 en el dispositivo cliente 102. Más adelante se proporcionan detalles adicionales en lo que se refiere a este proceso con respecto a la figura 4.

30 La figura 4 es un diagrama de flujo que muestra aspectos de un proceso ilustrativo que se ha revelado en la presente memoria descriptiva para satisfacer las dependencias de la aplicación virtualizada 112 en el dispositivo cliente 102, de acuerdo con una realización presentada en la presente memoria descriptiva. En particular, la figura 4 muestra una rutina 400 que ilustra las operaciones realizadas en el tiempo de ejecución de la aplicación 112. En particular, la rutina 400 comienza en la operación 402, en la que el componente de cliente de virtualización de aplicaciones 110 recibe una solicitud para desplegar la aplicación virtualizada 112 en el dispositivo cliente 102. De acuerdo con una implementación, un usuario del dispositivo cliente 102 puede enviar la solicitud para desplegar la aplicación virtualizada 112 en el dispositivo cliente 102.

40 Desde la operación 402, la rutina 400 procede a la operación 404, en la que el componente de cliente de virtualización de aplicaciones 110 determina las dependencias 114 de la aplicación virtualizada 112 al interrogar a la base de datos de análisis de dependencias 124. El componente de cliente de virtualización de aplicaciones 110 también puede interrogar a la base de datos de análisis de dependencias para determinar si las dependencias identificadas son proporcionadas por un sistema operativo 108. Si es así, las dependencias no necesitan ser restauradas en tiempo de ejecución por el componente de cliente de virtualización de aplicaciones 110.

45 Desde la operación 404, la rutina 400 procede a la operación 406, en la que el componente de cliente de virtualización de aplicaciones 110 determina si todas las dependencias 114 están presentes. De acuerdo con algunas implementaciones, el componente de cliente de virtualización de aplicaciones 110 puede hacerlo comparando las dependencias 114 determinadas por el motor de análisis de dependencias 122 durante el modo de aprendizaje con las dependencias presentes en el dispositivo cliente 102. Si el componente de cliente de virtualización de aplicaciones 110 determina que todas las dependencias 114 están presentes, la rutina 400 procede a la operación 414, en la que el componente de cliente de virtualización de aplicaciones 110 implementa la aplicación virtualizada 112 en el dispositivo cliente 102.

50 Sin embargo, si en la operación 406 el componente de cliente de virtualización de aplicaciones 110 determina que todas las dependencias 114 no están presentes en el dispositivo cliente 102, la rutina 400 procede desde la operación 406 a la operación 408, en la que el componente de cliente de virtualización de aplicaciones 110 identifica una dependencia que falta. Una dependencia que falta puede ser un recurso que no está presente o no está configurado correctamente. Por ejemplo, si una de las dependencias 114 incluye activar un servidor de seguridad en el dispositivo cliente 102, el componente de cliente de virtualización de aplicaciones 110 puede verificar el dispositivo cliente 102 para ver si el servidor de seguridad está activado. Si el cortafuegos no está activado, entonces el componente de cliente de virtualización de aplicaciones 110 identifica el cortafuegos como una dependencias que falta.

Desde de la operación 408, la rutina 400 procede a la operación 410, en la que el componente de cliente de virtualización de aplicaciones 110 configura el dispositivo cliente 102 de tal manera que se satisfacen las dependencias que falta. El componente de cliente de virtualización de aplicaciones 110 puede hacerlo configurando el dispositivo cliente 102 de tal manera que los ajustes de configuración del dispositivo cliente 102 coincidan con los ajustes de configuración del servidor 104 cuando la aplicación virtualizada 112 se instaló y se ejecutó en el servidor 104. Usando el ejemplo que se ha descrito más arriba, el componente de cliente de virtualización de aplicaciones 110 puede activar automáticamente el cortafuegos al cambiar la configuración del cortafuegos del dispositivo cliente 102. En algunas realizaciones, el componente de cliente de virtualización de aplicaciones 110 también puede hacer que un recurso esté disponible. De esta manera, el componente de cliente de virtualización de aplicaciones 110 puede configurar el dispositivo cliente 102 de tal manera que se cumplan los requisitos de dependencias 114 para desplegar y / o ejecutar la aplicación.

Desde la operación 410, la rutina 400 procede a la operación 412, en la que el componente de cliente de virtualización de aplicaciones 110 determina si faltan más dependencias 114. Si el componente de cliente de virtualización de aplicaciones 110 determina que faltan más dependencias 114, la rutina 400 procede retornando a la operación 406, en la que la rutina 400 identifica otras dependencias que faltan. Sin embargo, si el componente de cliente de virtualización de aplicaciones 110 determina que no faltan más dependencias 114, la rutina 400 procede a la operación 414, en la que el componente de cliente de virtualización de aplicaciones 110 despliega la aplicación virtualizada 112. Desde la operación 414, la rutina 400 procede a la operación 416, en la que finaliza la rutina 400. Se debe apreciar que el componente de cliente de virtualización de aplicaciones 110 puede no ser capaz de satisfacer todas las dependencias requeridas para desplegar la aplicación virtualizada 112. Por ejemplo, en el ejemplo que se ha descrito más arriba, el componente de cliente de virtualización de aplicaciones 110 puede ser incapaz de satisfacer una dependencia, tal como la presencia de una dirección IP corporativa, que se proporciona solo a los dispositivos que operan dentro del edificio corporativo.

La figura 5 es un diagrama esquemático que muestra un proceso para proporcionar un límite de seguridad para evitar que las aplicaciones externas accedan a los recursos disponibles a la aplicación 112. De acuerdo con algunas implementaciones, las aplicaciones no empresariales instaladas en el dispositivo cliente 102, tal como la aplicación externa 118, pueden intentar obtener acceso a los recursos utilizados por la aplicación virtualizada 112. Esto puede representar una amenaza para la seguridad y, por lo tanto, se pueden hacer esfuerzos para evitar que la aplicación externa 118 interactúe con los recursos. En una implementación, el dispositivo cliente 102 proporciona funcionalidad para redireccionar las llamadas a la API realizadas por aplicaciones externas para evitar que la aplicación externa 118 acceda a los recursos. En algunas implementaciones, este redireccionamiento se puede producir modificando los nombres de los recursos (a lo que se puede hacer referencia en la presente memoria descriptiva como "manipulación del espacio de nombres"). La figura 5 ilustra un proceso proporcionado en la presente memoria descriptiva para realizar esta funcionalidad.

