

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 754 323**

51 Int. Cl.:

G16H 10/60 (2008.01)

G06F 16/00 (2009.01)

G06F 11/07 (2006.01)

G06F 9/54 (2006.01)

G06F 16/215 (2009.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **10.06.2015** **E 15171520 (8)**

97 Fecha y número de publicación de la concesión europea: **14.08.2019** **EP 2958037**

54 Título: **Recopilación y limpieza de datos en la fuente**

30 Prioridad:

17.06.2014 US 201414307235

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

17.04.2020

73 Titular/es:

**IQVIA INC. (100.0%)
100 IMS Drive
Parsippany, NJ 07054, US**

72 Inventor/es:

**FLOREZ, ELKIN y
BLAIR, CHARLES**

74 Agente/Representante:

PONS ARIÑO, Ángel

ES 2 754 323 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

Recopilación y limpieza de datos en la fuente

5 ANTECEDENTES

Campo de la invención

Las realizaciones de la presente invención generalmente se refieren a la recopilación de datos y, en particular, a un sistema y procedimiento para definir la estructura de datos dinámicamente y detectar la recopilación de datos erróneos más cerca de una fuente de datos, y corregir los datos erróneos o limitar su uso posterior.

Descripción de la técnica relacionada

15 Salud móvil ("mHealth") es un término para la práctica médica y de salud pública soportada por terminales de comunicación como teléfonos móviles, dispositivos de supervisión de pacientes, asistentes digitales personales (PDA) y otros dispositivos móviles o inalámbricos. mHealth implica el uso de servicios de voz y mensajes cortos (SMS), así como tecnologías más complejas, como sistemas de comunicación de datos móviles (por ejemplo, 3G, 4G, 4GLTE, etc.), sistemas de posicionamiento global (GPS) y tecnología Bluetooth.

20 Una aplicación móvil (o aplicación móvil) es una aplicación de software diseñada para ejecutarse en teléfonos inteligentes, ordenadores tipo tableta y otros dispositivos móviles. Algunas aplicaciones móviles se utilizan para entregar información personal confidencial, como información de atención sanitaria a los consumidores, o para reunir y enviar información del estado de salud de un consumidor a un proveedor de atención sanitaria. No todas las aplicaciones móviles que se relacionan con el intercambio de información personal confidencial, por ejemplo, aquellas que se han desarrollado en la atención sanitaria, están ampliamente disponibles para los consumidores. Algunas de las aplicaciones médicas más avanzadas no están necesariamente diseñadas para consumidores generales objetivo. Algunas aplicaciones móviles se han diseñado para profesionales de la atención sanitaria, otras son para pacientes pero requieren una receta, y otras están destinadas solo a un pequeño subconjunto de pacientes. Como se usa en este documento, el término "aplicación móvil" o "aplicación móvil" puede incluir una aplicación que se ejecuta en un PC (por ejemplo, de escritorio, de torre, ordenador portátil, netbook, etc.) u otro dispositivo informático de consumo de uso general, sin limitación a un dispositivo móvil a menos que la movilidad proporcione un beneficio declarado o que, de lo contrario, esté claramente restringido por el contexto de uso.

35 Un sistema de información, como un sistema utilizado para la atención sanitaria, puede generar y usar datos electrónicos recopilados en múltiples fuentes, como aplicaciones móviles, aplicaciones de escritorio, aplicaciones web, etc. Los datos electrónicos están inherentemente sujetos a errores de entrada de datos, independientemente de si los datos se introducen manualmente o mediante lecturas de sensor. Por ejemplo, los datos electrónicos introducidos manualmente pueden incluir errores tipográficos (por ejemplo, dígito mal escrito, dígitos transpuestos, dígitos desaparecidos, doble pulsación de tecla, pulsación de tecla sin dígitos, punto decimal equivocado, etc.). Las lecturas electrónicas de sensor pueden sufrir fallos del sensor (por ejemplo, producir como salida todo ceros o algún otro código de invalidez), colocación incorrecta o desplazamiento del sensor, fallos de la línea de comunicación, influencia indebida del ambiente (por ejemplo, efectos de la temperatura, vibración o interferencia de RF), inexactitud de la base de tiempo, etc. Los errores de entrada de datos deben corregirse o eliminarse para que el análisis y las decisiones se realicen utilizando solo datos fidedignos, es decir, datos electrónicos que estén sustancialmente libres de errores o que tengan errores residuales insignificantes. Los datos también deberían ser significativos, es decir, ser lo suficientemente exhaustivos o completos, como por incluir un muestreo estadísticamente significativo, o un ciclo completo de un proceso subyacente, o similares, según lo dicte la situación. Los datos electrónicos fidedignos y significativos, es decir, información que sea lo suficientemente fiable y completa como para actuar, es información procesable que se puede analizar y utilizar para tomar y respaldar decisiones.

Los procesos actualmente conocidos para transformar datos en información procesable requieren múltiples etapas, son costosos y requieren mucho tiempo y a menudo dan lugar a datos incompletos o inexactos. Un primer problema es que es muy costoso transformar los datos en información procesable cuando (como en la mayoría de los casos) al menos algunos de los datos no son válidos de alguna manera, como por estar incompletos, ser inexactos o contener elementos basura. Los datos recopilados se envían a un sitio de limpieza, por ejemplo, un sistema de procesamiento en el que se pueden utilizar sistemas manuales o semiautomatizados para detectar presuntos datos no válidos, y para marcar, corregir y/o eliminar los presuntos datos no válidos. Cuando los datos con este tipo de deficiencias pasan por un proceso de limpieza, la información resultante aún puede contener lagunas, inexactitudes y/o información no válida.

60 Un segundo problema es que un proceso de limpieza de datos convencional lleva mucho tiempo y generalmente se realiza utilizando un proceso por lotes y la fuente de datos no está disponible para proporcionar correcciones. Por

ejemplo, primero se tienen que recopilar los datos, a continuación los datos se envían a un servidor para su limpieza. Se utiliza un proceso por lotes para que el proceso de limpieza se pueda realizar en periodos de inactividad cuando es menos probable que se necesite capacidad de cómputo para otras tareas, como la asistencia al usuario, o para que puedan someterse a los trabajos de limpieza sin que la fuente de la información asista además al proceso.

5

Un tercer problema de la técnica convencional concierne a aplicaciones que pueden procesar datos estructurados y realizar una validación simple de los datos de su fuente. Tales aplicaciones normalmente utilizan una expresión regular (como se conoce en su sentido informático) para impedir que un usuario introduzca datos erróneos o para proporcionar un límite de información que impida que el usuario introduzca información poco realista (por ejemplo, presión arterial superior a 500 mmHg). El problema con esta técnica convencional es que los escenarios de datos interrelacionados o dependientes más complejos requieren otros tipos de validaciones, que no se manejan fácilmente mediante una expresión regular. Para una validación de datos complejos, las aplicaciones tendrían que enviar los datos recopilados a un servidor de back-end para el procesamiento y la administración de control de calidad, lo que sería un proceso complicado y que requeriría mucho tiempo.

15

Un cuarto problema de la técnica convencional es que los desarrolladores de aplicaciones generalmente codifican de manera fija estructuras de datos específicas o reglas de validación (también conocidas como reglas de consistencia de datos) en sus aplicaciones para casos de uso específicos. Después, si la estructura de datos cambia o si cambian las reglas de validación de datos, se debe cambiar el código fuente de la aplicación y distribuir una nueva versión de la aplicación a la base de usuarios. Será necesario que el usuario descargue e instale una nueva versión de la aplicación para acceder a los últimos cambios. Un usuario que continúa utilizando una versión anterior de la aplicación puede introducir datos erróneos o datos que de otro modo no pasan las reglas de validación de datos más recientes.

20

Los antecedentes de la técnica conocidos no pueden actualizar sus estructuras de datos y reglas de calidad sobre la marcha. Los antecedentes de la técnica conocidos requieren cambios en el código fuente de la aplicación para cada actualización, y es necesario que los usuarios descarguen cada versión actualizada, lo cual es poco práctico e incómodo, y potencialmente propenso a problemas continuos con la calidad de datos si los usuarios pueden diferir o rechazar la descarga e instalación de versiones actualizadas de la aplicación.

25

Además, los antecedentes de la técnica conocidos normalmente realizan procesamiento de calidad de datos (por ejemplo, limpieza de datos) en el lado del servidor, especialmente para datos complejos. Por lo tanto, los datos del usuario final se recopilan en la aplicación y después se envían al servidor de aplicaciones para procesamiento por lotes. Esto lleva tiempo, es costoso y no se presta bien a un modelo interactivo.

30

El documento EP 2590124 se refiere a una solución de calidad de datos basada en el conocimiento que se basa en una base de conocimiento abundante. La solución de calidad de datos puede proporcionar una mejora continua y puede basarse en la adquisición continua (o constante) de conocimiento. La solución de calidad de datos se puede construir una vez y se puede reutilizar para múltiples mejoras de calidad de datos, que pueden ser para los mismos datos o para datos similares.

35

El documento US 2006/247944 se refiere a la provisión de limpieza escalable y mejora de valor de los datos en el contexto de un repositorio de datos de múltiples fuentes y múltiples poseedores. Los datos de origen provienen de múltiples fuentes y sobre múltiples temas. Se utilizan etiquetas de datos de origen rastreados evolutivamente para contener información de seguimiento que refleja la naturaleza y las fuentes de cada cambio en los datos, tal como se ven afectados durante las diversas etapas del procesamiento de datos. Las etapas de procesamiento incluyen validación, normalización, limpieza de fuente única y procesos de fuente cruzada. Se aplican diversas reglas durante estas etapas, y se utilizan etiquetas de datos de origen rastreados evolutivamente para registrar fuentes y agentes de todos los cambios en los datos. A medida que se procesa, transforma y añade información al repositorio, las etiquetas de datos de origen rastreados evolutivamente correspondientes se almacenan en asociación con los diversos elementos de información. La información contenida en estas etiquetas se puede utilizar para imponer permisos de datos en un entorno de repositorio de datos de múltiples poseedores.

