

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 716 820**

51 Int. Cl.:

G06F 9/48

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **11.10.2011 PCT/US2011/055826**

87 Fecha y número de publicación internacional: **21.03.2013 WO13039533**

96 Fecha de presentación y número de la solicitud europea: **11.10.2011 E 11872484 (8)**

97 Fecha y número de publicación de la concesión europea: **26.12.2018 EP 2756389**

54 Título: **Gestión de procesos en estados de suspensión y estados de ejecución**

30 Prioridad:

12.09.2011 US 201113230677

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

17.06.2019

73 Titular/es:

MICROSOFT TECHNOLOGY LICENSING, LLC

(100.0%)

One Microsoft Way

Redmond, WA 98052, US

72 Inventor/es:

SINGH, NEERAJ KUMAR;

PULAPAKA, HARI;

KISHAN, ARUN y

SCHWARTZ JR., JAMES A.

74 Agente/Representante:

ARIAS SANZ, Juan

ES 2 716 820 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

Gestión de procesos en estados de suspensión y estados de ejecución.

Antecedentes

5 Muchos dispositivos informáticos, tales como ordenadores de sobremesa, ordenadores portátiles, teléfonos inteligentes y tabletas, se desarrollan para ahorrar el consumo de energía, mejorar la latencia de arranque del dispositivo y las aplicaciones, y mejorar la capacidad de respuesta en general. Por ejemplo, cuando un dispositivo no se usa durante algún tiempo, tal como diez minutos, el dispositivo se puede poner en un estado de reposo que puede reducir significativamente la actividad de las aplicaciones y, de este modo, puede dar como resultado una mejora del consumo de energía. Cuando un usuario intenta usar el dispositivo posteriormente, el dispositivo se despierta de este estado de reposo para proporcionar al usuario todas las capacidades del dispositivo.

10 En David Quintana: "Mobile Multitasking", <http://davidquintana.com/entry/mobile-multitasking>, 2010, se discuten planteamientos de multitarea en sistemas operativos para dispositivos móviles. Las aplicaciones se suspenden cuando ya no son visibles por el usuario o cuando se mueven a un segundo plano. Permanecen en la memoria pero están congeladas.

15 El documento US 2006/225107 A1 describe un sistema para decodificadores. El sistema usa un entorno de ejecución basado en intérpretes con un gestor de aplicaciones respectivo. El gestor de aplicaciones detiene las aplicaciones actuales cuando otra aplicación presenta una interfaz de usuario e interfiere con la aplicación actual.

20 El documento US 2009/113444 A1 describe un método para la gestión de aplicaciones, en donde las aplicaciones ejecutadas se clasifican según uno o más criterios que cambian en respuesta a la interacción del usuario con la aplicación. Para terminar una aplicación que responde a las condiciones de la memoria, la información del estado de aplicación se puede guardar y usar posteriormente para reactivar la aplicación terminada.

El documento WO 01/63401 A2 se refiere a la utilización de recursos en un cliente compartido, que se basa en la gestión de sesiones por un gestor de sesión que mantiene una base de datos de correspondencias entre mecanismos de autenticación para usuarios y sesiones individuales.

25 Marshall Kirk Mckusick et al, "Process Management", The Design and Implementation of the 4.4BSD Operating System – Chapter 2 Design Overview of 4.4BSD, 2011, se refiere a la gestión de procesos en el sistema operativo BSD.

Compendio

30 La presente invención se define mediante un método y un sistema para suspender uno o más procesos asociados con una aplicación como se define en las reivindicaciones independientes. Las realizaciones preferidas se definen en las reivindicaciones dependientes.

35 Este Compendio se proporciona para introducir una selección de conceptos de una forma simplificada que se describen aún más a continuación en la Descripción Detallada. Este Compendio no está destinado a identificar factores clave o características esenciales de la materia objeto reivindicada, ni tampoco se pretende que sea usado para limitar el alcance de la materia objeto reivindicada.

Entre otras cosas, se tratan en la presente memoria uno o más sistemas y/o técnicas para suspender uno o más procesos asociados con una aplicación, determinar si reanudar un proceso suspendido en base a una o más políticas de activación, y/o gestionar un estado de aplicación de una aplicación.

40 Un entorno informático puede alojar aplicaciones que están asociadas con procesos, que se pueden ejecutar por uno o más procesadores como subprocesos. Se puede apreciar que uno o más sistemas y/o métodos descritos en la presente memoria pueden ser aplicables a aplicaciones, procesos y/o subprocesos (por ejemplo, dado que los procesos se componen de subprocesos que ejecutan código en uno o más procesadores, las técnicas y/o sistemas mencionados en la presente memoria se aplican igualmente a los subprocesos). Debido a que el entorno informático puede alojar una pluralidad de aplicaciones y/o procesos, se puede implementar una jerarquía de contenedores lógicos para gestionar dichas aplicaciones y/o procesos. En particular, las aplicaciones se pueden asignar a contenedores lógicos, tales como un contenedor lógico raíz, dentro de la jerarquía de contenedores lógicos. De manera similar, los procesos se pueden asignar a contenedores lógicos. La jerarquía de contenedores lógicos puede permitir el anidamiento jerárquico de aplicaciones, procesos y/o contenedores lógicos relacionados. Por ejemplo, una aplicación de editor de textos (por ejemplo, una aplicación/proceso principal) se puede asignar a un contenedor lógico raíz. La aplicación de editor de textos puede estar asociada con uno o más procesos secundarios, tales como un proceso de impresión de editor de textos, un proceso de guardado de editor de textos y/o un proceso de revisión ortográfica. Por consiguiente, el proceso de impresión de editor de textos, el proceso de guardado de editor de textos y/o el proceso de revisión ortográfica, por ejemplo, se pueden asignar a contenedores lógicos asociados con (por ejemplo, anidados dentro de) el contenedor lógico raíz asignado a la aplicación de editor de textos. De esta

forma, puede ser identificable una asociación (por ejemplo, una relación) entre la aplicación de editor de textos y/o los procesos secundarios.

5 En un ejemplo de suspensión uno o más procesos asociados con una aplicación como se proporciona en la presente memoria, se puede recibir una notificación de que una aplicación está inactiva (por ejemplo, la aplicación puede no ser visible por un usuario, el usuario puede no haber interactuado con la aplicación dentro de un tiempo predeterminado, la aplicación puede estar en un estado de segundo plano, etc.). Se pueden identificar uno o más procesos asociados con la aplicación. En un ejemplo, se puede consultar una jerarquía de contenedores lógicos para identificar el uno o más procesos. Por ejemplo, se puede identificar un contenedor lógico de raíz asignado a la aplicación. Se puede identificar un primer proceso asignado a un primer contenedor lógico asociado con (por ejemplo, anidado dentro de) el contenedor lógico de raíz. Se puede identificar un segundo proceso asignado al primer contenedor lógico asociado con (por ejemplo, anidado dentro de) el contenedor lógico de raíz. Se puede identificar un tercer proceso asignado a un segundo contenedor lógico asociado con (por ejemplo, anidado dentro de) el primer contenedor lógico. De esta forma, el primer, segundo y tercer proceso se pueden identificar como que están asociados con la aplicación. Se puede apreciar que una variedad de procesos y/o aplicaciones se pueden identificar como que están asociados con la aplicación. El uno o más procesos asociados con la aplicación se pueden poner en un estado de suspensión para ahorrar el consumo de energía haciendo que los procesos suspendidos no sean elegibles para la programación de ejecución del procesador. Una vez suspendidos, los subprocesos y/o procesos recién creados asociados con la aplicación suspendida (por ejemplo, los procesos recién creados asignados a los contenedores lógicos anidados dentro del contenedor lógico de raíz) se pueden suspender para mantener la consistencia de la suspensión en todos los procesos asociados con la aplicación suspendida. De esta forma, los procesos actualmente asociados con la aplicación y/o los procesos recién creados se pueden identificar como que están asociados con la aplicación, y se pueden suspender en consecuencia. Se puede apreciar que la suspensión de uno o más procesos que se proporcionan en la presente memoria pueden ocurrir cuando el dispositivo está o no en modo de reposo, pero donde una o más aplicaciones están inactivas o no se están usando. De este modo, se puede ahorrar energía y/o vida útil de batería extendida incluso cuando un dispositivo está en uso.