Cuando una aplicación virtualizada 112 realiza una llamada a la API para acceder o modificar un recurso externo, una capa de enrutamiento 502 intercepta la llamada a la API y realiza dos funciones. En primer lugar, la capa de enrutamiento 502 hace que el nombre del recurso original 504 se modifique, creando así un recurso renombrado 506. Como resultado, la aplicación externa 118 que se ejecuta fuera del componente de cliente de virtualización de aplicaciones 110 no puede buscar el recurso en el lugar esperado. De esta manera, las aplicaciones curiosas son disuadidas de intentar acceder al recurso. Sin embargo, una aplicación externa determinada 118 aún puede abrir y acceder al recurso original 504. Para evitar que alguna aplicación externa determinada acceda al recurso, el componente de cliente de virtualización de aplicaciones 110 también puede modificar una lista de control de acceso (ACL) del recurso original 504 para una ACL personalizada de tal manera que la aplicación externa 118 no pueda abrir o acceder al recurso 504. La figura 6 ilustra aspectos adicionales de este proceso.

Se debe apreciar que el componente de cliente de virtualización de aplicaciones 110 puede tener múltiples copias de un recurso y puede redirigir diferentes procesos para ver diferentes versiones del recurso. Por ejemplo, el dispositivo cliente 102 podría tener dos copias de un valor de registro llamado "foo", en el que una copia se establece en 0 y la otra en 1. Cuando uno de los procesos accede al valor del registro, el dispositivo cliente 102 redirige el proceso de manera transparente a la copia del recurso que se establece en 0. Cuando el otro proceso accede a lo que cree que es el mismo valor de registro, el dispositivo cliente 102 redirige de manera transparente este proceso a la copia del recurso que se establece en 1. Ambos procesos están tratando de acceder al mismo recurso pero el dispositivo cliente 102 los está redirigiendo a instancias de ese recurso con nombres diferentes sin que ellos sean conscientes de esto. De esta manera, los dos procesos pueden realizar dos operaciones separadas ya que los dos procesos están recibiendo diferentes valores de entrada. Como resultado, el dispositivo cliente 102 es capaz de proporcionar a las aplicaciones autorizadas los valores de entrada correctos, mientras que proporciona a aplicaciones no autorizadas valores de entrada incorrectos que pueden proporcionar seguridad al dispositivo cliente 102.

En otro ejemplo, cuando dos procesos intentan acceder al mismo archivo, el dispositivo cliente 102 podría redirigir uno de ellos a una copia válida del archivo, y podría informar al otro de que el archivo no existía, o alternativamente, redirigirlo a una ubicación en la que no existía ningún archivo y permitir que el sistema operativo responda. Como resultado, el proceso no autorizado puede no tener acceso al archivo.

La figura 6 es un diagrama de flujo que muestra aspectos de un proceso ilustrativo que se ha descrito en la presente memoria descriptiva para proporcionar un límite de seguridad. En particular, la figura 6 muestra una rutina 600 que ilustra operaciones realizadas en una realización para proporcionar un límite de seguridad de la manera que se ha descrito más arriba con referencia a la figura 5.

5 La rutina 600 comienza en la operación 602, en la que la capa de enrutamiento 502 intercepta una llamada a la API realizada por la aplicación virtualizada 112 para acceder a un recurso. Como se ha descrito más arriba, la capa de enrutamiento 502 solo puede interceptar las llamadas a la API realizadas por la aplicación virtualizada 112. Desde la operación 602, la rutina 600 procede a la operación 604, en la que la capa de enrutamiento 502 selecciona una copia del recurso al que la llamada a la API realizada por la aplicación virtualizada 112 será redirigida. La copia del recurso puede haber sido guardada previamente por el sistema operativo. A partir de la operación 604, la rutina 600
10 procede a la operación 606, en la que el sistema operativo 108A reconfigura aún más una ACL asociada con el recurso, de manera que el recurso no sea accesible para las aplicaciones externas que operan fuera del componente de cliente de virtualización de aplicaciones 110. Desde la operación 606, la rutina 600 procede a la operación 608, en la que la llamada a la API interceptada se enruta al recurso 506 cuyo nombre ha sido renombrado.

15 Desde la operación 608, la rutina 600 procede a la operación 610, en la que el sistema operativo 108A recibe una llamada a la API que solicita acceso al recurso desde la aplicación externa 118. De acuerdo con las realizaciones, puesto que la aplicación externa 118 se comunica directamente con el sistema operativo 108A sin ir a través del componente de cliente de virtualización de aplicaciones 110, el sistema operativo 108A recibe la llamada a la API desde la aplicación externa 118. Sin embargo, se debe apreciar que la aplicación externa puede enviar llamadas a la API a través del componente de cliente de virtualización de aplicaciones 110. En tales realizaciones, el componente de cliente de virtualización de aplicaciones 110 no realiza ninguna de las redirecciones, sino que pasa la llamada a la API original al sistema operativo 108A para que el sistema operativo 108A la gestione.
20

Desde la operación 610, la rutina 600 procede a la operación 612, en la que el sistema operativo 108A enruta la llamada a la API desde la aplicación externa 118 al recurso original 504. De esta manera, las llamadas a la API realizadas desde aplicaciones no empresariales, tales como la aplicación externa 118, no pueden acceder al recurso 506. Desde la operación 612, la rutina 600 procede a la operación 614, en la que finaliza la rutina 600.
25

La figura 7 es un diagrama de arquitectura de ordenador que muestra una arquitectura de software y hardware de ordenador ilustrativa para un sistema informático capaz de desplegar las diversas realizaciones presentadas en la presente memoria descriptiva. La arquitectura de ordenador que se muestra en la figura 7 ilustra un ordenador de escritorio, ordenador portátil o servidor convencional, y puede utilizarse para ejecutar los diversos componentes de software que se han descrito en la presente memoria descriptiva.
30

La arquitectura de ordenador que se muestra en la figura 7 incluye una unidad central de procesamiento 702 ("CPU"), una memoria del sistema 708, que incluye una memoria de acceso aleatorio 714 ("RAM") y una memoria de solo lectura ("ROM") 716, y un bus de sistema 704 que acopla la memoria a la CPU 702. Un sistema básico de entrada / salida ("BIOS") que contiene las rutinas básicas que ayudan a transferir información entre elementos dentro del ordenador 700, como durante el inicio, se almacena en la ROM 716. El ordenador 700 incluye además un dispositivo de almacenamiento masivo 710 para almacenar un sistema operativo 718, programas de aplicación y otros módulos de programa, que se describirán con mayor detalle a continuación.
35

El dispositivo de almacenamiento masivo 710 está conectado a la CPU 702 a través de un controlador de almacenamiento masivo (no mostrado) conectado al bus 704. El dispositivo de almacenamiento masivo 710 y sus medios de almacenamiento asociados legibles por ordenador proporcionan almacenamiento no volátil para el ordenador 700. Aunque la descripción de los medios legibles por ordenador contenidos en la presente memoria descriptiva se refiere a un dispositivo de almacenamiento masivo, tal como un disco duro o una unidad de CD - ROM, los expertos en la técnica deberían apreciar que los medios de almacenamiento legibles por ordenador pueden ser cualquier medio de almacenamiento informático disponible. al que se puede acceder desde el ordenador 700.
40
45