45

El documento US 2009/018996 se refiere a un procedimiento que implica recibir un conjunto de datos en una plataforma analítica, incluyendo el conjunto de datos de hechos y datos de dimensiones para una pluralidad de categorías de productos distintas. Almacenar los datos en una jerarquía flexible, permitiendo la fijación temporal de datos a lo largo de una dimensión y consultas flexibles a lo largo de otras dimensiones de los datos. Añadir previamente ciertas combinaciones de datos para facilitar la consulta rápida, el añadido previo basado en la naturaleza de las consultas comunes. Facilitar la presentación de una vista de categorías cruzadas de una consulta analítica del conjunto de datos. Puede hacerse que la dimensión fija temporalmente resulte flexible tras una acción por parte del usuario.

55

60

US 2009/182580 se refiere a un sistema y procedimiento para añadir de manera automatizada datos relacionados

médicamente y difundir diferentes conjuntos de datos añadidos a dos o más miembros diferentes de un proveedor de atención médica. Algunas realizaciones difunden de manera automatizada e inteligente los datos añadidos de modo que los datos que se escriben una vez en la solución de almacenamiento de datos se pueden utilizar para una multitud de propósitos dentro de las funciones del proveedor de atención médica. Algunas realizaciones difunden de manera
5 inteligente los datos añadidos difundiendo solo subcomponentes relevantes de los datos a diferentes miembros del proveedor de atención médica en función de las necesidades de los miembros, de modo que diferentes miembros reciban diferentes subconjuntos de los datos.

Por lo tanto, existe la necesidad de mejorar la validación de datos más cerca de la fuente de los datos, con el fin de
10 proporcionar información más fidedigna y procesable para apoyar la toma de decisiones, la limpieza de datos en la fuente y, en última instancia, proporcionar una mejor satisfacción y satisfacción del cliente.

RESUMEN

15 Las realizaciones según la presente descripción se refieren a un dispositivo móvil para limpiar datos como se define en la reivindicación 1 y un procedimiento para limpiar datos como se define en la reivindicación 11. En las reivindicaciones dependientes se definen características ventajosas.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

20 Las características y ventajas anteriores y aún otras adicionales de la presente invención resultarán evidentes tras considerar la siguiente descripción detallada de realizaciones de la misma, especialmente cuando se toman en conjunto con los dibujos adjuntos, donde números de referencia similares en las diversas figuras se utilizan para designar componentes similares y donde:

25 la FIG. 1 ilustra un archivo de configuración ejemplar según una realización de la presente invención;

la FIG. 2 ilustra a un nivel de abstracción modular relativamente alto un sistema según una realización de la presente descripción;

30 la FIG. 3A ilustra a un nivel de abstracción de hardware relativamente alto un sistema según una realización de la presente descripción;

35 la FIG. 3B ilustra a un nivel de abstracción de hardware relativamente alto un sistema según otra realización de la presente descripción;

la FIG. 4 ilustra un proceso para utilizar un sistema por un servidor, según una realización de la presente descripción;
y

40 la FIG. 5 ilustra un proceso para utilizar un sistema por un dispositivo móvil, según una realización de la presente descripción.

Los encabezamientos utilizados en esta solicitud son únicamente con fines organizativos y no pretenden utilizarse para limitar el alcance de la descripción o las reivindicaciones. Tal como se utiliza en toda esta solicitud, la palabra
45 "puede/n" se utiliza en un sentido permisivo (es decir, con el significado de contar con el potencial para), en lugar del sentido obligatorio (es decir, con el significado de deber). De manera similar, las palabras "incluir", "que incluye" e "incluye" significan que incluyen pero no limitado a. Para facilitar la comprensión, se han utilizado números de referencia similares, dentro de lo posible, para designar elementos similares comunes a las figuras. Las partes opcionales de las figuras pueden ilustrarse utilizando líneas de trazos o de puntos, a menos que el contexto de uso
50 indique otra cosa.

DESCRIPCIÓN DETALLADA

La descripción se ilustrará a continuación junto con un sistema de comunicación ejemplar. Aunque es adecuado para
55 su uso con, por ejemplo, un sistema que utiliza un(os) servidor(es) y/o base(s) de datos, la descripción no se limita al uso con ningún tipo particular de sistema de comunicación o configuración de elementos del sistema. Los expertos en la materia reconocerán que las técnicas descritas pueden utilizarse en cualquier aplicación de comunicación en la que sea deseable proporcionar recopilación de datos más procesable.

60 Los sistemas y procedimientos ejemplares de la presente descripción también se describirán en relación con el software, los módulos y el hardware asociado. Sin embargo, para evitar complicar innecesariamente la presente descripción, la siguiente descripción omite estructuras, componentes y dispositivos bien conocidos que pueden

mostrarse en forma de diagrama de bloques, que sean bien conocidos o se resuman de otra manera.

En la siguiente descripción detallada, se presentan numerosos detalles específicos con el fin de proporcionar una comprensión profunda de las realizaciones u otros ejemplos descritos en esta solicitud. En algunos casos, no se han descrito en detalle métodos, procedimientos, componentes y circuitos bien conocidos para no complicar la siguiente descripción. Además, los ejemplos descritos son únicamente con fines ejemplares y se pueden emplear otros ejemplos en lugar de, o en combinación con los ejemplos descritos. Cabe señalar también que los ejemplos presentados en esta solicitud no deberían interpretarse como limitativos del alcance de las realizaciones de la presente invención, ya que son posibles y probables otros ejemplos igualmente efectivos.

Tal como se utiliza en esta solicitud, el término “módulo” se refiere, en general, a una secuencia o asociación lógica de etapas, procesos o componentes. Por ejemplo, un módulo de software puede comprender un conjunto de rutinas o subrutinas asociadas dentro de un programa informático. Alternativamente, un módulo puede comprender un dispositivo de hardware sustancialmente independiente. Un módulo puede comprender también un conjunto lógico de procesos independientemente de cualquier implementación de software o hardware.

Tal como se utiliza en este documento, el término “pasarela” puede comprender en general cualquier dispositivo que envíe y reciba datos entre dispositivos. Por ejemplo, una pasarela puede comprender encaminadores, conmutadores, puentes, cortafuegos, otros elementos de red y similares, cualquiera y una combinación de los mismos.

Tal como se utiliza en este documento, el término “transmisor” puede comprender en general cualquier dispositivo, circuito o aparato capaz de transmitir una señal. Tal como se utiliza en este documento, el término “receptor” puede comprender en general cualquier dispositivo, circuito o aparato capaz de recibir una señal. Tal como se utiliza en este documento, el término “transceptor” puede comprender en general cualquier dispositivo, circuito o aparato capaz de transmitir y recibir una señal. Tal como se utiliza en este documento, el término “señal” puede incluir una o más de una señal eléctrica, una señal de radio, una señal óptica, una señal acústica, etc.

Tal como se utiliza en este documento, el término “contenedor de aplicaciones” puede referirse en general a una aplicación móvil que puede alojar y soportar el uso de varias configuraciones de aplicaciones. Cada configuración describe la apariencia de la GUI, el flujo de la aplicación, la lógica y los datos relevantes para una aplicación. El contenedor puede comenzar con una configuración que se identifica como la primera configuración. La primera configuración puede permitir que un usuario seleccione las otras configuraciones que se usarán.

El término “medio legible por ordenador”, tal como se utiliza en este documento, se refiere a cualquier medio tangible, no transitorio de almacenamiento y/o transmisión que participe en el almacenamiento y/o en proporcionar instrucciones a un procesador para su ejecución. Tal medio puede adoptar muchas formas, incluyendo, pero no limitadas a medios no volátiles, medios volátiles y medios de transmisión. Los medios no volátiles incluyen, por ejemplo, NVRAM o discos magnéticos u ópticos. Los medios volátiles incluyen memoria dinámica, como memoria principal. Las formas comunes de medios legibles por ordenador incluyen, por ejemplo, un disquete, un disco flexible, disco duro, cinta magnética o cualquier otro medio magnético, medio magneto-óptico, un CD-ROM, cualquier otro medio óptico, tarjetas perforadas, cinta de papel, cualquier otro medio físico con patrones de agujeros, RAM, PROM, EPROM, FLASH-EPROM, medio de estado sólido como una tarjeta de memoria, cualquier otro chip o cartucho de memoria, una onda portadora como se describe en lo sucesivo, o cualquier otro medio del cual un ordenador puede leer. Un archivo adjunto digital al correo electrónico u otro archivo de información independiente o conjunto de archivos se considera un medio de distribución equivalente a un medio de almacenamiento tangible. Cuando los medios legibles por ordenador están configurados como una base de datos, debe entenderse que la base de datos puede ser cualquier tipo de base de datos, como relacional, jerárquica, orientada a objetos y/o similares. Por consiguiente, se considera que la descripción incluye un medio de almacenamiento tangible o medio de distribución y equivalentes reconocidos en la técnica anterior y medios sucesores, en los que se almacenan las implementaciones de software de la presente descripción.

Las realizaciones según la presente descripción proporcionan y utilizan un conjunto de procesos configurables genéricos, implementados como módulos de interfaz de programa de aplicación (API), para permitir la administración de estructuras de datos desconocidas y reglas de validación en tiempo de ejecución. Los programas de aplicación que incorporan las realizaciones aceptarán documentos o datos solo si cumplen con una estructura de datos y un conjunto de reglas de validación definido durante el proceso de inicialización.