En un ejemplo de determinación de si reanudar un proceso suspendido como se proporciona en la presente memoria, se puede mantener un conjunto de políticas de activación asociadas con un proceso suspendido. Una política de activación puede ser indicativa de una razón para reanudar el proceso suspendido a un estado de ejecución. En un ejemplo, una política de activación puede corresponder a una situación donde una aplicación solicitante envía una llamada de comunicación entre procesos a un proceso suspendido. En otro ejemplo, una política de activación puede corresponder a una situación donde una aplicación suspendida asociada con el proceso suspendido pasa a un estado de primer plano. En otro ejemplo, una política de activación puede corresponder a una situación donde se detecte una entrada de usuario asociada con la aplicación suspendida. Se puede apreciar que una política de activación puede corresponder a una variedad de situaciones y/o razones para activar el proceso suspendido. La política de activación se puede asociar con un contador indicativo de una serie de notificaciones de activación (por ejemplo, una notificación de que ha ocurrido un evento asociado con una política de activación, tal como una aplicación solicitante que envía comunicación entre procesos al proceso suspendido) asociadas con la política de activación que están actualmente pendientes para el proceso suspendido. El contador se puede aumentar (por ejemplo, puede estar pendiente un evento asociado con una notificación de activación) y/o disminuir (por ejemplo, se puede completar un evento asociado con una notificación de activación). Por ejemplo, un contador para una política de activación de comunicación entre procesos se puede aumentar a 1 tras recibir una primera notificación de activación de que una aplicación de hoja de cálculo envió una llamada de comunicación entre procesos a un editor de textos suspendido. El contador de la política de activación de comunicación entre procesos se puede aumentar a 2 tras recibir una segunda notificación de activación de que una aplicación de navegador envió una llamada de comunicación entre procesos al editor de textos suspendido.

El conjunto de políticas de activación se puede evaluar para determinar si poner el proceso suspendido en un estado de ejecución o retener el proceso suspendido en el estado de suspensión. En un ejemplo, el proceso suspendido se puede poner en el estado de ejecución en base a la determinación de que al menos un contador asociado con una política de activación comprende un valor por encima de un valor umbral (por ejemplo, al menos un contador tiene un valor por encima de 2). En otro ejemplo, los valores de uno o más contadores se pueden agregar juntos para crear un valor agregado (por ejemplo, tres contadores asociados con diversas políticas de activación pueden tener un valor de 1 y, de este modo, se puede determinar un valor agregado de 3). El valor agregado se puede ajustar en base a las ponderaciones asociadas con las políticas de activación (por ejemplo, una política de activación asociada con una entrada de usuario puede recibir un valor de ponderación de 2, mientras que una política de activación asociada con una solicitud de un tercero remoto puede recibir un valor de ponderación de 5 debido a que puede ser más ventajoso responder agresivamente a un usuario a diferencia de a un tercero remoto). Del mismo modo, un contador particular se puede observar incondicionalmente (por ejemplo, un contador de usuario), mientras que otro contador se puede observar condicionalmente (por ejemplo, en base a la frecuencia, ruido, fuente, etc.). Si el proceso suspendido se pone en el estado de ejecución como una aplicación en ejecución, entonces se puede evaluar el estado actual de las políticas de activación y/o el tiempo transcurrido desde la terminación de una notificación de activación en la determinación de si y/o cuándo volver a poner el proceso de ejecución en el estado de suspensión. Por ejemplo, el proceso de ejecución se puede retener en el estado de ejecución durante un período de tiempo en el caso de que se realicen llamadas de comunicación entre procesos adicionales. Como una

optimización, las notificaciones se pueden entregar en transiciones “de borde” (0-> 1, 1 -> 0) para un contador en particular. En el primer caso, se evalúa la política para la ejecución y en el segundo caso se evalúa la política para la suspensión.

5 Se puede apreciar que en un ejemplo, el conjunto de políticas de activación se puede asociar con un contenedor lógico asignado al proceso suspendido dentro de una jerarquía de contenedores lógicos. Por consiguiente, si el proceso suspendido se pone en el estado de ejecución, entonces uno o más procesos y/o aplicaciones asociadas con el proceso suspendido (por ejemplo, procesos asignados a contenedores lógicos asociados con un contenedor lógico asignado al proceso suspendido) se puede poner en el estado de ejecución.

10 En un ejemplo de gestión de un estado de aplicación como se proporciona en la presente memoria, se puede mantener una lista de temporizadores asociados con una aplicación. Por ejemplo, se pueden mantener temporizadores absolutos (por ejemplo, un tiempo de espera establecido a las 2:00 PM el 31/8/11), temporizadores relativos (por ejemplo, un tiempo de espera establecido en 10 minutos), y/u otros tipos de temporizadores. Tras recibir una notificación de que la aplicación ha de ser suspendida, se puede crear un estado de aplicación que comprenda los datos de rebasado de temporizador derivados de la lista de temporizadores. El estado de aplicación se puede crear debido a que ciertos temporizadores, tales como los temporizadores relativos, pueden expirar mientras la aplicación se suspende aún cuando no haya ocurrido la razón subyacente para el temporizador. Por ejemplo, una aplicación de editor de textos puede establecer un temporizador relativo de 10 minutos para un proceso de revisión ortográfica debido a que el proceso de revisión ortográfica generalmente no tarda más de 10 minutos en completarse, a menos que haya ocurrido un error. No obstante, el proceso de revisión ortográfica se puede suspender después de 2 minutos desde la creación del temporizador relativo, y se puede suspender durante 15 minutos. Sin ajustar el temporizador relativo, el temporizador relativo puede expirar (por ejemplo, el tiempo de espera puede ser incorrecto debido a que el tiempo de espera no está relacionado con un error de revisión ortográfica, sino simplemente con una suspensión de la revisión ortográfica). Debido a que la revisión ortográfica no ha errado, sino que simplemente está suspendido, se puede rebasar el temporizador relativo usando datos relativos de rebasado de temporizador, tales como un tiempo hasta el valor de expiración, dentro del estado de aplicación (por ejemplo, el estado de aplicación puede comprender datos relativos de rebasado de temporizador que comprenden un tiempo hasta el valor de expiración que indica un lapso de tiempo desde cuando el temporizador relativo se suspende a un tiempo de expiración original del temporizador relativo). Además, los temporizadores relativos se pueden ajustar de modo que, tras reanudar una aplicación suspendida, no ocurra una abundancia de expiraciones de temporizadores que, de otro modo, podrían causar un consumo innecesario de CPU y/o podrían disminuir la experiencia de usuario. Por el contrario, los temporizadores absolutos se pueden asociar con un tiempo absoluto (por ejemplo, un tiempo de expiración original), de manera que una aplicación puede desear ser notificada cuando ocurre el tiempo absoluto independientemente de si una aplicación asociada con el temporizador fue suspendida. Por consiguiente, el temporizador absoluto se puede rebasar usando datos de rebasado de temporizador absoluto, tal como un tiempo de expiración original, dentro del estado de aplicación (por ejemplo, el estado de aplicación puede comprender datos de rebasado de temporizador absoluto que comprenden el tiempo de expiración original). De esta forma, el estado de aplicación que comprende datos de rebasado de temporizador se pueden usar para rebasar los temporizadores tras la reanudación de la aplicación desde el estado de suspensión al estado de ejecución.