A modo de ejemplo, y no de limitación, los medios de almacenamiento legibles por ordenador pueden incluir medios volátiles y no volátiles, extraíbles y no extraíbles implementados en cualquier procedimiento o tecnología para el almacenamiento de información, tal como instrucciones legibles por ordenador, estructuras de datos, módulos de programas, u otros datos. Por ejemplo, los medios de almacenamiento legibles por ordenador incluyen, entre otros, RAM, ROM, EPROM, EEPROM, memoria flash u otra tecnología de memoria de estado sólido, CD - ROM, discos versátiles digitales ("DVD"), HD - DVD, BLU - RAY, u otro almacenamiento óptico, casetes magnéticas, cinta magnética, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio no transitorio que pueda ser usado para almacenar la información deseada y al que pueda acceder el ordenador 700.
50

Se debe apreciar que los medios legibles por ordenador que se han descrito en la presente memoria descriptiva también incluyen medios de comunicación. Los medios de comunicación generalmente incorporan instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos en una señal de datos modulada,
55

tal como una onda portadora u otro mecanismo de transporte, e incluye cualquier medio de entrega de información. El término "señal de datos modulada" significa una señal que tiene una o más de sus características configuradas o modificadas de tal manera que codifican información en la señal. A modo de ejemplo, y no de limitación, los medios de comunicación incluyen medios cableados, tales como una red cableada o conexión directa, y medios inalámbricos, tales como medios acústicos, de RF, infrarrojos y otros inalámbricos. Las combinaciones de cualquiera de los anteriores también deben incluirse dentro del alcance de los medios legibles por ordenador. Los medios de almacenamiento legibles por ordenador no incluyen medios de comunicación.

De acuerdo con varias realizaciones, el ordenador 700 puede operar en un entorno de red usando conexiones lógicas a ordenadores remotos a través de una red tal como la red 720. El ordenador 700 puede conectarse a la red 720 a través de una unidad de interfaz de red 706 conectada al bus 704. Se debe apreciar que la unidad de interfaz de red 706 también se puede utilizar para conectarse a otros tipos de redes y sistemas informáticos remotos. El ordenador 700 también puede incluir un controlador de entrada / salida 712 para recibir y procesar la entrada de una serie de otros dispositivos, incluido un teclado, un ratón o un lápiz electrónico (no mostrado en la figura 7). De manera similar, un controlador de entrada / salida puede proporcionar salida a una pantalla de visualización, una impresora u otro tipo de dispositivo de salida (tampoco mostrado en la figura 7).

Como se ha mencionado brevemente más arriba, varios módulos de programa y archivos de datos pueden almacenarse en el dispositivo de almacenamiento masivo 710 y RAM 714 del ordenador 700, incluido un sistema operativo 718 adecuado para controlar la operación de un ordenador de escritorio, ordenador portátil o servidor en red.. El dispositivo de almacenamiento masivo 710 y la RAM 714 también pueden almacenar uno o más módulos de programa. En particular, el dispositivo de almacenamiento masivo 710 y la RAM 714 pueden almacenar la aplicación virtualizada 112, el componente de cliente de virtualización de aplicaciones 110 y / o los otros componentes de software que se han descrito más arriba. El dispositivo de almacenamiento masivo 710 y la RAM 714 también pueden almacenar otros módulos de programa y datos, tales como las dependencias 114.

En general, las aplicaciones de software o los módulos, cuando se cargan en la CPU 702 y se ejecutan, pueden transformar la CPU 702 y el ordenador en conjunto 700 de un sistema informático de propósito general en un sistema informático de propósito especial personalizado para realizar la funcionalidad presentada en la presente memoria descriptiva. La CPU 702 se puede construir a partir de cualquier número de transistores u otros elementos de circuitos discretos, que pueden asumir individual o colectivamente cualquier número de estado. Más específicamente, la CPU 702 puede operar como una o más máquinas de estado finito, en respuesta a instrucciones ejecutables contenidas en el software o en los módulos. Estas instrucciones ejecutables por ordenador pueden transformar la CPU 702 especificando cómo la CPU 702 transita entre estados, transformando físicamente los transistores u otros elementos de hardware discretos que constituyen la CPU 702.

La codificación del software o los módulos en un dispositivo de almacenamiento masivo también puede transformar la estructura física del dispositivo de almacenamiento masivo o los medios de almacenamiento asociados legibles por ordenador. La transformación específica de la estructura física puede depender de varios factores, en diferentes implementaciones de esta descripción. Los ejemplos de tales factores pueden incluir, entre otros: la tecnología utilizada para desplegar los medios de almacenamiento legibles por ordenador, ya sea porque los medios de almacenamiento legibles por ordenador estén caracterizados como almacenamiento primario o secundario, y otros similares. Por ejemplo, si los medios de almacenamiento legibles por ordenador se implementan como memoria basada en semiconductores, el software o los módulos pueden transformar el estado físico de la memoria de semiconductores, cuando el software está codificado en la misma. Por ejemplo, el software puede transformar los estados de los transistores, condensadores u otros elementos discretos del circuito que constituyen la memoria del semiconductor.

Como otro ejemplo, los medios de almacenamiento legibles por ordenador pueden desplegarse utilizando tecnología magnética u óptica. En tales implementaciones, el software o los módulos pueden transformar el estado físico de los medios magnéticos u ópticos, cuando el software está codificado en ellos. Estas transformaciones pueden incluir alterar las características magnéticas de ubicaciones particulares dentro de los medios magnéticos dados. Estas transformaciones también pueden incluir alterar las características físicas o características de ubicaciones particulares dentro de los medios ópticos dados, para cambiar las características ópticas de esas ubicaciones. Otras transformaciones de los medios físicos son posibles sin apartarse del ámbito y el espíritu de la presente descripción, con los ejemplos anteriores proporcionados solo para facilitar esta explicación.

En base a lo anterior, se debe apreciar que las tecnologías para garantizar que las dependencias de la aplicación se satisfacen en el tiempo de ejecución se han presentado en la presente memoria descriptiva. Aunque el objeto presentado en la presente memoria descriptiva que se ha descrito en un lenguaje específico para características estructurales de ordenador, actos metodológicos y medios legibles por ordenador, se debe entender que la invención definida en las reivindicaciones adjuntas no está necesariamente limitada a las características, actos o acciones específicas o medios que se ha descritos en la presente memoria descriptiva. Más bien, las características, los actos y los medios específicos se revelan como formas ejemplares de implementación de las reivindicaciones.