Las realizaciones según la presente descripción pueden adaptarse para ejecutarse en una diversidad de plataformas informáticas de destino (por ejemplo, teléfono inteligente, tableta, ordenador portátil, netbook, otro dispositivo móvil, ordenador de escritorio, etc.), que se describen a continuación con mayor detalle en el contexto de las FIGS. 3A y 3B.

Los procesos configurables genéricos, como una API de datos, una API de reglas, un motor de reglas, etc., pueden escribirse en código nativo para las plataformas informáticas de destino. El código nativo se conoce como código de programación de ordenador que se compila para ejecutarse directamente con un procesador particular y su conjunto

de instrucciones (por ejemplo, código máquina). Los procesos configurables genéricos pueden empaquetarse para su distribución a la plataforma informática de destino como componentes de un contenedor de aplicaciones (“aplicaciones”) genérico, nuevas estructuras y reglas de datos, o cambios en las estructuras y reglas de datos existentes. Los procesos configurables genéricos pueden implementarse sin necesidad de cambiar o actualizar el
5 código fuente del contenedor de aplicaciones genérico.

Las realizaciones según la presente descripción permiten que diferentes aplicaciones, utilizando un tipo compartido de estructura de datos, recopilen conjuntos de información estructurada en la fuente. Los conjuntos de información estructurada serán de mayor calidad (por ejemplo, más fidedignos y significativos) y bien organizados (por ejemplo,
10 que posean una estructura similar y consistente), al tiempo que reducen el número de etapas necesarias para limpiar los datos recopilados, con el fin de lograr información procesable de los datos recopilados. La limpieza de datos se refiere a la identificación y corrección o eliminación de datos no fidedignos, por ejemplo, datos que incluyen errores significativos o datos reunidos en condiciones desconocidas cuando el conocimiento de esas condiciones sería significativo para fines de normalización, compensación, corrección o similares. Un ejemplo de errores que pueden no
15 ser significativos son las mediciones que varían dentro de la tolerancia o exactitud habitual de un sensor o temporizador, o el error de muestreo estadístico. Un ejemplo de errores que pueden ser significativos son las mediciones que incluyen una variación relativamente grande o un sesgo sistemático (en relación con la tolerancia, la exactitud o el error de muestreo habituales) debido a la influencia del entorno (por ejemplo, efectos de temperatura, vibración o interferencia de RF). Puede ser difícil distinguir entre los errores que deberían limpiarse y los cambios verdaderos en un proceso
20 subyacente que se está supervisando. Las realizaciones, reduciendo la necesidad de limpiar datos, ahorran recursos al tiempo que mejoran la calidad de la información producida.

Con respecto a un inconveniente de los antecedentes de la técnica discutidos anteriormente, es decir, que es muy costoso transformar datos en información procesable cuando (como en la mayoría de los casos) al menos algunos de
25 los datos son inválidos de alguna manera, las realizaciones según la presente descripción pueden ser utilizadas por múltiples fuentes de información (por ejemplo, teléfono inteligente, tableta, ordenador portátil, netbook, otro dispositivo móvil, ordenador de escritorio, etc.) con el fin de recopilar datos en la fuente de datos de manera más eficiente y coherente. Las realizaciones logran este objetivo utilizando estructuras de datos, reglas y configuraciones de control de calidad para garantizar que los datos recopilados se ajusten a lo esperado, proporcionando una coherencia forzada.
30 Debido a que esta coherencia se produce cuando se introducen los datos, los usuarios pueden ser alertados de información desaparecida, información inexacta, información incoherente o información que no se ajusta al formato esperado. Determinar la coherencia puede implicar el uso de datos históricos y datos de otras fuentes. Los usuarios pueden entonces corregir la información antes de enviarla a un servidor para su posterior uso y análisis.

Con respecto a otro inconveniente de los antecedentes de la técnica discutidos anteriormente, es decir, que un proceso de limpieza de datos convencional lleva mucho tiempo y generalmente se realiza utilizando un proceso por lotes, las realizaciones según la presente descripción permiten que una aplicación (por ejemplo, una aplicación relacionada con
35 mHealth) administre la entrada de información durante una interacción del usuario con la aplicación. Mediante este proceso, la recopilación y administración de información puede ser interactiva, transaccional y rápida.

Además, debido a las características transaccionales e interactivas de las aplicaciones, las realizaciones permiten que la información se procese sustancialmente en tiempo real. Las realizaciones facilitan la posibilidad de actualizar y
40 administrar estadísticas o informes casi instantáneamente una vez que la información es recibida por un servidor, porque se considera que los datos son procesables. Esto puede ser muy valioso para muchos tipos de aplicaciones donde el tiempo puede ser esencial (por ejemplo, aplicaciones que evalúan los efectos de campañas publicitarias, aplicaciones que detectan afecciones médicas agudas, etc.).
45

Con respecto a otro inconveniente de los antecedentes de la técnica discutidos anteriormente, es decir, que los escenarios de datos interrelacionados o dependientes más complejos requieren tipos de validaciones que son
50 demasiado complejas para especificar con una expresión regular, o que no son administradas fácilmente por una expresión regular, las realizaciones permiten tanto el procesamiento estructurado de datos como el control de calidad de datos de sustancialmente todos los tipos de conjuntos de datos (por ejemplo, conjuntos de datos simples o complejos) en la fuente, sin necesidad de construir una validación en el lado del servidor cuando la fuente de esos datos puede que ya no esté disponible.
55

Con respecto a otro inconveniente de los antecedentes de la técnica discutidos anteriormente, es decir, que los desarrolladores de aplicaciones generalmente codifican estructuras de datos o reglas de validación específicas en sus aplicaciones para casos de uso específicos, las realizaciones permiten que un desarrollador desarrolle una aplicación
60 de software sin tener que preocuparse por el formato de las estructuras de datos y las reglas de validación, o tener que preocuparse por los cambios en las estructuras de datos y las reglas de validación, o tener que preocuparse por la administración de la calidad de datos. Las realizaciones ayudan a garantizar que la aplicación de software pueda cumplir con estándares o iniciativas predefinidas, como para usos de mHealth, proporcionando realizaciones para

administrar la información, proporcionada por los usuarios, en la fuente de la información. Los cambios en la estructura de datos de la información que debe reunirse (por ejemplo, definición de campo) o los requisitos de calidad de la información introducida (por ejemplo, nivel o grado de verificación de errores y verificación de coherencia de la información numérica; cálculo de interrelaciones entre datos) no requerirán cambios en los archivos de origen de la aplicación de software. Las realizaciones pueden administrar estructuras de datos y un conjunto de reglas de validación (incluyendo cambios en los mismos) mediante el uso de procedimientos de configuración y control de cambios formales. Las realizaciones tenderán a reducir la complejidad del desarrollo de aplicaciones de software, como las aplicaciones de mHealth, porque las estructuras de datos y la configuración del conjunto de reglas de validación se administrarán de manera coherente y accesible para los desarrolladores de software.

10

Las realizaciones según la presente descripción proporcionan un marco flexible para recopilar datos y transformar esos datos en información procesable, en la fuente. La flexibilidad incluye poder reconfigurar sobre la marcha los requisitos de formato y coherencia relacionados con los datos recopilados, es decir, en tiempo real sin necesidad de redistribuir o reiniciar el programa de aplicación. Las estructuras de datos y las reglas de validación requeridas para un tipo de enfermedad específico son importantes y pueden cambiar a lo largo del tiempo con el avance del conocimiento médico. En un momento particular, lo más novedoso en cuanto al conocimiento médico puede considerar que ciertos factores o comportamiento son marcadores de riesgo elevado de una enfermedad. Por ejemplo, cuatro tazas de café o dos copas de vino tinto por día pueden considerarse saludables un año, pero en un año posterior nuevos estudios consideran que no son saludables o que no tienen un efecto estadísticamente significativo. Como resultado, a lo largo del tiempo puede ser deseable que una aplicación médica recopile diferentes datos del paciente a lo largo del tiempo, o que analice los datos recopilados de diferentes maneras a lo largo del tiempo. Las realizaciones pueden adaptarse a tales avances en el conocimiento médico permitiendo y facilitando cambios en las estructuras de datos y las reglas de validación requeridas con el fin de continuar recopilando información procesable, sin necesidad de redistribuir la aplicación médica en sí.

25

Las realizaciones descritas en este documento no se limitan al uso en un contexto médico, sino que pueden ser útiles en una diversidad de contextos o industrias. Las realizaciones facilitan la recopilación de datos coherente a partir de aplicaciones similares, incluso cuando están escritas por diferentes desarrolladores, y la administración coherente de las aplicaciones similares.

30

Aunque otras tecnologías (por ejemplo, XML y el esquema XML) pueden ofrecer flexibilidad limitada en la visualización de información, estas tecnologías son deficientes a la hora de ofrecer una capacidad flexible para verificar la exactitud y la coherencia de los datos en la fuente, o para administrar la estructura de datos integrada y los motores de reglas de validación. Las realizaciones facilitan la verificación mejorada de errores y coherencia.

35

Las realizaciones según la presente descripción pueden utilizar una API de datos y un motor de reglas de validación que se codifican una vez a nivel nativo, pero se pueden reconfigurar en tiempo de ejecución para producir información procesable relevante para el propósito de una aplicación de software. Las realizaciones permiten así que la aplicación de software soporte las estructuras de datos y las reglas de validación requeridas establecidas para la coherencia de datos, la calidad de datos y el flujo de proceso, para producir información procesable.

40

Otra ventaja de las realizaciones según la presente descripción es que los cambios en las estructuras de datos y las reglas de validación pueden implementarse a un nivel de configuración en lugar de a un nivel de código fuente de aplicación. Por ejemplo, las implementaciones a nivel de configuración pueden basarse en un programa de aplicación que lea un archivo de configuración en el momento de la inicialización y, opcionalmente, periódicamente u otras veces adicionales a partir de entonces. Los cambios en las estructuras de datos y las reglas de validación pueden codificarse en el archivo de configuración. Por lo tanto, puede haber soporte para cambios sobre la marcha (es decir, cambios en tiempo real) que afectan el flujo de trabajo de la aplicación, la experiencia de usuario (UX) o la interfaz de usuario (UI), el tipo de datos o las reglas de validación sin causar una necesidad de cambiar, actualizar, o reiniciar la aplicación de software.