40 En otro ejemplo de gestión de un estado de aplicación como se proporciona en la presente memoria, se puede mantener una cola de mensajes de aplicación asociada con una aplicación mientras que la aplicación está en un estado de suspensión. La cola de mensajes de aplicación se puede configurar para almacenar mensajes de sistema para la aplicación (por ejemplo, mensajes de sistema asociados con los ajustes dentro de un panel de control). Por ejemplo, un sistema operativo de un dispositivo de tableta puede poner en cola los mensajes del sistema para un editor de textos, tal como un mensaje de sistema color de interfaz de usuario = verde, un mensaje de sistema idioma = inglés, el mensaje de sistema orientación de vista de tableta = retrato, etc. La cola de mensajes de aplicación puede recibir y/o almacenar mensajes de sistema mientras que la aplicación está suspendida. Debido a que los mensajes superpuestos se pueden recibir a lo largo del tiempo, lo que puede hacer que los mensajes puestos en cola actualmente sean irrelevantes (por ejemplo, mientras que están suspendidos, se puede recibir un primer mensaje de que la orientación de la vista de la tableta se ha establecido a retrato, entonces más tarde se puede recibir un segundo mensaje de que la orientación de la vista de la tableta se ha establecido a paisaje, lo que puede hacer que el primer mensaje sea irrelevante), puede ser ventajoso para gestionar la cola de mensajes de aplicación de modo que la aplicación se pueda poner en un estado de ejecución en un estado de aplicación actualizado actual. Por consiguiente, se puede recibir un primer mensaje de sistema dirigido la cola de aplicación mientras que la aplicación está en el estado de suspensión. Se puede hacer una determinación en cuanto a si el primer mensaje de sistema corresponde a un segundo mensaje de sistema dentro de la cola de mensajes de aplicación. Tras determinar que el primer mensaje de sistema corresponde al segundo mensaje del sistema, el primer mensaje de sistema y/o el segundo mensaje de sistema se puede modificar y/o eliminar (por ejemplo, un primer mensaje de sistema recién recibido de que el color de interfaz de usuario = verde se puede poner en la cola de mensajes de aplicación, mientras que se puede eliminar un segundo mensaje de sistema que ya está en la cola de que el color de interfaz de usuario = roja). De esta forma, tras la reanudación de la aplicación al estado de ejecución, la aplicación puede procesar mensajes relevantes del sistema desde la cola de mensajes de aplicación en lugar de mensajes de sistema “viejos” y/o irrelevantes. La gestión de la cola de mensajes de aplicación puede tener un impacto positivo

adicional en el rendimiento, ya que los mensajes redundantes no consumen innecesariamente recursos del sistema cuando el proceso de destino no es ejecutable (por ejemplo, un nuevo mensaje se puede dejar caer/eliminados, mensajes correspondientes más antiguos se pueden dejar caer/eliminar mientras que el nuevo mensaje se puede retener, los mensajes se pueden unir inteligentemente, etc.).

5 Para el logro de los fines anteriores y relacionados, la siguiente descripción y los dibujos anexos exponen ciertos aspectos ilustrativos e implementaciones. Éstos son indicativos de unas pocas de las diversas formas en que se pueden emplear uno o más aspectos. Otros aspectos, ventajas y características novedosas de la descripción llegarán a ser evidentes a partir de la siguiente descripción detallada cuando se considere junto con los dibujos anexos.

10 Descripción de los dibujos

La Fig. 1 es un diagrama de flujo que ilustra un método ejemplar de suspensión de uno o más procesos asociados con una aplicación.

La Fig. 2 es un diagrama de flujo que ilustra un método ejemplar de suspensión de uno o más procesos asociados con una aplicación.

15 La Fig. 3 es un diagrama de bloques de componentes que ilustra un sistema ejemplar para suspender uno o más procesos asociados con una aplicación.

La Fig. 4 es un diagrama de flujo que ilustra un método ejemplar de determinación de si reanudar un proceso suspendido en base a un conjunto de políticas de activación.

20 La Fig. 5 es un diagrama de flujo que ilustra un método ejemplar de determinación de si reanudar un proceso suspendido en base a un conjunto de políticas de activación.

La Fig. 6 es un diagrama de bloques de componentes que ilustra un sistema ejemplar para determinar si reanudar un proceso suspendido en base a una política de activación.

La Fig. 7 es un diagrama de flujo que ilustra un método ejemplar de gestión de un estado de aplicación de una aplicación.

25 La Fig. 8 es un diagrama de flujo que ilustra un método ejemplar de gestión de un estado de aplicación de una aplicación.

La Fig. 9 es un diagrama de bloques de componentes que ilustra un sistema ejemplar para gestionar un estado de aplicación de una aplicación.

30 La Fig. 10 es un diagrama de bloques de componentes que ilustra un sistema ejemplar para gestionar un estado de aplicación de una aplicación.

La Fig. 11 es una ilustración de un medio legible por ordenador ejemplar en donde se pueden comprender instrucciones ejecutables por procesador configuradas para incorporar una o más de las disposiciones expuestas en la presente memoria.

35 La Fig. 12 ilustra un entorno informático ejemplar en donde se pueden implementar una o más de las disposiciones expuestas en la presente memoria.

Descripción detallada

40 La materia objeto reivindicada se describe ahora con referencia a los dibujos, en donde números de referencia iguales se usan de manera general para referirse a elementos similares en todas partes. En la siguiente descripción, con propósitos de explicación, se exponen numerosos detalles específicos con el fin de proporcionar una comprensión minuciosa de la materia objeto reivindicada. No obstante, puede ser evidente que la materia objeto reivindicada se puede practicar sin estos detalles específicos. En otros casos, se ilustran estructuras y dispositivos en forma de diagrama de bloques con el fin de facilitar la descripción de la materia objeto reivindicada.

45 Se proporcionan en la presente memoria uno o más sistemas y/o técnicas para suspender uno o más procesos asociados con una aplicación, determinar si reanudar un proceso suspendido en base a una o más políticas de activación, y gestionar un estado de aplicación de una aplicación. En particular, uno o más procesos asociados con una aplicación se pueden suspender (por ejemplo, procesos asociados con una aplicación se pueden identificar en base a una jerarquía de contenedores lógicos, y se pueden suspender como un grupo). Un proceso suspendido se puede reanudar en base a mantener un conjunto de políticas de activación que indican las razones para reanudar el proceso suspendido a un estado de ejecución (por ejemplo, si se recibe un número umbral de llamadas de comunicación entre procesos desde una aplicación solicitante a un proceso suspendido, entonces el proceso suspendido se puede poner en un estado de ejecución para procesar las llamadas de comunicación entre procesos). Durante la suspensión de una aplicación, un estado de aplicación se puede derivar de los temporizadores asociados

con una aplicación suspendida, de manera que los temporizadores se pueden rebasar usando el estado de aplicación tras la reanudación de la aplicación suspendida a un estado de ejecución. Además, se puede gestionar una cola de mensajes de aplicación mientras que se suspende una aplicación de modo que se modifiquen y/o eliminen mensajes irrelevantes y/u viejos de la cola de aplicación. De esta forma, tras entrar en el estado de ejecución, la aplicación puede procesar mensajes de sistema relevantes de modo que la aplicación se pueda poner en un estado de aplicación actual.