El objeto que se ha descrito más arriba se proporciona solo a modo de ilustración y no se debe interpretar como limitante. Se pueden realizar varias modificaciones y cambios al objeto que se ha descrito en la presente memoria descriptiva sin seguir las realizaciones ejemplares y las aplicaciones ilustradas y que se describen, y sin apartarse del alcance de la presente invención, que se expone en las siguientes reivindicaciones.

5

REIVINDICACIONES

1. Un procedimiento implementado por ordenador (400) que comprende realizar operaciones implementadas en ordenador para:
 - recibir (402) una solicitud para desplegar una aplicación virtualizada (112) en un dispositivo cliente (102);
 - 5 determinar (404) si una o más dependencias (114) requeridas para desplegar la aplicación virtualizada está presente en el dispositivo cliente; y
 - al determinar que cada una de las una o más dependencias necesarias para desplegar la aplicación virtualizada no está presente, configurar (410) el dispositivo cliente de tal manera que las dependencias estén presentes,
 - 10 en el que la una o más dependencias comprenden ajustes de configuración, componentes de hardware, conexiones de red o privilegios de acceso,
 - el procedimiento comprende además
 - al determinar que cada una de las una o más dependencias necesarias para desplegar la aplicación virtualizada están presentes, se despliega (414) la aplicación virtualizada en el dispositivo cliente; y
 - 15 al desplegar la aplicación virtualizada en el dispositivo cliente, interceptar (602) una interfaz de programación de aplicaciones, API, llamar desde la aplicación virtualizada la llamada a la API que solicita acceso a un recurso;
 - seleccionar (604) una copia del recurso (506) almacenado en una ubicación privada a la que se redirige la llamada a la API ; y
 - 20 redireccionar (608) la llamada a la API al recurso almacenado en la ubicación privada.
2. El procedimiento implementado por ordenador de la reivindicación 1, que comprende además:
 - proteger (606) la ubicación privada con una lista de control de acceso, ACL, asociada con el recurso.
3. El procedimiento implementado por ordenador de la reivindicación 1, que comprende (300) además, antes de desplegar la aplicación virtualizada, instalar (302) la aplicación virtualizada en una estación de monitorización (104); monitorizar (304) los cambios realizados en la estación de monitorización durante la instalación de la aplicación virtualizada para identificar las dependencias; y almacenar (312) datos que identifican las dependencias en una base de datos de análisis de dependencias (124).
4. El procedimiento implementado por ordenador de la reivindicación 3, que además comprende antes de desplegar la aplicación virtualizada:
 - 30 ejecutar (306) la aplicación virtualizada en la estación de monitorización;
 - monitorizar los cambios realizados en la estación de monitorización durante la ejecución de la aplicación virtualizada para identificar las dependencias; y
 - almacenar (312) datos que identifican las dependencias en la base de datos de análisis de dependencias (124).
5. El procedimiento implementado por ordenador de la reivindicación 4, en el que los cambios de monitorización realizados en la estación de monitorización durante la ejecución de la aplicación virtualizada comprenden:
 - interceptar (308) llamadas a la API realizadas por la aplicación virtualizada durante la ejecución; e
 - inspeccionar (310) los parámetros contenidos en las llamadas a la API para identificar las dependencias.
6. El procedimiento implementado por ordenador de la reivindicación 5, que comprende además:
 - 40 determinar (314) si cada una de las una o más dependencias se agrega o configura durante la instalación o despliegue de la aplicación virtualizada al determinar que cada una de las una o más dependencias se agrega o configura durante la instalación o despliegue del almacenamiento de la aplicación virtualizada (316) almacenando datos que indican que las una o más dependencias que se agregan o configuran durante la instalación o despliegue de la aplicación virtualizada deberán estar presentes en un tiempo de ejecución de la aplicación virtualizada.
 - 45

7. El procedimiento implementado por ordenador de la reivindicación 6, en el que determinar si cada una de las una o más dependencias se agrega o configura durante la instalación o despliegue de la aplicación virtualizada comprende consultar un registro provisto por el sistema operativo (108) de la estación de monitorización.
- 5 8. El procedimiento implementado por ordenador de la reivindicación 6, en el que determinar si cada una de las una o más de las dependencias se agrega o configura durante la instalación o despliegue de la aplicación virtualizada comprende consultar una tabla de asignación interna que coincida con las llamadas a la API a las dependencias.
9. Un medio de almacenamiento legible por ordenador que tiene instrucciones ejecutables por ordenador almacenadas, después de esto, cuando son ejecutadas por un ordenador, hacen que el ordenador:
- 10 reciba (402) una solicitud para desplegar una aplicación virtualizada (112);
- determine (404) si cada una de una o más dependencias (114) requeridas para desplegar la aplicación virtualizada está presente;
- al determinar que cada una de las una o más dependencias no está presente, configure (410) un dispositivo cliente en el que la aplicación virtualizada se está ejecutando de tal manera que las dependencias estén presentes,
- 15 en el que la una o más dependencias comprende ajustes de configuración, componentes de hardware, conexiones de red o privilegios de acceso;
- despliegue (414) la aplicación virtualizada en el dispositivo cliente en respuesta a la determinación de que cada una de las una o más dependencias necesarias para desplegar la aplicación virtualizada están presentes; y
- 20 al desplegar la aplicación virtualizada en el dispositivo cliente, intercepte (602) una interfaz de programación de aplicaciones, API, llame desde la aplicación virtualizada, solicitando la llamada a la API acceso a un recurso;
- seleccione (604) una copia del recurso (506) almacenado en una ubicación privada a la que se redirige la llamada a la API ; y
- 25 redirige (612) la llamada a la API al recurso almacenado en la ubicación privada.
10. El medio de almacenamiento legible por ordenador de la reivindicación 9, que tiene más instrucciones ejecutables por ordenador almacenadas en el mismo que, cuando son ejecutadas por el ordenador, hacen que el ordenador:
- 30 proteja (606) la ubicación privada con una lista de control de acceso, ACL, asociada con el recurso.
11. El medio de almacenamiento legible por ordenador de la reivindicación 9, que tiene más instrucciones ejecutables por ordenador almacenadas en el mismo que, cuando son ejecutadas por el ordenador, hacen que el ordenador:
- instale (302) la aplicación virtualizada en una estación de monitorización (104);
- 35 monitorice (304) los cambios realizados en la estación de monitorización durante la instalación de la aplicación virtualizada para identificar las dependencias; y
- empaquete datos que identifican las dependencias en un paquete de aplicaciones configurado para ser transmitido al dispositivo cliente.
12. El medio de almacenamiento legible por ordenador de la reivindicación 11, que tiene más instrucciones ejecutables por ordenador almacenadas en el mismo que, cuando son ejecutadas por el ordenador, hacen que el ordenador:
- 40 ejecute (306) la aplicación virtualizada instalada en la estación de monitorización;
- monitorice los cambios realizados en la estación de monitorización durante la ejecución de la aplicación virtualizada para identificar las dependencias; y
- 45 empaquete datos que identifican las dependencias en el paquete de aplicaciones configurado para ser transmitido al dispositivo cliente.