50

Se puede proporcionar un archivo de configuración mediante el uso de un lenguaje que se puede implementar de diferentes maneras, como utilizando la notación de objetos JavaScript (JSON) o el lenguaje de marcado extensible (XML), o sustancialmente cualquier otra implementación específica. El archivo de configuración puede comprender una descripción de estructura de datos que contenga una especificación de jerarquía y que tenga los campos de datos y marcadores de estructura asociados con las reglas de validación. Un archivo de configuración ejemplar que utiliza JSON se ilustra en la FIG. 1. Aunque la FIG. 1 ilustra un ejemplo de a qué puede parecerse un archivo de configuración, se contemplan otras implementaciones de un archivo de configuración.

55

Desde una perspectiva de implementación, la configuración sería administrada por la plataforma y la API de datos y los componentes del motor de reglas de validación se integrarían en un contenedor de aplicaciones, por ejemplo, mediante el uso de datos de configuración, archivo de inicialización y similares. Una API de datos y un motor de reglas

60

de validación leen las configuraciones y administran el comportamiento requerido en consecuencia. Un contenedor de aplicaciones (“aplicaciones”) puede ser una aplicación móvil que puede alojar y soportar el uso de varias configuraciones de aplicaciones. Cada configuración describe la apariencia de la GUI, el flujo de la aplicación, la lógica y los datos. El contenedor puede comenzar con una configuración que se identifica como la primera configuración. La primera configuración puede permitir que un usuario seleccione las otras configuraciones que se usarán.

Por ejemplo, supóngase que una dosis de un fármaco se especifica en unidades de “UI”. Además, supóngase que un usuario suministra mediciones del fármaco (por ejemplo, para supervisar u obligar al cumplimiento) en unidades de “mgs”. En tal situación, el motor de reglas de validación detectará este error.

Sin embargo, supóngase además que más adelante existe la necesidad o el deseo de actualizar las reglas de validación para aceptar la medición en unidades de “mgs”. Para los sistemas de la técnica conocida, tal actualización de las reglas de validación no puede realizarse sin la necesidad de actualizar el código fuente de la aplicación y después distribuir a los usuarios versiones compiladas o en tiempo de ejecución de la aplicación de software. Esto obliga a los usuarios a actualizar sus dispositivos con la nueva versión actualizada de la aplicación de software. Tal actualización es particularmente inconveniente para los usuarios móviles. Por ejemplo, pueden surgir inconveniencias debido a que los usuarios exceden su límite de transferencia de datos mensual, o los usuarios móviles pueden habituarse a actualizaciones frecuentes, o debido a un retraso en la disponibilidad de la aplicación de software causado por la necesidad de someter la aplicación de software a pruebas de validación y aprobación antes de que los usuarios puedan descargarla. Puede haber un retraso entre el momento en que una actualización de software está lista y el momento que un usuario permite o decide actualizar la aplicación de software.

Además, en ausencia de las realizaciones descritas en este documento, los datos recopilados de los usuarios aún tendrían que ser revisados, normalmente por una persona o un sistema, y limpiados si hay alguna deficiencia. Todo este procesamiento probablemente se realizaría en modo por lotes y llevaría mucho tiempo. Además, este procesamiento puede ser propenso a errores porque la limpieza de los datos es al menos parcialmente un proceso manual y, por lo tanto, está sujeto a errores manuales concomitantes.

Por el contrario, y tras tal solicitud de actualización de las reglas de validación, las realizaciones pueden reconfigurar la aplicación de software sin tener que distribuir una actualización de la aplicación de software en sí, a los usuarios finales, de modo que los usuarios finales pueden entonces introducir una dosis de su medicamento en unidades de “mgs”. De manera similar, el conjunto de reglas de validación también puede actualizarse para abordar los cambios en la dosificación de un fármaco, el número de repeticiones, la validación de interacciones del fármaco o la identificación de interacciones dañinas, etc.

La FIG. 2 ilustra a un nivel de abstracción modular o funcional relativamente alto un sistema (200) según una realización de la presente descripción; El sistema (200) incluye un dispositivo móvil (201) en comunicación con un servidor (203). El dispositivo móvil (201) puede incluir un programa de aplicación (205) que funciona para intercambiar información y/o comandos con el servidor (203), por ejemplo, para recibir comandos del servidor (203) con respecto al control de mediciones de datos, y para comunicar datos medidos desde el programa de aplicación (205) al servidor (203).

El dispositivo móvil (201) puede incluir además un módulo de descubrimiento e instanciación (207). El módulo de descubrimiento e instanciación (207) puede ser invocado por el programa de aplicación (205) cuando el programa de aplicación (205) necesita tomar una medida. La funcionalidad de instanciación del módulo de descubrimiento e instanciación (207) es utilizada por el programa de aplicación (205) para construir en la memoria todos los objetos requeridos por la API (es decir, las realizaciones) para funcionar. El programa de aplicación (205) utilizará la funcionalidad de descubrimiento del módulo de descubrimiento e instanciación (207) para aprender la estructura de datos disponible según la configuración proporcionada a la API. El programa de aplicación (205) puede utilizar la descripción de la información proporcionada por el módulo de descubrimiento e instanciación (207) para conocer los datos que necesitan ser solicitados o proporcionados al usuario. El programa de aplicación (205) también puede usar la API proporcionada por el módulo de descubrimiento e instanciación (207) para proporcionar los datos correspondientes según la configuración de datos. El módulo de descubrimiento e instanciación (207) delegará la validación de datos en el módulo administrador de estructura (209) descrito a continuación.

El dispositivo móvil (201) puede incluir además un módulo administrador de estructura (209) que está en contacto comunicativo con el módulo de descubrimiento e instanciación (207). El módulo de descubrimiento e instanciación (207) puede delegar en el módulo administrador de estructura (209) ciertas tareas relacionadas con la validación de datos. Por ejemplo, cuando el módulo de descubrimiento e instanciación (207) tiene una o más mediciones que necesitan ser validadas, el módulo de descubrimiento e instanciación (207) puede enviar los datos al módulo administrador de estructura (209) con una solicitud para validar los datos. La validación de datos puede incluir, por ejemplo, asegurar que instanciación de datos particular que se verifica esté dentro de los límites (por ejemplo, límites

de valor mínimo y máximo) y no contenga errores de entrada de datos (por ejemplo: una medición que tenga caracteres no numéricos, puntos decimales desaparecidos o adicionales, o un signo negativo si no es apropiado para los datos; caracteres no válidos en una dirección de correo electrónico, como no tener exactamente un “símbolo arroba”; caracteres no válidos en una dirección postal; etcétera), o que los datos se ajustan de otro modo a una estructura
 5 requerida (por ejemplo, que una medición de la presión arterial incluye tanto una medición sistólica como diastólica). La validación de datos realizada por el módulo administrador de estructura (209) puede considerarse que es una validación autorreferencial, en la medida en que la validación examina los datos particulares de forma aislada, sin
 10 comparar necesariamente los datos que deben ser validados con otros datos. El módulo de descubrimiento e instanciación (207) también puede enviar información que identifica los datos, de modo que el módulo administrador de estructura (209) pueda identificar el conjunto correcto de reglas de validación que se deben aplicar a los datos.

El dispositivo móvil (201) puede incluir además un módulo administrador de reglas (211) que está en contacto comunicativo con el módulo administrador de estructura (209). El módulo administrador de estructura (209) puede
 15 delegar en el módulo administrador de reglas (211) ciertas tareas relacionadas con la validación de reglas. Por ejemplo, cuando el módulo de descubrimiento e instanciación (207) tiene una o más mediciones que necesitan ser validadas, parte de la validación puede implicar un examen más exhaustivo de las mediciones que deben ser validadas, con el fin de determinar si la medición se ajusta a las reglas relacionadas con las relaciones de esta medición con otras mediciones del mismo parámetro o un parámetro diferente. Por ejemplo, se puede examinar una medición de un
 20 monitor continuo de glucosa para ver si no ha cambiado durante un número predeterminado de lecturas consecutivas (lo que puede ser indicativo de un problema del sensor) o si los datos no siguen patrones diarios esperados o no siguen patrones relacionados con otros eventos externos, como comer comidas. La validación de datos realizada por el módulo administrador de reglas (211) puede considerarse que es una validación relativa, en la medida en que la validación examina los datos particulares comparando los datos que deben ser validados con otros datos.

25 El dispositivo móvil (201) puede incluir además un módulo administrador de configuración (213) que está en contacto comunicativo con el programa de aplicación (205). El módulo administrador de configuración (213) proporciona funciones que pueden utilizarse para facilitar el cambio de la estructura de datos o las reglas de validación. Por ejemplo, se puede determinar que la estructura de datos de una serie de mediciones necesita cambiarse (por ejemplo, cambiando los límites mínimo/máximo de lecturas aceptables), o que las reglas necesitan cambiarse (por ejemplo,
 30 incluyendo un componente diario en la relación aceptable entre una medición de presión arterial sistólica y diastólica). El cambio solicitado puede ser introducido en el programa de aplicación (205) por un usuario (por ejemplo, a través del servidor (203)), o puede ser determinado algorítmicamente mediante el programa de aplicación (205) (por ejemplo, como un intento automatizado de reducir falsas alarmas). Se determinará si el cambio implica un cambio en la validación de datos, en la validación de reglas o en ambos. En algunas realizaciones, esta determinación puede
 35 hacerse mediante el programa de aplicación (205) y comunicarse al módulo administrador de configuración (213) junto con el propio cambio. En otras realizaciones, la determinación puede hacerse sola mediante el módulo administrador de configuración (213) a partir del cambio, sin necesidad de que el programa de aplicación (205) haga esta determinación.