Una realización de suspensión de uno o más procesos asociados con una aplicación se ilustra mediante un método 100 ejemplar en la Fig. 1. En 102, comienza el método. En 104, se puede recibir una notificación de que una aplicación está inactiva (por ejemplo, la aplicación se puede pasar a un estado de segundo plano, la aplicación puede ser invisible para un usuario (por ejemplo, detrás de una aplicación diferente que ha sido traída al primer plano en un monitor), el usuario puede no haber interactuado con la aplicación durante un período de tiempo, etc.). En 106, se pueden identificar uno o más procesos asociados con la aplicación. En un ejemplo, se puede consultar una jerarquía de contenedores lógicos (por ejemplo, la jerarquía de contenedores lógicos 308 de la Fig. 3) para determinar qué procesos están asociados con la aplicación. Por ejemplo, la aplicación se puede asignar a un contenedor lógico raíz dentro de la jerarquía de contenedores lógicos, mientras que los procesos asociados con la aplicación se pueden asignar a los contenedores lógicos asociados con (por ejemplo, anidados dentro de) el contenedor lógico raíz.

En un ejemplo de identificación de uno o más procesos, un primer proceso se puede determinar como que está asociado con la aplicación en base a una determinación de que el primer proceso se asigna a un primer contenedor lógico asociado con (por ejemplo, anidado dentro de) el contenedor lógico raíz (por ejemplo, el primer proceso puede ser un proceso secundario directo de la aplicación). En otro ejemplo, un segundo proceso se puede determinar como que está asociado con la aplicación en base a una determinación de que el segundo proceso se asigna al primer contenedor lógico asociado con el contenedor lógico raíz (por ejemplo, el segundo proceso puede ser un proceso secundario directo de la aplicación). En otro ejemplo, un segundo proceso se puede determinar como que está asociado con la aplicación en base a una determinación de que el segundo proceso se asigna a un segundo contenedor lógico asociado con el primer contenedor lógico (por ejemplo, el segundo proceso puede ser un proceso secundario directo del primer proceso y/o un proceso secundario indirecto de la aplicación). De esta forma, se pueden determinar el uno o más procesos asociados con la aplicación.

En 108, el uno o más procesos se pueden poner en un estado de suspensión. Por ejemplo, un proceso suspendido se puede marcar como no elegible para la ejecución del procesador, de manera que los subprocesos que comprenden el proceso suspendido no se pueden considerar como candidatos por un programador de subprocesos para su ejecución. En un ejemplo, la aplicación y/u otras aplicaciones asociadas con el uno o más procesos pueden ser informados de la suspensión, lo que puede proporcionar una oportunidad para que tales aplicaciones guarden información de estado (por ejemplo, una aplicación de red social puede guardar información de estado asociada con un perfil actual que está siendo visto por un usuario, de manera que tras la reanudación, la aplicación de red social pueda presentar el perfil actual al usuario). El contenido de la memoria asociado con un proceso suspendido se puede guardar en un dispositivo de almacenamiento como contenido de memoria guardado. De esta forma, el contenido de memoria guardado se puede devolver a la memoria, lo que puede permitir que el proceso suspendido reanude rápidamente la ejecución.

Se puede apreciar que un proceso creado recientemente (por ejemplo, un proceso creado después de que se suspendan el uno o más procesos) se puede poner en un estado de suspensión en base a una determinación de que el proceso recién creado está asociado con uno o más de los procesos suspendidos y/o la aplicación inactiva. Por ejemplo, un proceso recién creado se puede determinar como asociado con la aplicación inactiva en base a determinar que el proceso recién creado se asigna a un contenedor lógico asociado con el contenedor lógico raíz de la aplicación inactiva. De esta forma, el proceso recién creado se puede poner en el estado de suspensión. En 110, el método termina.

Una realización de suspensión de uno o más procesos asociados con una aplicación se ilustra mediante un método 200 ejemplar en la Fig. 2. En 202, comienza el método. En 204, se puede recibir una notificación de que una aplicación está inactiva. En 206, se puede atravesar iterativamente una jerarquía de contenedores lógicos para identificar uno o más procesos asociados con la aplicación. En un ejemplo, un contenedor lógico raíz asignado a una aplicación se puede identificar dentro de la jerarquía de contenedores lógicos. Un primer proceso se puede determinar como que está asociado con la aplicación en base a determinar que el primer proceso se asigna a un primer contenedor lógico asociado con (por ejemplo, anidado dentro de) el contenedor lógico raíz. En otro ejemplo, un segundo proceso se puede determinar como que está asociado con la aplicación en base a una determinación de que el segundo proceso se asigna al primer contenedor lógico asociado con el contenedor lógico raíz (por ejemplo, el segundo proceso puede ser un proceso secundario directo de la aplicación). En otro ejemplo, un segundo proceso se puede determinar como que está asociado con la aplicación en base a una determinación de que el segundo proceso se asigna a un segundo contenedor lógico asociado con el primer contenedor lógico (por ejemplo, el segundo proceso puede ser un proceso secundario directo del primer proceso y/o un proceso secundario indirecto de la aplicación). De esta forma, se pueden identificar el uno o más procesos asociados con la aplicación. En 208, el uno o más procesos se pueden poner en un estado de suspensión. De esta forma, se pueden suspender los procesos y/o procesos recién creados asociados con la aplicación inactiva. En 210, el método termina.

La Fig. 3 ilustra un ejemplo de un sistema 300 configurado para suspender uno o más procesos asociados a una aplicación. El sistema 300 puede comprender un componente de suspensión 322. El componente de suspensión 322 se puede configurar para recibir una notificación de que una aplicación está inactiva. Por ejemplo, una aplicación de editor de textos 302 y una aplicación de hoja de cálculo 304 se pueden alojar dentro de un entorno informático. Un usuario puede conmutar de la aplicación de editor de textos 302 a la aplicación de hoja de cálculo 304, lo que puede dar como resultado la aplicación de editor de textos 302 que pase a un estado de segundo plano. De esta forma, el componente de suspensión 322 puede recibir una notificación inactiva de editor de textos 306.

El componente de suspensión 322 se puede configurar para atravesar iterativamente una jerarquía de contenedores lógicos 308 para identificar uno o más procesos asociados con la aplicación. En un ejemplo, la aplicación de editor de textos 302 se puede asociar con un proceso de editor de textos 312 asignado a un contenedor lógico raíz de editor de textos 310. Mientras que se ejecuta, la aplicación de editor de textos 302 puede haber creado uno o más procesos secundarios, tales como un proceso de impresión de editor de textos 314 y/o un proceso de guardado de editor de textos 316, que puede haber sido asignado al contenedor lógico raíz de editor de textos 310. Además, la aplicación de editor de textos 302 puede haber invocado una funcionalidad de revisión ortográfica asociada con un proceso de revisión ortográfica 320 asignado a un contenedor lógico de revisión ortográfica 318. El contenedor lógico de revisión ortográfica 318 puede estar anidado dentro del contenedor lógico raíz de editor de textos 310 debido a que el proceso de revisión ortográfica 320 puede estar asociado con la aplicación de editor de textos 302. Por consiguiente, el componente de suspensión 322 puede determinar que el proceso de impresión de editor de textos 314, el proceso de guardado de editor de textos 316, y/o el proceso de revisión ortográfica 320 se pueden asociar con la aplicación de editor de textos 302 en base a atravesar la jerarquía de contenedores lógicos 308. De esta forma, el componente de suspensión 322 puede suspender 324 el proceso de editor de textos 312, el proceso de impresión de editor de textos 314, el proceso de guardado de editor de textos 316, el proceso de revisión ortográfica 320 y/o los procesos recién creados asociados con la aplicación de editor de textos 302 en base a la aplicación de editor de textos 302 que está inactiva, lo que puede dar como resultado un consumo de energía mejorado debido a que tales procesos y/o aplicaciones pueden no ser elegibles para consumir recursos de ejecución del procesador y/u otros recursos que consuman energía.