13. El medio de almacenamiento legible por ordenador de la reivindicación 12, en el que los cambios de monitorización realizados en la estación de monitorización durante la ejecución de la aplicación virtualizada para identificar las dependencias comprende:

- interceptar (308) llamadas a la API realizadas por la aplicación virtualizada durante la ejecución; e
- 5 inspeccionar (310) los parámetros contenidos en las llamadas a la API para identificar las dependencias
- proteger una ubicación privada con una lista de control de acceso, ACL, asociada con el recurso; y
- redireccionar la llamada a la API al recurso almacenado en la ubicación privada.

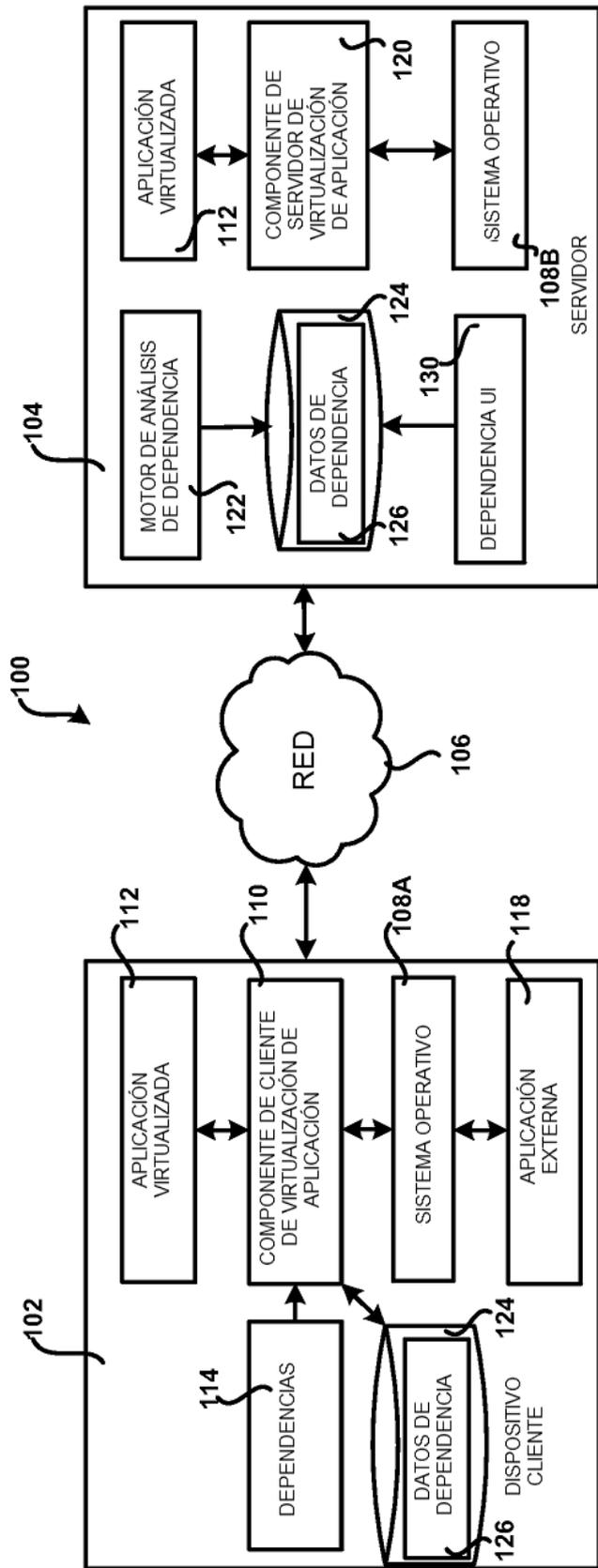


FIG. 1

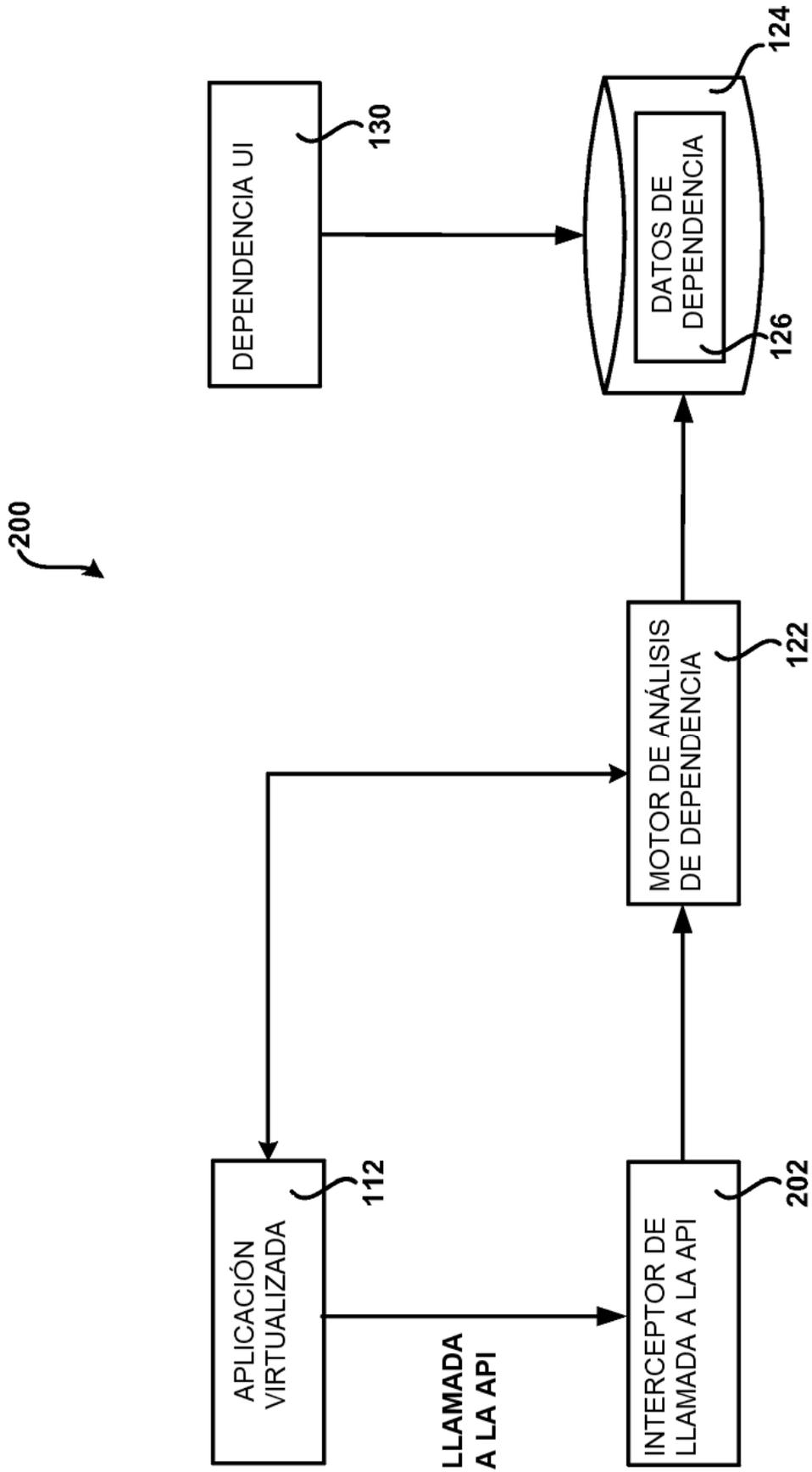


FIG. 2