40 El módulo administrador de configuración (213) está en contacto comunicativo con el módulo administrador de estructura (209) y con el módulo administrador de reglas (211). El módulo administrador de configuración (213) puede reconocer (ya sea por estar informado o por su propia determinación) si la solicitud implica un cambio en la estructura de datos o en la validación de reglas. Si el cambio se refiere a la estructura de datos, el módulo administrador de configuración (213) se comunicará con el módulo administrador de estructura (209) con el fin de actualizar la estructura
 45 de datos requerida. Si el cambio se refiere a la validación de reglas, el módulo administrador de configuración (213) se comunicará con el módulo administrador de reglas (211) con el fin de actualizar la validación de reglas requerida.

Cuando tiene lugar la validación de datos, puede producirse uno de dos escenarios: Primero, la información puede estar según las reglas de validación; o segundo, la información puede no estar según las reglas de validación. En el
 50 segundo caso, las realizaciones notifican al programa de aplicación (205) que algo estaba equivocado, y también notificarán al programa de aplicación (205) qué estaba equivocado. Las realizaciones pueden devolver uno o más de varios tipos de notificación al programa de aplicación (205), por ejemplo, con el fin de indicar la gravedad (por ejemplo, errores, advertencias), para indicar una presunta causa del error, y así sucesivamente.

55 La FIG. 3A ilustra a un nivel de abstracción de hardware relativamente alto un sistema (300) según una realización de la presente descripción. El Sistema (300) puede ser útil para ilustrar una muestra de flujo de trabajo de recolección y limpieza de datos en la fuente.

El sistema (300) incluye un dispositivo móvil (301) en comunicación con un servidor (303) a través de la red de
 60 comunicación (304). El dispositivo móvil (301) puede incluir una interfaz de comunicación (por ejemplo, Ethernet cableada, Wi-Fi inalámbrico o transceptor Bluetooth, etc.) para comunicarse con la red de comunicación (304), utilizando hardware y protocolos conocidos por los expertos en la materia. El dispositivo móvil (301) puede incluir un

procesador (305) acoplado a una memoria (307). La memoria (307) puede estar configurada para almacenar uno o más programas de aplicación (o aplicación) (309a), (309b), etc. que, cuando son ejecutados por el procesador (305), funcionan para intercambiar información y/o comandos con el servidor (303), por ejemplo, para recibir comandos del servidor (303) con respecto al control de la entrada de usuario, como mediciones de datos u otra información teclada, y para comunicar los datos medidos desde el procesador (305) al servidor (303). Un programa de aplicación individual pero no específico puede denominarse programa de aplicación (309). Se permite que usuario autorizado (302) se comuniquen con el servidor (303) y configure el servidor (303), por ejemplo, definiendo conjuntos de reglas, estructuras de datos y otras descripciones para indicar datos válidos. Tales configuraciones a su vez se comunicarán al dispositivo móvil (301).

10

La red de comunicación (304) puede ser con conmutación de paquetes y/o con conmutación de circuitos. Una red de comunicación ejemplar (304) incluye, sin limitación, una red de área amplia (WAN), como Internet, una red telefónica pública conmutada (PSTN), una red de servicio telefónico antiguo (POTS), una red de comunicaciones celulares o combinaciones de las mismas. En una configuración, la red de comunicación (304) es una red pública que soporta la serie de protocolos TCP/IP.

15

El dispositivo móvil (301) puede incluir además un circuito de interfaz de recepción, por ejemplo, el receptor (311a), que está configurado para recibir mediciones desde un sensor (313). Las mediciones pueden incluir lecturas u otra información que deba ser limpiada. El receptor (311a) puede estar acoplado al procesador (305) para pasar mediciones desde el sensor (313) al procesador (305).

20

El dispositivo móvil (301) puede incluir además un circuito de interfaz de recepción, por ejemplo, el receptor (311b), que está configurado para recibir datos desde un dispositivo de entrada/salida (I/O) de usuario (314) como un teclado, interfaz de pantalla táctil, micrófono para reconocimiento de voz, etcétera. Los datos pueden incluir datos numéricos u otra información que deba ser limpiada. El receptor (311b) puede estar acoplado al procesador (305) con el fin de pasar la I/O de usuario desde el dispositivo de I/O (314) al procesador (305).

25

El procesador (305) puede estar configurado para acceder al contenedor de aplicaciones (315). El contenedor de aplicaciones (315) incluye código de software utilizado para llevar a cabo los procedimientos descritos en este documento, junto con un conjunto de API utilizadas por un proceso de llamada (por ejemplo, padre) para acceder al software. El acceso puede ser por medio de llamadas de función, controladores de eventos o similares, llamados o que se comunican de otro modo con el programa de aplicación (309). Por ejemplo, un manejador de eventos puede detectar la presencia o disponibilidad de nuevos datos leídos a través del receptor (311a) o (311b), y después invocar una API en el contenedor de aplicaciones (315) con el fin de procesar los datos. Después de que se procesan los datos, se pueden suministrar al programa de aplicación (309) para su posterior procesamiento o análisis. El contenedor de aplicaciones (315) puede estar alojado en una memoria dentro del dispositivo móvil (301) (por ejemplo, que incluye la memoria (307) o una memoria diferente), o puede estar alojado de forma remota y accesible al dispositivo móvil (301) por un enlace de comunicación como se ilustra en la FIG. 3B, o alguna combinación de los dos.

30

El contenedor de aplicaciones (315) puede incluir API como la API de descubrimiento e instanciación (317), la API de administrador de estructura (319), la API de administrador de reglas (321) y la API de administrador de configuración (323). La API de descubrimiento e instanciación (317) proporciona acceso programático al módulo de descubrimiento e instanciación (207). La API de administrador de estructuras (319) proporciona acceso programático al módulo administrador de estructuras (209). La API de administrador de reglas (321) proporciona acceso programático al módulo administrador de reglas (211). La API de administrador de configuración (323) proporciona acceso programático al módulo administrador de configuración (213).

40

El servidor (303) puede estar configurado para utilizar el módulo administrador de configuración (213), a través del procesador (305) y la API de administrador de configuración (323), con el fin de cambiar las reglas contenidas dentro del módulo administrador de estructura (209) y/o el módulo administrador de reglas (211). Los cambios pueden incluir, por ejemplo, cambios en el conjunto de reglas o cambios en la descripción de la estructura de datos.

50

La FIG. 3B ilustra a un nivel de abstracción de hardware relativamente alto un sistema (350) según una realización de la presente descripción. En comparación con el sistema (300) de la FIG. 3, el sistema (350) almacena el contenedor de aplicaciones (315) en una memoria (353) que está alejada del dispositivo móvil (351), pero está en contacto de comunicación con el dispositivo móvil (351) a través del enlace de comunicación (355). El dispositivo móvil (351) puede incluir una interfaz de comunicación (por ejemplo, Ethernet cableada, Wi-Fi inalámbrico o transceptor Bluetooth, etc.) con el fin de comunicarse con la red de comunicación (304) y/o la memoria (353), utilizando hardware y protocolos conocidos por expertos en la materia. Otros elementos del sistema (350) se han descrito en el contexto de elementos numerados de manera similar del sistema (300).

55

60

La FIG. 4 ilustra a un proceso (400) para utilizar un sistema (300), según una realización de la presente descripción.

Al menos algunas etapas del proceso (400) pueden ponerse en práctica mediante el servidor (303). El proceso (400) comienza en la etapa (401), en la que un usuario autorizado (302) puede publicar en el servidor (303), y el servidor (303) puede recibir configuraciones para uno o más de: (A) un conjunto de reglas que debe ser implementado por el módulo (211) del administrador de reglas; (B) una descripción de estructura de datos que debe ser implementada por el módulo administrador de estructura (209); y (C) el programa de aplicación (309).

El proceso (400) a continuación pasa a la etapa (403), en la que las estructuras publicadas y las configuraciones de reglas pueden sincronizarse con el dispositivo móvil (301). Tal sincronización puede tener lugar sustancialmente en cualquier momento. Puede producirse una actualización del dispositivo móvil después de que el programa de aplicación (309) ya se esté ejecutando, por ejemplo, si una nueva investigación médica indica que deberían recopilarse tipos adicionales de datos o si debería realizarse una verificación de datos adicional de los datos recopilados (por ejemplo, para compensar un sesgo diurno no reconocido previamente, o para compensar los nuevos medicamentos que el paciente pueda estar tomando, etcétera). En una realización, el servidor (303) puede entonces ensamblar la información publicada en un contenedor de aplicaciones para su transmisión al dispositivo móvil (301). En otra realización, la información publicada puede transferirse al dispositivo móvil (301) durante un proceso de sincronización como el proceso (500) (descrito más adelante). La transferencia de información al dispositivo móvil (301) puede repetirse para datos adicionales o conjuntos de datos que deben ser recopilados. Algunos datos (por ejemplo, una medición de la presión arterial) pueden incluir componentes (por ejemplo, mediciones sistólicas y diastólicas) que pueden tener algún valor individualmente, pero los datos pueden tener más valor como un conjunto que incluya todos los componentes medidos aproximadamente al mismo tiempo. Los conjuntos de datos también pueden incluir mediciones de un mismo proceso pero repetidas a lo largo del tiempo (por ejemplo, un monitor de pulsaciones durante el esfuerzo o un monitor de glucosa durante las comidas).

El proceso (400) a continuación pasa a la etapa (405), en la que el servidor (303) transmitirá entonces la información publicada a un almacenamiento de memoria que es accesible a los programas de aplicación (309). En algunas realizaciones, el almacenamiento de memoria puede estar dentro del dispositivo móvil (301) tal como se ilustra en la FIG. 3A. En otras realizaciones, el almacenamiento de memoria puede ser externo al dispositivo móvil (351) tal como se ilustra en la FIG. 3B. Al concluir el proceso (400), el control puede transferirse al proceso (500) de la FIG. 5.