Una realización de determinación de si reanudar un proceso suspendido en base a un conjunto de políticas de activación se ilustra mediante un método 400 ejemplar en la Fig. 4. En 402, comienza el método. En 404, se puede mantener un conjunto de políticas de activación asociadas con un proceso suspendido. Una política de activación puede ser indicativa de una razón para reanudar el proceso suspendido a un estado de ejecución. En un ejemplo, una política de activación puede corresponder a una llamada de comunicación entre procesos desde una aplicación solicitante para el proceso suspendido (por ejemplo, una aplicación de hoja de cálculo puede intentar solicitar texto "cortado" de una aplicación de editor de textos para satisfacer una operación de "pegar"; una aplicación de presentación puede comprender un objeto de texto "pegado" cuya funcionalidad y/o datos se pueden proporcionar por la aplicación de editor de textos; etc.). En otro ejemplo, una política de activación puede corresponder a una entrada de usuario asociada con la aplicación suspendida (por ejemplo, un usuario puede intentar interactuar con una aplicación de editor de textos suspendida). En otro ejemplo, un componente (por ejemplo, un componente de hardware, un componente de software, un componente de un tercero remoto, un componente web, etc.) puede solicitar acceso al proceso suspendido. En otro ejemplo, una política de activación puede corresponder a una transición de una aplicación suspendida asociada con el proceso suspendido desde un estado de segundo plano a un estado de primer plano. De esta forma, el conjunto de políticas de activación puede comprender una variedad de políticas de activación asociadas con diversos eventos/razones para reanudar el proceso suspendido.

Un contador puede estar asociado con una política de activación. El contador puede ser indicativo de una serie de notificaciones de activación asociadas con la política de activación (por ejemplo, casos de un evento/razón especificado por una política de activación) que están actualmente pendientes para el proceso suspendido. Una notificación de activación puede ser una notificación de un evento/razón especificado por una política de activación, tal como una notificación de una llamada de comunicación entre procesos. El contador se puede aumentar (por ejemplo, puede estar pendiente un evento asociado a una notificación de activación) y/o disminuir (por ejemplo, se pueden completar un evento asociado con una notificación de activación). Se puede apreciar que diversas políticas de activación y contadores correspondientes se pueden mantener dentro del conjunto de políticas de activación. Por ejemplo, un primer contador para una política de activación de entrada de usuario se puede aumentar a 1 tras recibir una primera notificación de activación de que un usuario intentó interactuar con un editor de textos suspendido. El primer contador para la política de activación de entrada de usuario se puede aumentar a 2 tras recibir una segunda notificación de activación de que el usuario hizo un segundo intento de interactuar con el editor de textos suspendido. Un segundo contador para una política de activación de comunicación entre procesos se puede aumentar a 1 tras recibir una tercera notificación de activación de que una aplicación de hoja de cálculo intentó comunicarse con el editor de textos suspendido. El segundo contador se puede disminuir a 0 tras una notificación de que la aplicación de hoja de cálculo ya no desea para comunicarse con el editor de textos suspendido (por ejemplo, un usuario cierra la aplicación de hoja de cálculo). De esta forma, se pueden mantener los contadores asociados con las políticas de activación dentro del conjunto de políticas de activación.

En un ejemplo, las notificaciones de activación se pueden filtrar (por ejemplo, ignorar, eliminar y/o abstener de ser emitida) en base a cómo se hace la transición de un contador a la luz de un estado de ejecución/suspensión del proceso. En un ejemplo, tras determinar que el proceso está en el estado de ejecución, se puede filtrar una

notificación de activación que hace la transición del contador de un valor 0 a un valor 1 (por ejemplo, una notificación de activación que puede desencadenar una política de activación para poner el proceso que ya se está ejecutando en el estado de ejecución se puede filtrar debido a que el proceso de ejecución ya se está ejecutando). En otro ejemplo, tras determinar que el proceso está en el estado suspendido, se puede filtrar una notificación de activación que hace la transición del contador de un valor 1 a un valor 0 (por ejemplo, una notificación de activación que puede desencadenar una política de activación para poner el proceso ya suspendido en el estado suspendido se puede filtrar debido a que el proceso suspendido ya está suspendido).

En 406, el conjunto de políticas de activación asociadas con el proceso suspendido se puede evaluar para determinar si poner el proceso suspendido en un estado de ejecución o retener el proceso suspendido en un estado de suspensión. En un ejemplo, el proceso suspendido se puede poner en el estado de ejecución en base a determinar al menos un contador asociado con una política de activación que comprende un valor por encima de un valor umbral (por ejemplo, al menos un contador tiene un valor mayor que 0). Si ningún contador comprende un valor por encima del valor umbral, entonces el proceso suspendido se puede mantener en el estado de suspensión. En otro ejemplo, los valores de uno o más contadores se pueden agregar juntos para crear un valor agregado. Por ejemplo, se puede determinar un valor agregado de 4 en base a un primer contador asociado con una política de activación de entrada de usuario que tiene un valor de 2 y un segundo contador asociado con una política de activación de llamada de comunicación entre procesos de terceros remotos que tiene un valor de 2. El valor agregado de 4 se puede ajustar en base a las ponderaciones asociadas con las políticas de activación. Por ejemplo, la política de activación de entrada de usuario puede tener un valor ponderado de 4 (por ejemplo, en base a la aplicación de un modificador de ponderación x2 al valor de 2) debido a que puede ser deseable reanudar/activar el proceso suspendido en respuesta a la entrada de usuario de modo que una aplicación asociada con el proceso suspendido es altamente sensible al usuario. La política de comunicación entre procesos de terceros remotos puede tener un valor ponderado de 1 (por ejemplo, en base a aplicar un modificador de ponderación x5 al valor de 2) debido a que puede ser deseable reducir la capacidad de respuesta para terceros remotos que pueden estar intentando comunicar “ruidosamente” con el proceso suspendido por razones no críticas y/o no deseables. El proceso suspendido se puede poner en el estado de ejecución en base a determinar si el valor agregado está por encima de un valor umbral (por ejemplo, un valor agregado de 5 puede estar por encima de un valor umbral de 3).

Se puede apreciar que en un ejemplo, el conjunto de políticas de activación se puede asociar con un contenedor lógico asignado al proceso suspendido dentro de una jerarquía de contenedores lógicos. Por consiguiente, si el proceso suspendido se pone en el estado de ejecución, entonces uno o más procesos y/o aplicaciones asociados con el proceso suspendido se pueden poner en el estado de ejecución. Por ejemplo, un proceso de editor de textos se puede poner en el estado de ejecución. El proceso de editor de textos se puede asignar a un contenedor lógico raíz de editor de textos. Un proceso de revisión ortográfica también se puede poner en el estado de ejecución en base a determinar el proceso de revisión ortográfica con que se asigna a un contenedor lógico asociado (por ejemplo, anidado dentro de) el contenedor lógico raíz de editor de textos. De esta forma, uno o más procesos relacionados se pueden reanudar al estado de ejecución. En un ejemplo, el contenido de la memoria guardada asociado con los procesos reanudados se puede restaurar a la memoria, de modo que tales procesos puedan reanudar inmediatamente la ejecución de una manera sensible, mientras que se ahorra energía.

Una vez puesto en el estado de ejecución como un proceso de ejecución, se puede hacer una determinación en cuanto a si y/o cuándo devolver el proceso de ejecución al estado de suspensión. En un ejemplo, se puede evaluar el estado actual del conjunto de políticas de activación (por ejemplo, si los valores de contador y/o un valor agregado de los valores de contador no están por encima del valor umbral, entonces el proceso de ejecución se puede volver a colocar en el estado de suspensión). En otro ejemplo, se puede evaluar un tiempo transcurrido desde la terminación de una notificación de activación que hizo que el proceso de ejecución se pusiese en el estado de ejecución. Por ejemplo, una aplicación de presentación puede comprender un objeto de texto “pegado” cuya funcionalidad y/o datos se pueden proporcionar por una aplicación de editor de textos suspendida. Se puede hacer una llamada de comunicación entre procesos desde la aplicación de presentación a la aplicación de editor de textos suspendida con respecto al objeto de texto “pegado”. Se puede recibir una notificación de activación de la llamada de comunicación, y la aplicación de editor de textos suspendida se puede poner en un estado de ejecución para satisfacer la llamada de comunicación entre procesos. Puede ser ventajoso permitir que la aplicación de editor de textos permanezca en el estado de ejecución durante un periodo de tiempo (por ejemplo, a diferencia de volver a poner inmediatamente la aplicación de editor de textos en el estado suspendido) debido a que la aplicación de presentación puede hacer llamadas de comunicación entre procesos posteriores a la aplicación de editor de textos con respecto al objeto de texto “pegado”. En 408, termina el método.