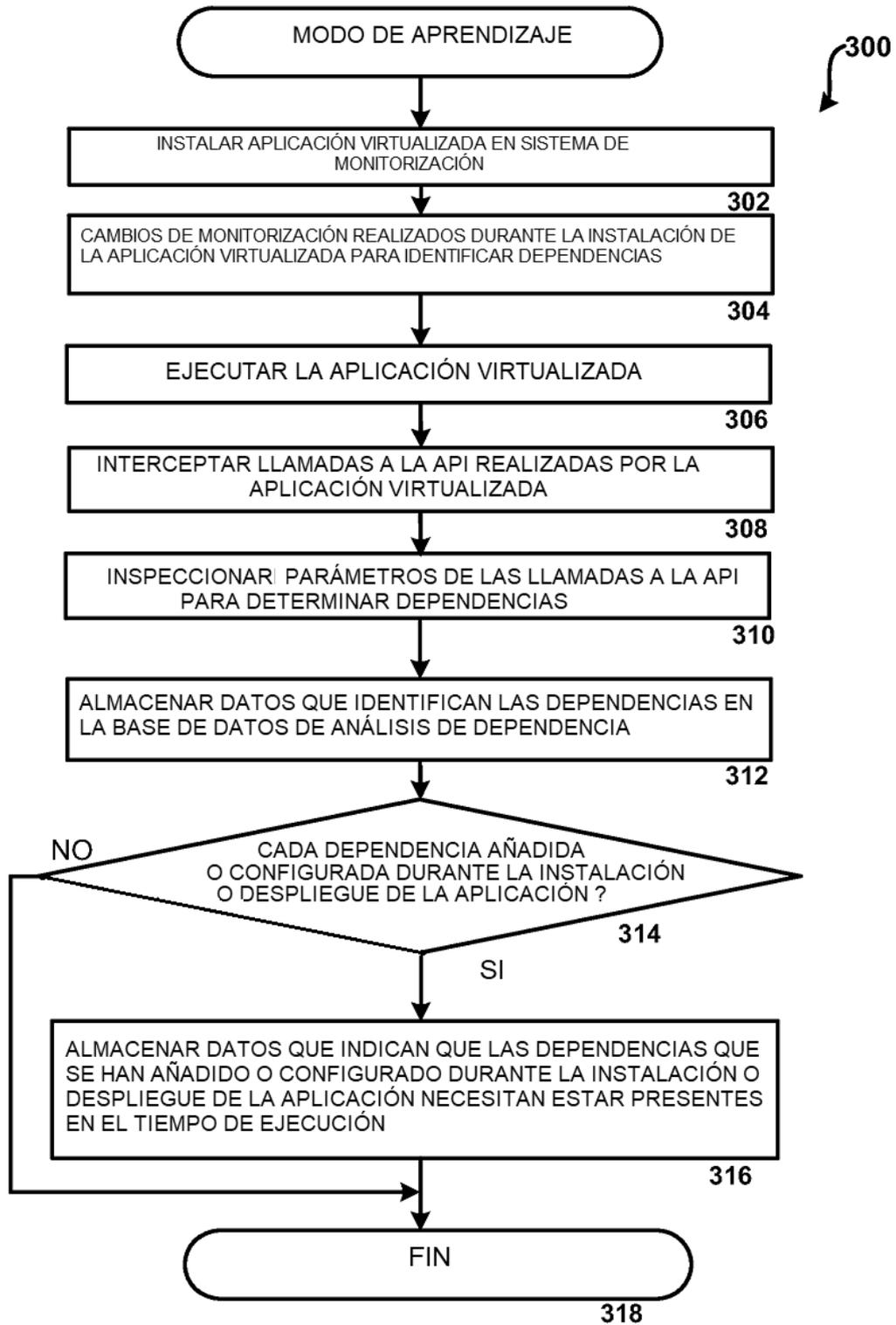


FIG. 3

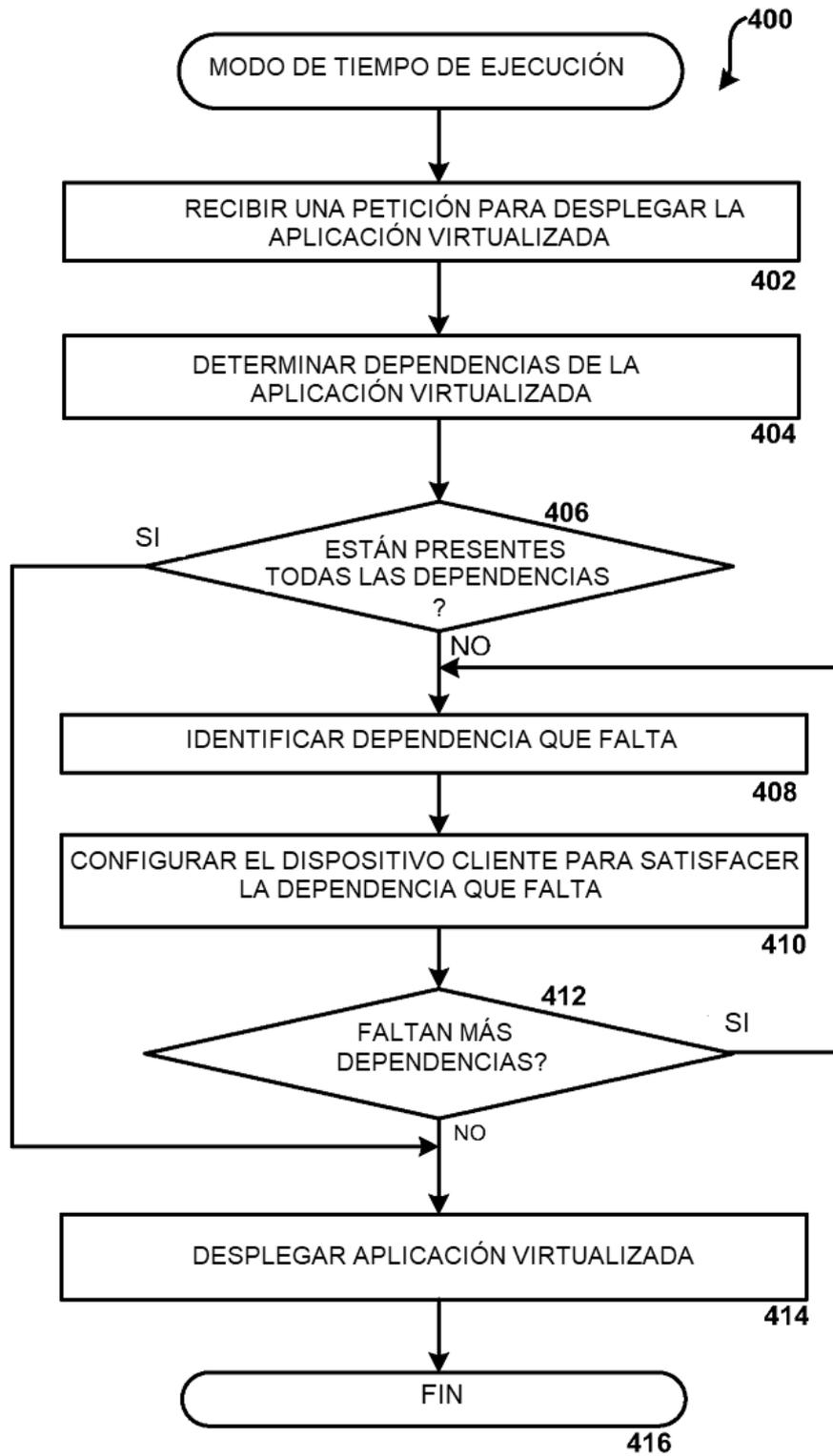


FIG. 4

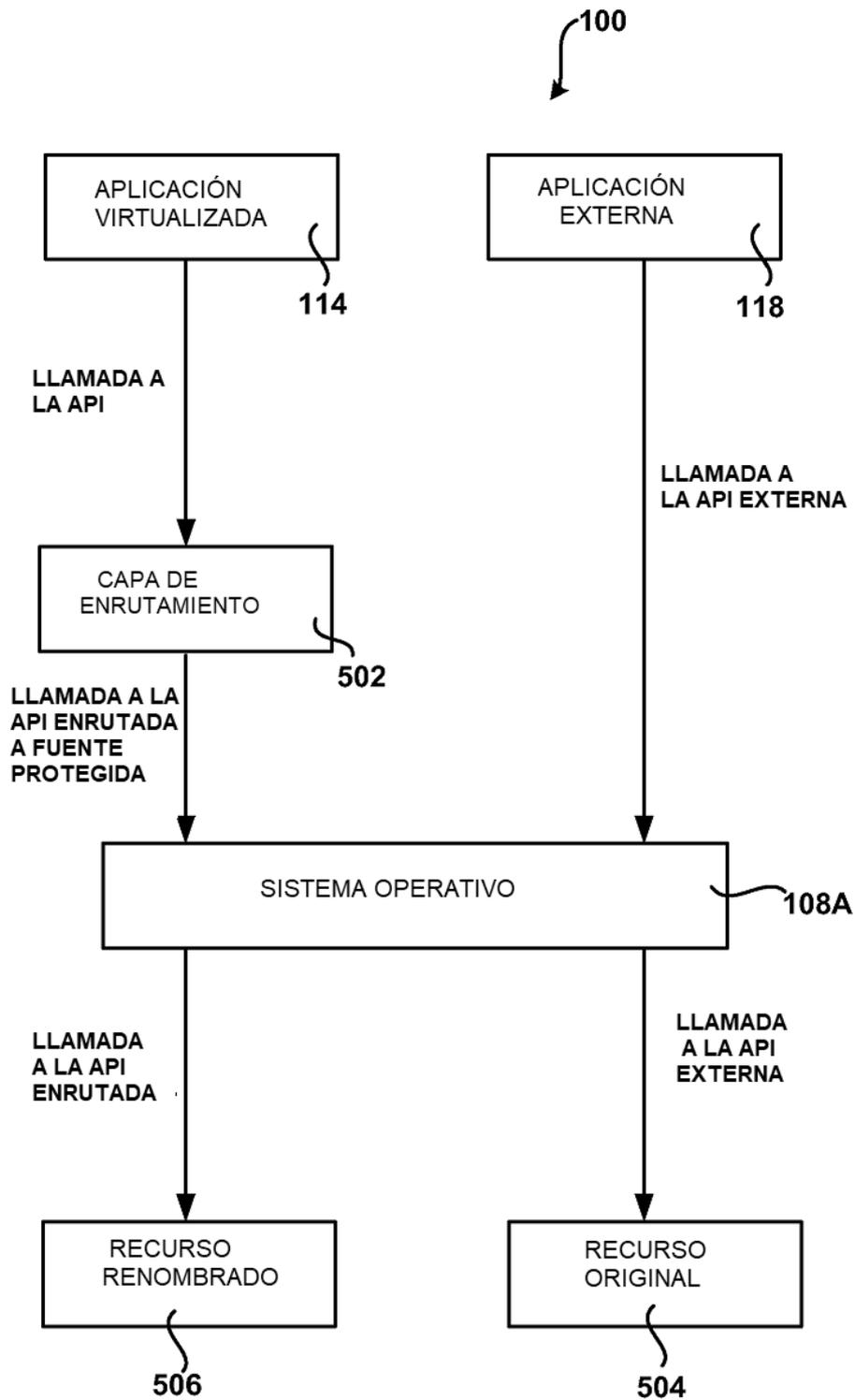


FIG. 5

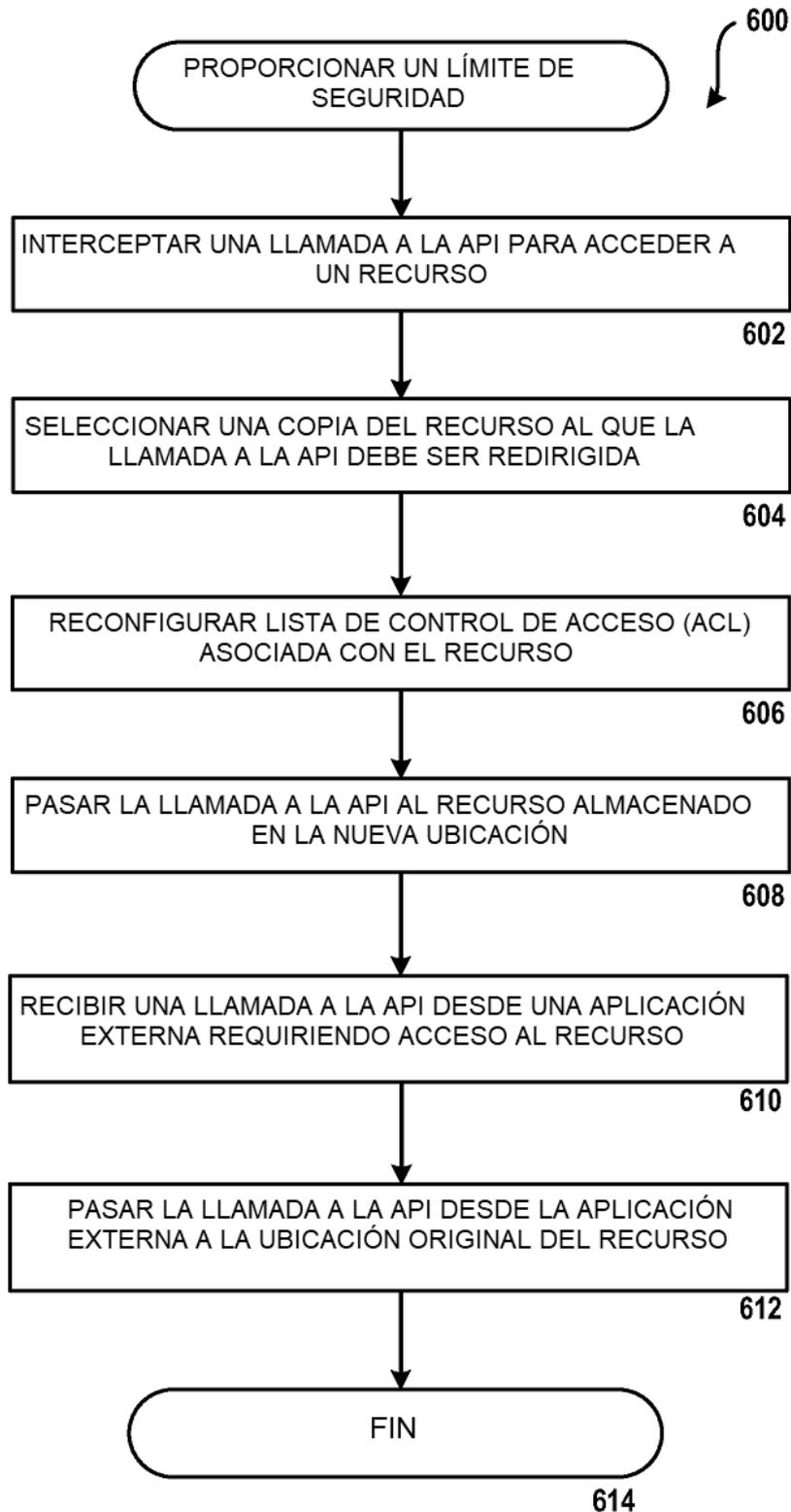


FIG. 6

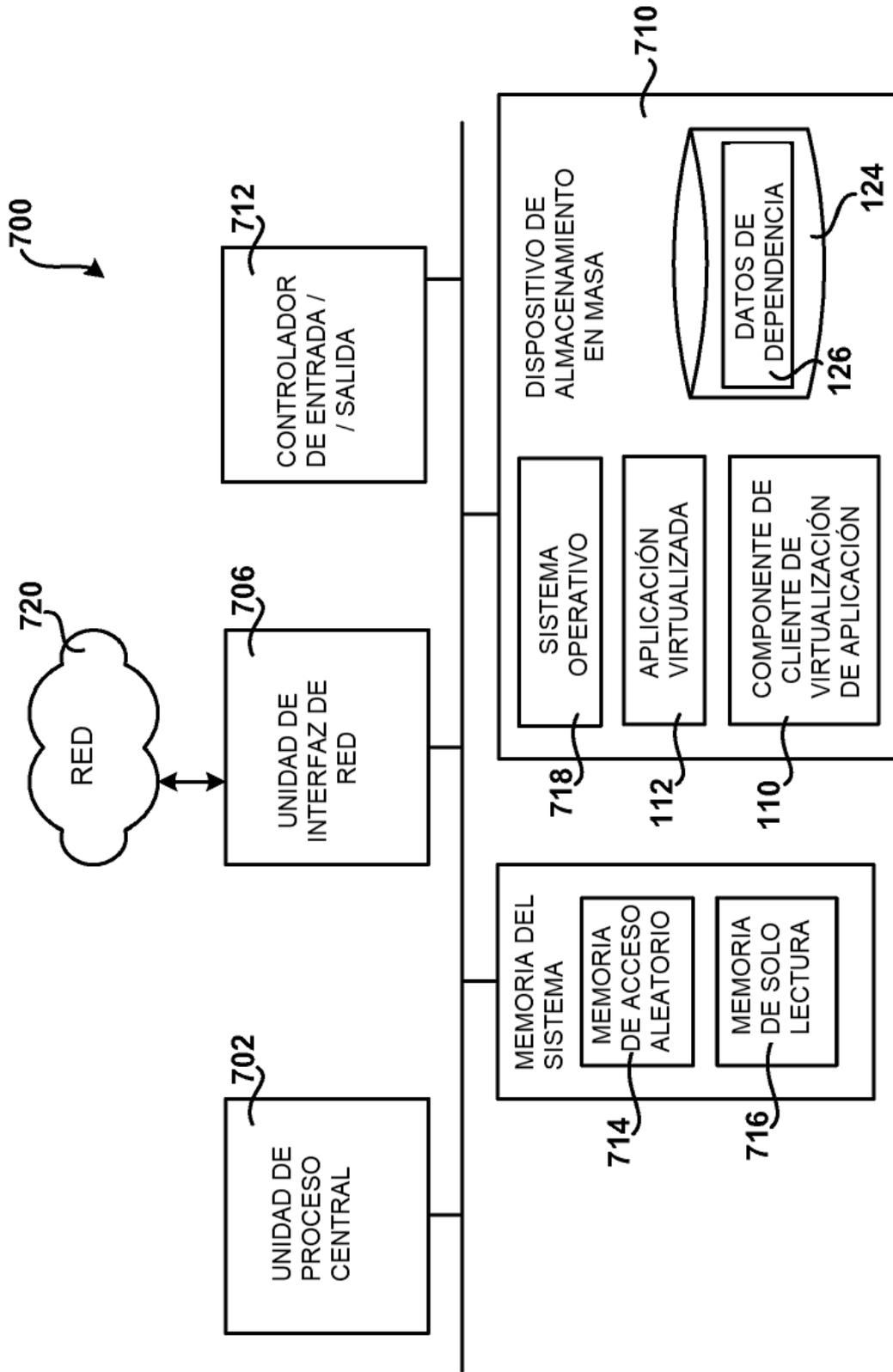


FIG. 7