La FIG. 5 ilustra a un proceso (500) para utilizar un sistema (300), según una realización de la presente descripción. Al menos algunas etapas del proceso (500) pueden ponerse en práctica mediante el dispositivo móvil (301) y/o el programa de aplicación (309) dentro del dispositivo móvil (301). Aunque el proceso (500) se describe en referencia al almacenamiento del contenedor de aplicaciones (315) dentro del dispositivo móvil (301), los expertos en la materia reconocerán cómo adaptar el proceso (500) para que incluya el almacenamiento fuera del dispositivo móvil (301). El proceso (500) comienza en la etapa (501), en la que un dispositivo móvil (301) recibe el contenedor de aplicaciones (315) desde el servidor (303). El contenedor de aplicaciones (315) incluye las API utilizadas para acceder al software en el contenedor de aplicaciones (315).

El proceso (500) a continuación pasa a la etapa (503), en la que el contenedor recibido en la etapa (501) es almacenado en una memoria accesible para el programa de aplicación (309) dentro del dispositivo móvil (301).

El proceso (500) a continuación pasa a la etapa (505), en la que se pone en marcha un programa de aplicación (309). Las API individuales dentro del contenedor de aplicaciones (315) están disponibles para su uso por los programas de aplicación (309) con el fin de crear estructuras de datos, crear y actualizar datos, y validar datos con el fin de ayudar a garantizar datos de mayor calidad. Por ejemplo, un desarrollador de software de un programa de aplicación (309) o un usuario autorizado (302) podría utilizar la descripción de la estructura de datos y el conjunto de reglas, accesibles a través de la API de administrador de estructura (319) y la API de administrador de reglas (321), con el fin de desarrollar o utilizar un tipo específico de aplicación de mHealth (por ejemplo, una aplicación de diabetes). El uso del contenedor de aplicaciones (315) ayuda a garantizar que el programa de aplicación (309) pueda administrar de manera fiable la coherencia de datos y la calidad de datos de la misma manera que otras aplicaciones similares. Las API individuales forman una biblioteca de interfaces funcionales que pueden ser utilizadas por múltiples programas de aplicación (309) con el fin de proporcionar uniformidad de la experiencia de usuario y reducir el tiempo de desarrollo a través de los programas de aplicación (309).

El contenedor de aplicaciones (315) puede ser una aplicación genérica (es decir, software para ejecutar en el dispositivo móvil (301), que no está directamente relacionado con la recopilación y limpieza de datos en la fuente), que contiene la API de datos configurables, la API de administrador de reglas (321) y (opcionalmente) componentes de reglas. Los componentes de reglas se refieren a reglas reales (por ejemplo, sistólica < 190), mientras que la API de reglas se refiere al conjunto de procedimientos utilizados para realizar la validación. El contenedor de aplicaciones no necesita tener conocimiento previo del contenido o la configuración de la estructura de datos y las reglas asociadas que hacen cumplir las API, es decir, no hay una codificación fija de la estructura de datos y las reglas asociadas dentro del programa de aplicación (309). Esto permite que el programa de aplicación (309) se diseñe y utilice para administrar

sustancialmente cualquier tipo de configuración de aplicación.

El proceso (500) a continuación pasa a la etapa (507), en la que el programa de aplicación (309) puede sincronizarse con el servidor (303) con el fin de regenerar la estructura de datos y las reglas de validación utilizadas por el programa de aplicación (309). Por ejemplo, la regeneración puede ser necesaria cuando el programa de aplicación (309) se ejecuta por primera vez, si un contenedor de aplicaciones (315) ya no reside dentro del dispositivo móvil (301). Si un contenedor de aplicaciones (315) ya es residente con el fin de dar servicio al programa de aplicación (309a), entonces puede que no sea necesario instalar otro contenedor de aplicaciones (315) para manejar el programa de aplicación (309b). Si el contenedor de aplicaciones (315) ya es residente, entonces el programa de aplicación (309) puede acceder y actualizar la configuración de la estructura de datos, la configuración del conjunto de reglas y la información de datos de usuario para el tipo de aplicación del programa de aplicación (309).

El proceso (500) a continuación pasa a la etapa (509), en la que el programa de aplicación (309) puede entonces inicializarse e inicializa el módulo administrador de reglas (211) y el módulo administrador de estructura (209) con las configuraciones recibidas desde el servidor (303).

El proceso (500) a continuación pasa a la etapa (509), en la que el programa de aplicación (309) puede entonces ejecutarse normalmente, según la configuración recibida, y acepta información de un usuario, por ejemplo, a través del sensor (313) o la I/O de usuario (314). Después de que el programa de aplicación (309) crea objetos de la estructura de datos utilizando la API de descubrimiento e instanciación (317), los objetos posteriores se utilizarán para validar la información de usuario mediante el uso de la API de administrador de estructura (319) y la API de administrador de reglas (321).

Cuando se entrega una configuración de estructura de datos o una configuración de conjunto de reglas actualizada al dispositivo móvil (301), la API de administrador de estructura (319) y la API de administrador de reglas (321) pueden inicializarse para soportar un nuevo comportamiento. Por lo tanto, las realizaciones proporcionan recopilación de datos de alta calidad en una fuente sin cambiar el código fuente de una aplicación. Lo único que necesita cambiarse es la información de configuración asociada con la aplicación.

Las realizaciones según la presente descripción son útiles para las organizaciones de recopilación de datos que, de otro modo, pueden gastar una gran cantidad de recursos para limpiar los datos porque los datos recopilados en la fuente no son fiables sin la práctica de las realizaciones descritas en este documento. Las realizaciones ayudarán a los usuarios a reducir el coste de la limpieza de datos y mejorar la coherencia con la que los programas de aplicación que dependen de datos procesables pueden recopilar y administrar datos en una fuente de datos para diferentes tipos de información (por ejemplo, tipos de datos). Por ejemplo, para algunas aplicaciones relacionadas con la atención sanitaria, la información se necesita sustancialmente en tiempo real para análisis y participación del paciente. Las realizaciones según la presente descripción pueden permitir que los datos se recopilen en una fuente, lo que puede proporcionar un proceso más racionalizado para hacer que los datos estén disponibles.

Además, las realizaciones según la presente descripción facilitan la administración y el seguimiento del uso de la aplicación y la información sanitaria para una mejor atención al paciente a través de múltiples aplicaciones y múltiples grupos de usuarios.

Las realizaciones de la presente invención incluyen un sistema que tiene una o más unidades de procesamiento acopladas a una o más memorias. La una o más memorias pueden estar configuradas para almacenar software que, cuando es ejecutado por una o más unidades de procesamiento, permite la práctica de la invención, al menos mediante el uso de los procesos descritos en este documento, que incluyen al menos en las FIGS. 4-5, y el texto relacionado.

Los procedimientos descritos pueden implementarse fácilmente en software, como utilizando entornos de desarrollo de programas de objetos u orientados a objetos que proporcionan código fuente portátil que puede utilizarse en una diversidad de plataformas informáticas o de estaciones de trabajo. Alternativamente, el sistema descrito puede implementarse parcial o completamente en hardware, como utilizando circuitos lógicos estándar o diseño VLSI. Que pueda utilizarse software o hardware para implementar los sistemas según diversas realizaciones de la presente descripción puede depender de diversas consideraciones, como los requisitos de velocidad o eficiencia del sistema, la función particular y los sistemas de software y hardware particulares que se estén utilizando.

Aunque lo anterior está dirigido a realizaciones de la presente invención, pueden concebirse otras realizaciones o realizaciones adicionales de la presente invención sin apartarse del alcance básico de la misma. Se entiende que diversas realizaciones descritas en este documento pueden utilizarse en combinación con cualquier otra realización descrita, sin apartarse del alcance contenido en este documento. Además, la descripción anterior no pretende ser exhaustiva o limitar la invención a la forma precisa descrita. Son posibles modificaciones y variaciones a la luz de las enseñanzas anteriores o pueden adquirirse a partir de la práctica de la invención. Ciertas realizaciones ejemplares

pueden identificarse mediante el uso de una lista abierta que incluye una redacción para indicar que los artículos de la lista son representativos de las realizaciones y que la lista no pretende representar una lista cerrada exclusiva de realizaciones adicionales. Tal redacción puede incluir "p. ej.", "etc.", "como", "por ejemplo", "etcétera", "y similares", etc., y otra redacción como resultará evidente del contexto circundante.

- 5 Ningún elemento, acto o instrucción utilizado en la descripción de la presente solicitud debería interpretarse como crítico o esencial para la invención, a menos que se describa explícitamente como tal. Además, tal como se usa en este documento, el artículo "un/una" pretende incluir uno o más artículos. Cuando se pretende únicamente un artículo, se utiliza el término "un/uno" o un lenguaje similar. Además, los términos "cualquiera de" seguidos por un listado de
- 10 una pluralidad de artículos y/o una pluralidad de categorías de artículos, tal como se utiliza en este documento, pretenden incluir "cualquiera de", "cualquier combinación de", "cualquier múltiplo de" y/o "cualquier combinación de múltiplos de" los artículos y/o las categorías de artículos, individualmente o junto con otros artículos y/u otras categorías de artículos.
- 15 Asimismo, las reivindicaciones no deberían considerarse limitadas al orden o los elementos descritos a menos que se indique a ese efecto. Además, el uso del término "significa" en cualquier reivindicación pretende acogerse al título 35 del U.S.C. §112, ¶ 6, y no se pretende lo mismo para ninguna reivindicación sin la palabra "significa".