Una realización de determinación de si reanudar un proceso suspendido en base a un conjunto de políticas de activación se ilustra mediante un método 500 ejemplar en la Fig. 5. En 502, comienza el método. En 504, se puede mantener una política de activación asociada con un contenedor lógico asignado a un proceso suspendido dentro de una jerarquía de contenedores lógicos. La política de activación puede ser indicativa de una razón para reanudar el proceso suspendido a un estado de ejecución. La política de activación puede comprender un contador indicativo de una serie de notificaciones de activación asociadas con la política de activación que están actualmente pendientes para el proceso suspendido. Una notificación de activación puede ser una notificación de un evento/razón para reanudar el proceso suspendido especificada por una política de activación, tal como una notificación de una llamada de comunicación entre procesos. El contador se puede aumentar en base a recibir una notificación de

activación asociada con la política de activación (por ejemplo, un contador asociado con una política de activación de entrada de usuario se puede aumentar en base a recibir una notificación de activación que indica que un usuario intentó interactuar con una aplicación suspendida asociada con el proceso suspendido). El contador se puede disminuir en base a recibir una notificación de que la notificación de activación no está pendiente actualmente (por ejemplo, el contador se puede disminuir en base a una notificación de que el usuario ha cerrado la sesión en un dispositivo informático que aloja la aplicación suspendida).

En un ejemplo, tras determinar que un valor del contador está por encima de un umbral, el proceso suspendido se puede poner en un estado de ejecución, en 506. En un ejemplo, la jerarquía del contenedores lógicos se puede atravesar para identificar uno o más procesos suspendidos adicionales asociados con el proceso suspendido en base a determinar que el uno o más procesos suspendidos adicionales se asignan al contenedor lógico asociado con (por ejemplo, anidados dentro de) el contenedor lógico del proceso suspendido. De esta forma, el uno o más procesos suspendidos adicionales se pueden poner en el estado de ejecución. En otro ejemplo, tras determinar que el valor del contador está por debajo del umbral, el proceso suspendido se puede mantener en el estado suspendido, en 508. De esta forma, el proceso suspendido se puede poner en el estado de ejecución o retener en el estado suspendido en base a la política de activación. En 510, termina el método.

La Fig. 6 ilustra un ejemplo de un sistema 600 configurado para determinar si reanudar un proceso suspendido en base a una política de activación. El sistema 600 puede comprender un componente de activación 624. El componente de activación 624 se puede configurar para mantener una política de activación (por ejemplo, un conjunto de políticas de activación 626, tales como la política de activación de comunicación entre procesos 628, la política de activación de entrada de usuario 630, la política de activación de segundo plano a primer plano 632 y/u otras políticas de activación). La política de activación se puede asociar con un contenedor lógico asignado a un proceso suspendido dentro de una jerarquía de contenedores lógicos 610. Por ejemplo, el conjunto de políticas de activación 626 se puede asociar con un contenedor lógico de editor de textos 612 asociado con un proceso de editor de textos 614 suspendido de una aplicación de editor de textos 602 suspendida. En un ejemplo, un proceso de impresión de editor de textos 616 suspendido, un proceso de guardado de editor de textos 618 suspendido y/o un proceso de revisión ortográfica 622 suspendido pueden haber sido suspendidos en base a la suspensión del proceso de editor de textos 614 debido a que tales procesos pueden estar relacionados (por ejemplo, el proceso de revisión ortográfica 622 suspendido se puede asignar a un contenedor lógico de revisión ortográfica 620 anidado dentro del contenedor lógico de editor de textos 612).

La política de activación, tal como la política de activación de comunicación entre procesos 628, se puede asociar con un contador indicativo de una serie de notificaciones de activación asociadas con la política de activación que están actualmente pendientes para el proceso suspendido. El componente de activación 624 se puede configurar para aumentar el contador en base a recibir una notificación de activación asociada con la política de activación y/o disminuir el contador en base a recibir una notificación de que la notificación de activación no está pendiente actualmente. Por ejemplo, se puede recibir una notificación de activación 608. La notificación de activación 608 puede indicar que una aplicación de hoja de cálculo 604 ha enviado una llamada de comunicación entre procesos a la aplicación de editor de textos 602 suspendida con respecto a un objeto de texto pegado 606 (por ejemplo, el objeto de texto pegado 606 puede derivar en funcionalidad y/o datos de la aplicación de editor de textos 602 suspendida y, de este modo, la aplicación de hoja de cálculo 604 solicita tal funcionalidad y/o datos de la aplicación de editor de textos 602 suspendida). El componente de activación 624 puede aumentar el contador asociado con la política de activación de comunicación entre procesos 628 a 1 en base a la notificación de activación 608.

El componente de activación 624 se puede configurar para poner el proceso suspendido en un estado de ejecución en base a determinar que un valor del contador está por encima de un umbral, de otro modo, el componente de activación 624 puede retener el proceso suspendido en un estado de suspensión. Por ejemplo, el componente de activación 624 puede poner el proceso de editor de textos 614 suspendido en un estado de ejecución en base al contador de la política de activación de comunicación entre procesos 628 que comprende un valor de 1 que es mayor que 0. El componente de activación 624 puede atravesar la jerarquía de contenedores lógicos 610 para identificar uno o más procesos suspendidos adicionales asociados con el proceso de editor de textos 614 suspendido que se ha de reanudar al estado de ejecución. Por ejemplo, el proceso de impresión de editor de textos 616 suspendido, el proceso de guardado de editor de textos 618 suspendido y/o el proceso de revisión ortográfica 622 suspendido se pueden identificar en base a determinar que tales procesos están asignados al contenedor lógico de editor de textos 612 y/o contenedores lógicos asociados con (por ejemplo, anidados dentro de) el contenedor lógico de editor de textos 612, tal como el contenedor lógico de revisión ortográfica 620. De esta forma, el componente de activación 624 puede poner 634 el proceso de editor de textos 614 suspendido, el proceso de impresión de editor de textos 616 suspendido, el proceso de editor de textos 616 suspendido, el proceso de guardado de editor de textos 618 suspendido, y/o el proceso de revisión ortográfica 622 suspendido en el estado de ejecución, de manera que la aplicación de editor de textos 602 puede responder a la llamada de comunicación entre procesos desde la aplicación de hoja de cálculo 604. En un ejemplo, el componente de activación 624 puede poner el contenido de memoria guardado asociado con los procesos reanudados en la memoria para mejorar la capacidad de respuesta de la aplicación de editor de textos 602.

Una realización de gestión de un estado de aplicación de una aplicación se ilustra mediante un método 700 ejemplar en la Fig. 7. En 702, comienza el método. En 704, se puede mantener una lista de temporizadores asociados con

una aplicación (por ejemplo, se pueden mantener uno o más temporizadores de procesos asociados con la aplicación). Es decir, los procesos asociados con la aplicación se pueden asociar con uno o más temporizadores, tales como temporizadores relativos (por ejemplo, un tiempo de espera establecido en 20 segundos por una aplicación de editor de textos para un proceso de guardado de editor de textos), temporizadores absolutos (por ejemplo, un tiempo de espera establecido a las 5:00 AM del 15/9/11 por una aplicación de calendario), y/u otros tipos de temporizadores. En un ejemplo de un temporizador, una aplicación de editor de textos puede establecer un temporizador relativo de 20 segundos para un proceso de guardado de editor de textos debido a que el proceso de guardado de editor de textos generalmente no tarda más de 20 segundos en completarse a menos que haya ocurrido un error. En otro ejemplo de un temporizador, una aplicación de calendario puede establecer un temporizador absoluto de 5:00 AM del 15/9/11 como recordatorio para proporcionar a un usuario una notificación de activación a las 6:00 AM del 15/9/11.