REIVINDICACIONES

1. Un dispositivo móvil para limpiar datos, que comprende:
 - 5 un receptor (311a, 311b) para recopilar datos electrónicos para limpiar;

un procesador (305) acoplado al receptor (311a, 311b), el procesador (305) configurado para recibir los datos recopilados por el receptor y configurado además para recibir un contenedor de aplicaciones (315) que incluye una primera interfaz de programación de aplicaciones (317) y una segunda interfaz de programación de aplicaciones (323)

10 desde un servidor (303);

una memoria (307) acoplada al procesador (305), la memoria (307) configurada para almacenar un programa de aplicación (309a, 309b), un archivo de configuración y el contenedor de aplicaciones (315) que incluye la primera interfaz de programación de aplicaciones (317) y la segunda interfaz de programación de aplicaciones (323); donde el

15 archivo de configuración comprende reglas de validación y estructuras de datos y el contenedor de aplicaciones aloja y soporta el uso de varias configuraciones de aplicaciones;

donde la primera interfaz (317) proporciona acceso programático a un módulo de instanciación (207), el módulo de instanciación (207) es utilizado por el programa de aplicación (309a, 309b) para construir en la memoria (307)

20 objetos requeridos por la primera interfaz (317), pudiendo operar el programa de aplicación (309a, 309b) para utilizar la funcionalidad de descubrimiento del módulo de instanciación (207) para aprender una estructura de datos de los datos recopilados según una configuración proporcionada por la primera interfaz (317);

donde la segunda interfaz (323) proporciona acceso programático a un módulo administrador de configuración (213), el módulo administrador de configuración (213) configurado para actualizar las estructuras de datos y las reglas de validación del archivo de configuración, donde el archivo de configuración es utilizado por el módulo de instanciación para procesar datos, y donde el código fuente del programa de aplicación no cambia,

25 donde la primera interfaz (317) y la segunda interfaz (323) son invocables desde el programa de aplicación (309a, 309b) para limpiar los datos recopilados por el receptor; por lo cual la limpieza de datos incluye la identificación y corrección o eliminación de datos identificados.

30
2. El dispositivo móvil según la reivindicación 1, donde el módulo de instanciación (207) y el módulo administrador de configuración (213) están alojados por el dispositivo móvil.

35
3. El dispositivo móvil según la reivindicación 1, donde al menos uno del módulo de instanciación (207) y el módulo administrador de configuración (213) están alojados externamente al dispositivo móvil.
4. El dispositivo móvil según la reivindicación 1, donde la primera interfaz (317) y la segunda interfaz (323)

40 comprenden una interfaz de programa de aplicación (API) respectiva.
5. El dispositivo móvil según la reivindicación 1, que comprende además un módulo administrador de estructura (209) acoplado al módulo de instanciación (207), el módulo administrador de estructura (209) configurado para validar los objetos.

45
6. El dispositivo móvil según la reivindicación 5, que comprende además un módulo administrador de reglas (211) acoplado al módulo de instanciación (207), el módulo administrador de reglas (211) configurado para validar datos procesados por el módulo de instanciación (207), mediante el uso de reglas para comparar un dato procesado por el módulo de instanciación (207) en relación con otro dato.

50
7. El dispositivo móvil según la reivindicación 6, donde el módulo administrador de estructura (209) y el módulo administrador de reglas (211) son configurables sin cambios en el programa de aplicación (309a, 309b).
8. El dispositivo móvil según la reivindicación 7, donde los cambios de configuración en al menos uno del

55 módulo administrador de estructura (209) y el módulo administrador de reglas (211) tienen efecto sin reiniciar el programa de aplicación (309a, 309b).
9. El dispositivo móvil según la reivindicación 7, donde los cambios de configuración en al menos uno del

60 módulo administrador de estructura (209) y el módulo administrador de reglas (211) se determinan algorítmicamente.
10. El dispositivo móvil según la reivindicación 6, donde el módulo administrador de estructura (209) y el módulo administrador de reglas (211) operan sustancialmente en tiempo real para validar datos.

11. Un procedimiento para limpiar datos, que comprende:

5 proporcionar un dispositivo móvil (301) que comprende un procesador (305) acoplado a una memoria (307), el procesador (305) configurado para recibir un contenedor de aplicaciones (315) que incluye una primera interfaz de programación de aplicaciones (317) y una segunda interfaz de programación de aplicaciones (323) desde un servidor (303), la memoria (307) configurada para almacenar un programa de aplicación (309a, 309b), un archivo de configuración y un contenedor de aplicaciones (315) que incluye la primera interfaz de programación de aplicaciones (317) y la segunda interfaz de programación de aplicaciones (323), donde el archivo de configuración comprende
10 reglas de validación y estructuras de datos y el contenedor de aplicaciones aloja y soporta el uso de varias configuraciones de aplicaciones;

recopilar, mediante un receptor (311a, 311b) acoplado a un procesador (305), datos electrónicos para limpiar;

15 donde la primera interfaz (317) proporciona acceso programático a un módulo de instanciación (207); el módulo de instanciación (207) es utilizado por el programa de aplicación (309a, 309b) para construir en la memoria (307) objetos requeridos por la primera interfaz (317), pudiendo operar el programa de aplicación (309a, 309b) para utilizar la funcionalidad de descubrimiento del módulo de instanciación (207) para aprender una estructura de datos de los datos recopilados según una configuración proporcionada por la primera interfaz (317);
20

actualizar las estructuras de datos y las reglas de validación del archivo de configuración, mediante el uso de la segunda interfaz para ejecutar un módulo administrador de configuración por parte del procesador, donde el archivo de configuración es utilizado por el módulo de instanciación para procesar datos, y donde el código fuente del programa de aplicación no cambia,
25

30 donde la primera interfaz y la segunda interfaz son invocables desde el programa de aplicación (309a, 309b) para limpiar los datos recopilados por el receptor (311a, 311b); por lo cual la limpieza de datos incluye la identificación y corrección o eliminación de datos identificados.

12. El procedimiento según la reivindicación 11, donde el módulo de instanciación y el módulo administrador de configuración son alojados por el dispositivo móvil.

13. El procedimiento según la reivindicación 11, donde al menos uno del módulo de instanciación (207) y el módulo administrador de configuración están alojados externamente al dispositivo móvil.
35

14. El procedimiento según la reivindicación 11, donde la etapa de procesar datos recopilados por el receptor comprende validar los objetos.

15. El procedimiento según la reivindicación 14, donde la etapa de procesar datos recopilados por el receptor (311a, 311b) comprende además validar datos mediante el uso de reglas para comparar un dato en relación con otro dato.
40

FIG. 1100

```

{"person":{
  "type":"structure",
  "rules" {
    "pre-rules": "<string with rules>",
    "post-rules": "<string with rules>",
    "val-rules": "<string with rules>"
  }
  "content":{
    {"type":"integer",
     "rules" {
       "pre-rules": "<string with rules>",
       "post-rules": "<string with rules>",
       "val-rules": "<string with rules>"
     },
     "name":"ID",
     "mandatory":"true"
    ...
  }
  {"type":"string",
   "rules": {
     ...
   }
   "name":"First Name"
  ...
  }
}
} .