En un ejemplo, uno o más de los temporizadores se pueden asociar con un contenedor lógico raíz asignado a la aplicación y/o uno o más contenedores lógicos asignados a procesos asociados con la aplicación dentro de una jerarquía de contenedores lógicos. Por ejemplo, un primer temporizador asociado con un primer proceso de la aplicación se puede mantener en base a determinar que el primer proceso está asignado a un primer contenedor lógico asociado con (por ejemplo, anidado dentro de) el contenedor lógico raíz dentro de la jerarquía de contenedores lógicos. Se puede mantener un segundo temporizador asociado con un segundo proceso de la aplicación en base a determinar que el segundo proceso está asignado a un segundo contenedor lógico asociado con el contenedor lógico raíz y/o el primer contenedor lógico. De esta forma, se puede mantener la lista de temporizadores asociados con los procesos de la aplicación.

Tras recibir una notificación de que la aplicación se ha de poner en un estado de suspensión, se puede crear un estado de aplicación que comprende datos de rebasado de temporizador derivados de la lista de temporizadores, en 706. En un ejemplo, los datos de rebasado de temporizador absoluto asociados con un temporizador absoluto se pueden almacenar dentro del estado de aplicación. Los datos de rebasado de temporizador absoluto pueden comprender un tiempo de expiración original del temporizador absoluto (por ejemplo, la aplicación del calendario puede desear una notificación a las 5:00 AM del 15/9/11, independientemente de si el tiempo ha pasado o no debido a un proceso y/o aplicación que se pone en un estado de suspensión). En otro ejemplo, los datos de rebasado de temporizador relativo asociados con un temporizador relativo se pueden almacenar dentro del estado de aplicación (por ejemplo, la aplicación de editor de textos puede desear una notificación de tiempo de espera si el proceso de guardado de editor de textos no se completa dentro de 20 segundos mientras que está en el estado de ejecución). Los datos de rebasado de temporizador relativo pueden comprender un tiempo hasta el valor de expiración (por ejemplo, un lapso de tiempo desde cuando el temporizador relativo se suspende hasta un tiempo de expiración original del temporizador relativo). Los datos de rebasado de temporizador relativo pueden, por ejemplo, comprender un tiempo hasta el valor de expiración diferente al tiempo de expiración original debido a que la aplicación de editor de textos puede estar ocupada con si el proceso de guardado de editor de textos tardó más de 20 segundos en completarse, mientras que en el estado de ejecución (por ejemplo, tardar más de 20 segundos puede indicar que ocurrió un error fatal), de manera que una suspensión del proceso de guardado de editor de textos puede afectar al temporizador relativo debido a que la expiración del temporizador relativo debida a que el proceso de guardado de editor de textos que está suspendido puede no ser representativo de un tiempo de espera fatal con el que puede estar ocupada la aplicación de editor de textos. De esta forma, se puede crear el estado de aplicación.

Tras recibir la notificación de que la aplicación ha de ser puesta en un estado de ejecución desde el estado de suspensión, los datos de rebasado de temporizador dentro del estado de aplicación se puede aplicar a uno o más temporizadores asociados con la lista de temporizadores. En un ejemplo, los datos de rebasado de temporizador absoluto se pueden aplicar a un temporizador absoluto para crear un temporizador absoluto rebasado. El temporizador absoluto rebasado puede comprender un tiempo de expiración original del temporizador absoluto. Si el tiempo de expiración original ha expirado, entonces se puede proporcionar una notificación de la expiración a la aplicación. En otro ejemplo, los datos de rebasado de temporizador relativo se pueden aplicar a un temporizador relativo para crear un temporizador relativo rebasado. El temporizador relativo rebasado puede comprender un temporizador de expiración rebasado derivado de aplicar el tiempo hasta el valor de expiración a una hora actual. De esta forma, la aplicación y/o los procesos asociados con la aplicación se pueden poner en un estado de aplicación actual con respecto a los temporizadores. En 708, termina el método.

Una realización de gestión de un estado de aplicación de una aplicación se ilustra mediante un método 800 ejemplar en la Fig. 8. En 802, comienza el método. Una aplicación se puede asociar con una cola de mensajes de aplicación. La cola de mensajes de aplicación se puede configurar para almacenar mensajes de sistema para la aplicación. De esta forma, la aplicación puede recibir mensajes de sistema a través de la cola de mensajes de aplicación (por ejemplo, la aplicación puede sacar de la cola los mensajes de sistema mientras que está en un estado de ejecución). Un mensaje de sistema puede corresponder a una variedad de información relacionada con un estado de aplicación, tal como un cambio en los ajustes del sistema. En un ejemplo, un usuario puede girar un dispositivo de tableta, que puede cambiar la orientación de la vista desde un modo de vista de paisaje al modo de vista de retrato. Un sistema operativo del dispositivo de tableta puede poner en cola un mensaje de sistema en la cola de mensajes de aplicación indicando que el modo de vista actual se ha establecido en retrato. De esta forma, la aplicación puede sacar de la cola el mensaje de sistema, y puede reproducirse a sí misma en modo retrato para que coincida con el estado actual del dispositivo de tableta. Se puede apreciar que un mensaje de sistema puede

comprender una variedad de otra información, tal como el color de la interfaz de usuario, los ajustes de idioma, la información del dispositivo de entrada, etc.

En 804, la cola de mensajes de aplicación asociada con una aplicación se puede mantener mientras que la aplicación está en un estado de suspensión. Es decir, la cola de mensajes de aplicación puede recibir generalmente mensajes de sistema independientemente de si la aplicación está en estado suspendido y/o en un estado de ejecución (por ejemplo, mientras que una aplicación de editor de textos está suspendida, un usuario puede girar un dispositivo de tableta múltiples veces y, de este modo, el sistema operativo puede poner en cola múltiples mensajes de sistema correspondientes a los modos de vista actuales). Desafortunadamente, múltiples mensajes de sistema correspondientes se pueden poner en cola dentro de la cola de mensajes de aplicación mientras que la aplicación está en el estado suspendido, donde uno o más mensajes pueden llegar a ser irrelevantes y/o viejos. Por ejemplo, mientras que una aplicación de editor de textos está suspendida, se pueden poner en cola 4 mensajes de sistema en la cola de mensajes de aplicación con el tiempo. Un usuario puede cambiar un ajuste de idioma de sistema de un dispositivo de tableta a alemán y, de este modo, un mensaje de sistema idioma = alemán puede poner en cola en la cola de sistema de aplicación. A continuación, el usuario puede girar el dispositivo de tableta a modo paisaje, y de este modo un modo de vista = retrato se puede poner en cola en la cola de sistema de aplicación. El usuario puede cambiar entonces el ajuste de idioma de sistema del dispositivo de tableta a inglés y, de este modo, un mensaje de sistema idioma = inglés se puede poner en cola en la cola de sistema de aplicación. Finalmente, el usuario puede girar el dispositivo de tableta a modo retrato, y de este modo, un modo de vista = retrato se puede poner en cola en la cola de sistema de aplicación. De esta forma, la cola de mensajes de aplicación puede comprender mensajes de sistema irrelevantes y/o viejos, tales como el mensaje de sistema idioma = alemán y el modo de vista = paisaje debido a que el idioma y el modo de vista se han cambiado a otros valores. Por consiguiente, la cola de mensajes de aplicación se puede mantener para eliminar y/o modificar los mensajes irrelevantes y/o viejos.