```


FIG. 2

200

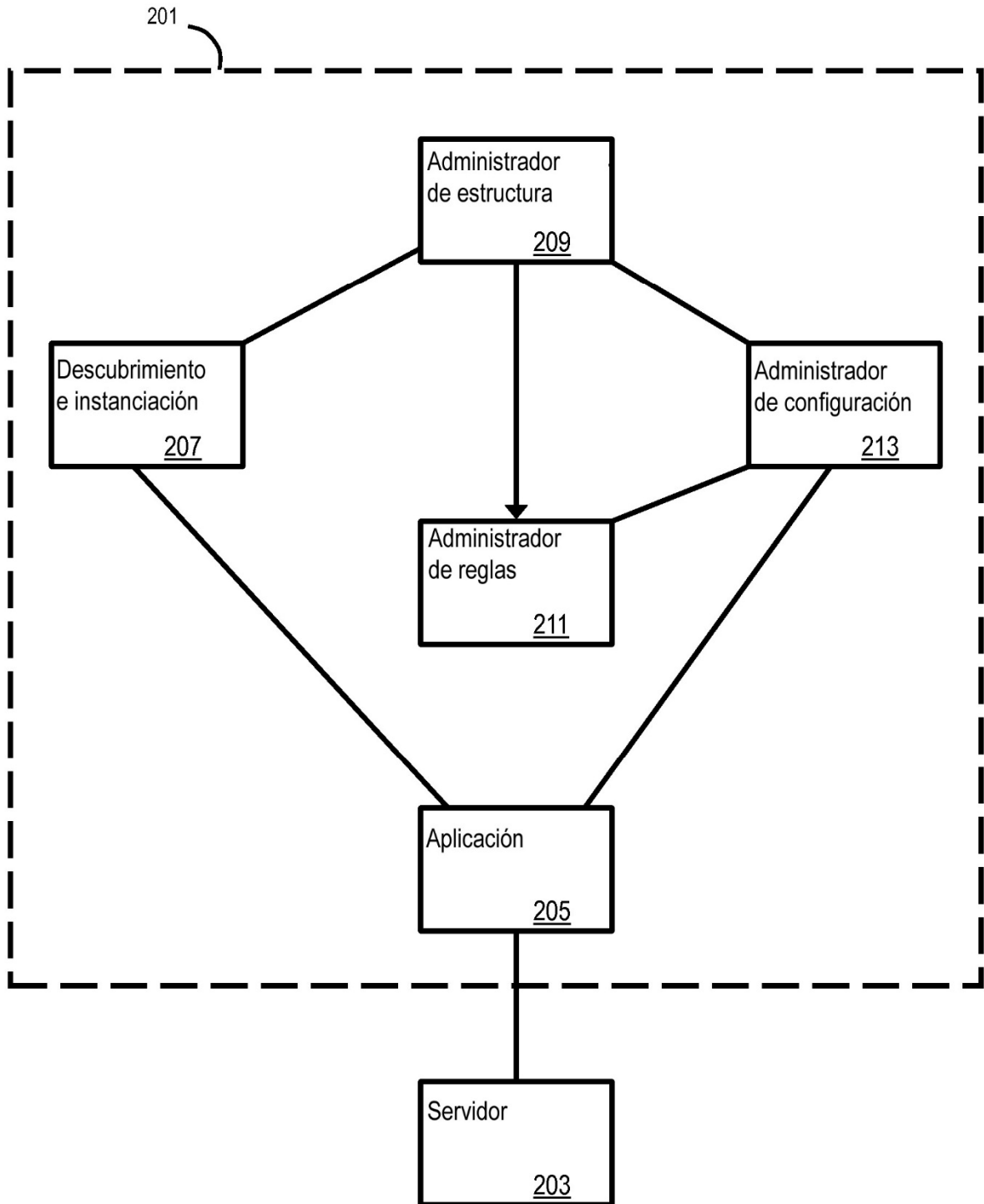


FIG. 3A

300

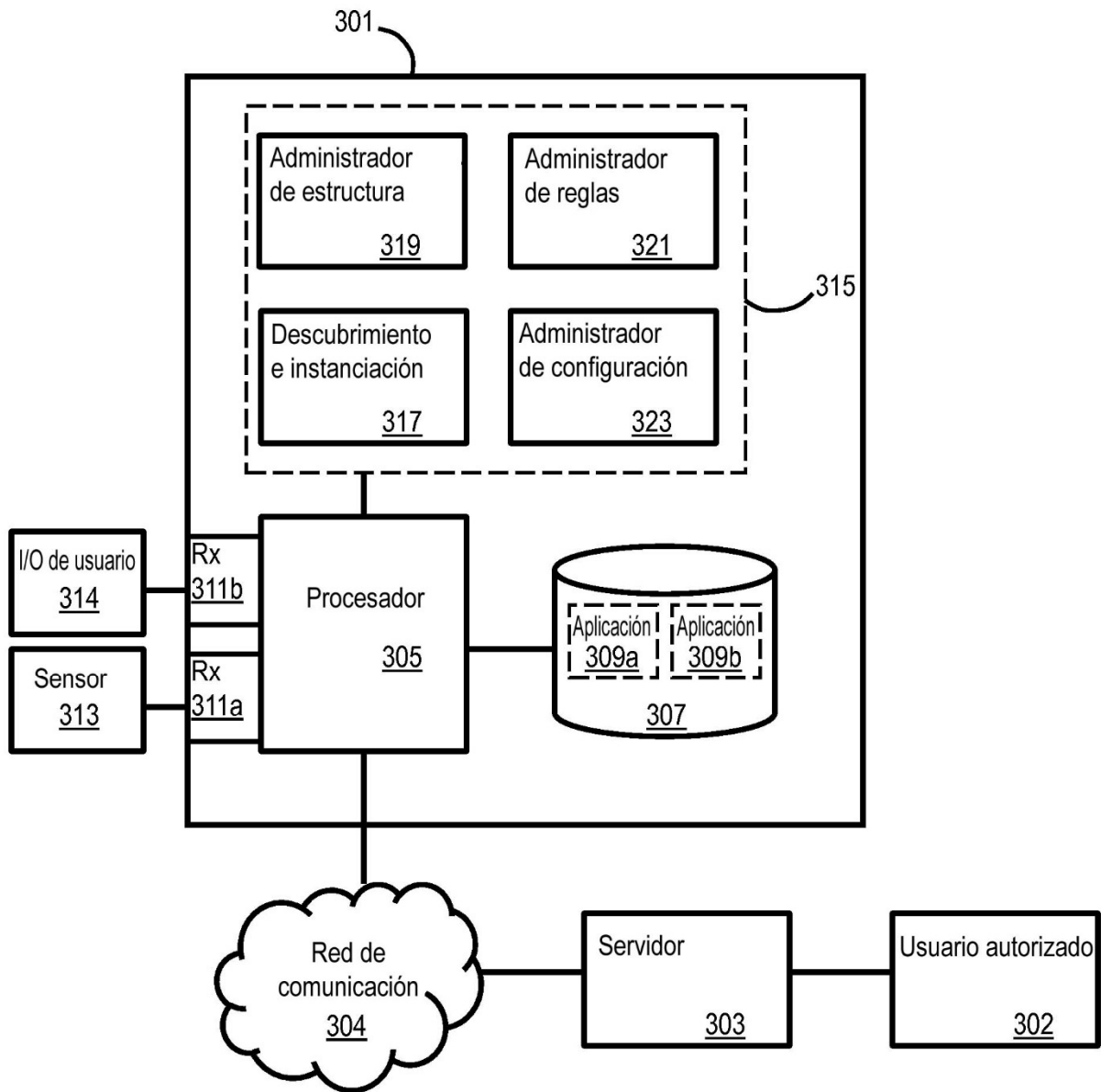


FIG. 3B

350

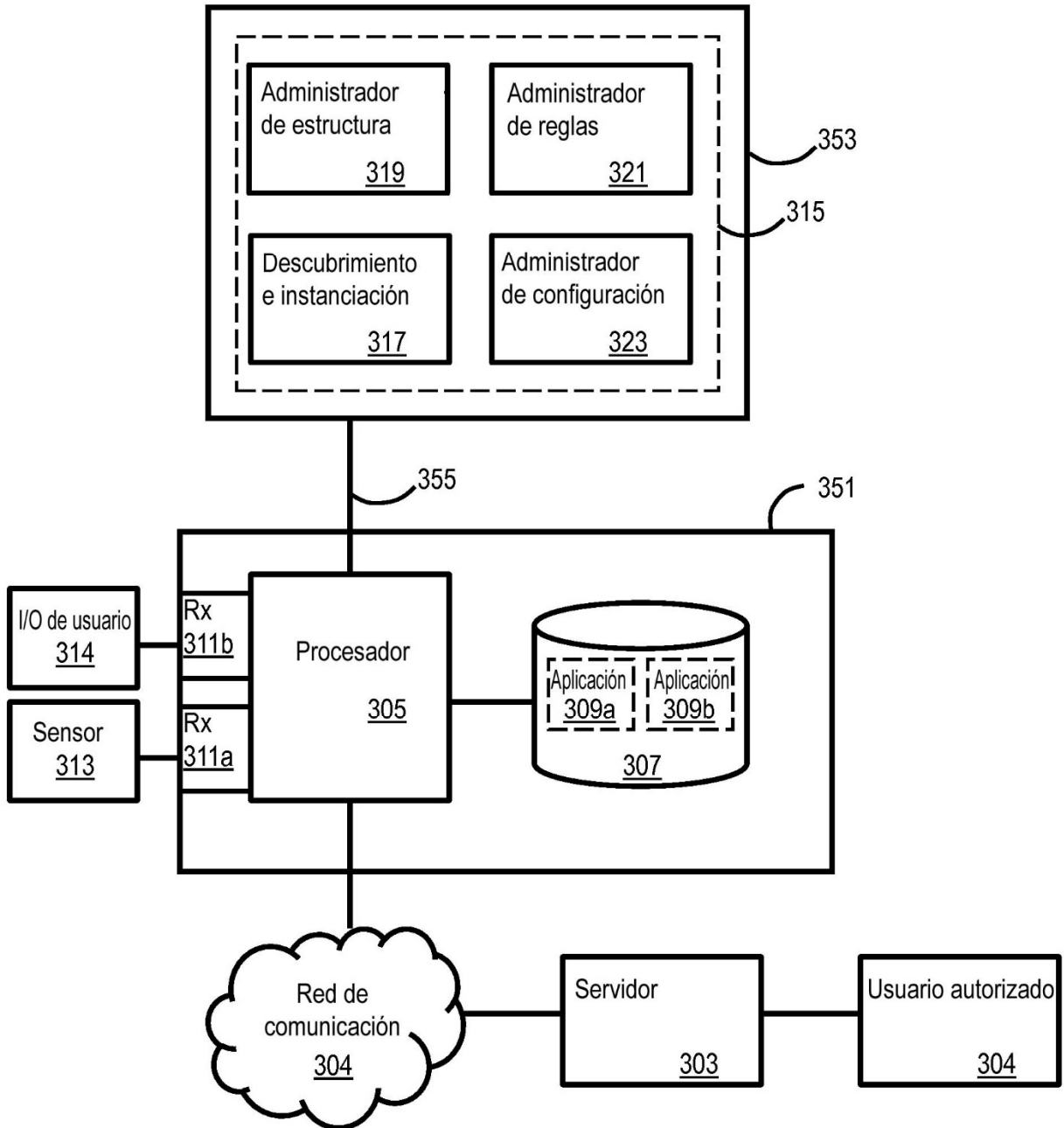


FIG. 4

400

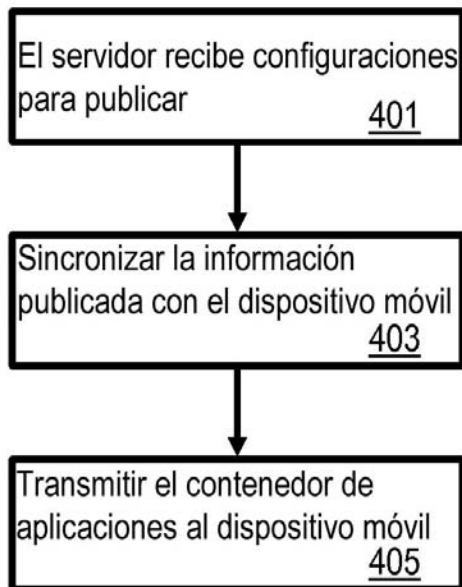


FIG. 5

500