En un ejemplo, un primer mensaje de sistema directo a la cola de mensajes de aplicación se puede recibir mientras que la aplicación está en el estado de suspensión. Se puede hacer una determinación en cuanto a si el primer mensaje de sistema corresponde a un segundo mensaje de sistema dentro de la cola de mensajes de aplicación (por ejemplo, si el primer mensaje de sistema corresponde a un mensaje de sistema de idioma, entonces la cola de mensajes de aplicación se puede atravesar para identificar uno o más mensajes de sistema correspondientes al mensaje de sistema idioma). Tras determinar que el primer mensaje de sistema correspondiente al segundo mensaje de sistema, el primer mensaje de sistema y/o el segundo mensaje de sistema se pueden eliminar y/o modificar (por ejemplo, el primer mensaje de sistema, que puede ser más nuevo y/o más relevante, se puede retener, mientras que el segundo mensaje de sistema se puede eliminar). Por ejemplo, las clasificaciones de relevancia se pueden asignar al primer y al segundo mensaje de sistema para determinar qué mensaje de sistema modificar y/o eliminar. En 806, termina el método.

La Fig. 9 ilustra un ejemplo de un sistema 900 configurado para gestionar un estado de aplicación de una aplicación, tal como datos de temporizador y/o datos de mensajes de sistema. El sistema 900 puede comprender un gestor de estado de aplicación 910. El gestor de estado de aplicación 910 se puede configurar para mantener una lista de temporizadores 904 asociados con una aplicación. Por ejemplo, la lista de temporizadores 904 puede comprender un temporizador relativo de revisión ortográfica 906 establecido a un tiempo de espera después de 10 minutos, un temporizador absoluto de editor de textos 908 establecido a un tiempo de espera a las 3:00 PM del 12/8/2011, y/u otros temporizadores. El gestor de estado de aplicación 910 puede recibir una notificación 902 de que la aplicación se ha de colocar en un estado de suspensión. Por ejemplo, la notificación 902 puede indicar que una aplicación de editor de textos y/o una aplicación de revisión ortográfica se han de poner en el estado de suspensión.

Tras recibir la notificación 902, el gestor de estado de aplicación 910 puede crear un estado de aplicación 912 que comprende datos de rebasado de temporizador derivados de la lista de temporizadores 904. Por ejemplo, se pueden crear los datos de rebasado de temporizador relativo 914 para el temporizador relativo de revisión ortográfica 906. Los datos de rebasado de temporizador relativo 914 pueden comprender un tiempo hasta el valor de expiración de 8 minutos debido a que 2 de entre los 10 minutos pueden haber transcurrido en el momento en que se suspendió la aplicación de revisión ortográfica y, de este modo, quedan 8 minutos hasta la expiración del temporizador relativo de 10 minutos. Se pueden crear los datos de rebasado de temporizador absoluto 916 para el temporizador absoluto de editor de texto 908. Los datos de rebasado de temporizador absoluto 916 pueden comprender un tiempo de expiración original de 3:00 PM del 12/8/2011. De esta forma, se puede crear el estado de aplicación 912.

Tras recibir la notificación de que la aplicación de editor de textos y/o la aplicación de revisión ortográfica se han de poner en un estado de ejecución, se pueden aplicar los datos de rebasado de temporizador relativo 914 y/o los datos de rebasado de temporizador absoluto 916 dentro del estado de aplicación 912 al temporizador relativo de revisión ortográfica 906 y/o al temporizador absoluto de editor de textos 908. De esta forma, se puede crear un temporizador relativo rebasado de 8 minutos y se puede crear un temporizador absoluto rebasado de 3:00 PM del 12/8/2011.

La Fig. 10 ilustra un ejemplo de un sistema 1000 configurado para gestionar un estado de aplicación de una aplicación, tal como mensajes de sistemas. El sistema 1000 puede comprender un componente de gestión de mensajes 1006. El componente de gestión de mensajes 1006 se puede configurar para mantener una cola de mensajes de aplicación 1004 asociada con una aplicación mientras la aplicación está en estado de suspensión. La cola de mensajes de aplicación 1004 se puede configurar para almacenar mensajes de sistema para la aplicación.

En un ejemplo, una aplicación de editor de textos se puede asociar con la cola de mensajes de aplicación 1004. La aplicación de editor de textos se puede poner en el estado de suspensión. Se puede apreciar que los nuevos mensajes de sistema se pueden poner en cola dentro de la cola de mensajes de aplicación 1004 mientras se suspende la aplicación de editor de textos. Un estado actual de la cola de mensajes de aplicación 1004 puede comprender un mensaje del sistema de modo de vista = paisaje, un mensaje de sistema de interfaz de usuario = verde, un mensaje de sistema de idioma = inglés, un mensaje de sistema de nuevo dispositivo de entrada = ratón y/u otros.

Mientras que la aplicación se suspende, se pueden recibir nuevos mensajes de sistema 1002 (por ejemplo, nuevo mensaje de sistema de modo de vista = retrato, nuevo mensaje de sistema de interfaz de usuario = roja, nuevo mensaje de sistema de nuevo dispositivo de entrada = teclado y/u otros nuevos mensajes de sistema). El componente de gestión de mensajes 1006 se puede configurar para mantener la cola de mensajes de aplicación 1004 a la luz del nuevo mensaje de sistema 1002, de modo que los mensajes de sistema irrelevantes y/o viejos se eliminen y/o modifiquen de la cola de mensajes de aplicación 1004. En un ejemplo, el nuevo mensaje de sistema de modo de vista = retrato se puede determinar como correspondiente al mensaje de sistema de modo de vista = paisaje dentro de la cola de mensajes de aplicación 1004. Debido a que el nuevo mensaje de sistema de modo de vista = retrato puede hacer que el mensaje de sistema de modo de vista = paisaje sea irrelevante y/o viejo, el mensaje de sistema de modo de vista = paisaje se puede eliminar 1012 y el nuevo mensaje de sistema de modo de vista = retrato se puede agregar 1008 a la cola de mensajes de aplicación 1004.

En otro ejemplo, el nuevo mensaje de sistema de interfaz de usuario = roja se puede determinar como correspondiente al mensaje de sistema de interfaz de usuario = verde dentro de la cola de mensajes de aplicación 1004. Debido a que el nuevo mensaje de sistema de interfaz de usuario = roja puede hacer que el mensaje de sistema de interfaz de usuario = verde sea irrelevante y/o viejo, el mensaje de sistema de interfaz de usuario = verde se puede eliminar 1014 y el nuevo mensaje de sistema de interfaz de usuario = roja se puede agregar a la cola de mensajes de aplicación 1004. En otro ejemplo, un nuevo mensaje de sistema de nuevo dispositivo de entrada = teclado se puede determinar como correspondiente al mensaje de sistema de nuevo dispositivo de entrada = ratón. Debido a que el nuevo mensaje de sistema de nuevo dispositivo de entrada = teclado puede ser complementario al mensaje de sistema de nuevo dispositivo de entrada = ratón, el mensaje de sistema de nuevo dispositivo de entrada = ratón se puede modificar 1016 al mensaje de sistema de nuevo dispositivo de entrada = ratón y teclado. De esta forma, la cola de mensajes de aplicación 1004 se puede mantener por el componente de gestión de mensajes mientras que la aplicación está suspendida, de modo que la aplicación, una vez reanudada, puede quitar de la cola los mensajes de sistema meramente relevantes.

Otra realización más implica un medio legible por ordenador que comprende instrucciones ejecutables por procesador configuradas para implementar una o más de las técnicas presentadas en la presente memoria. Un medio legible por ordenador ejemplar que se puede idear de estas formas se ilustra en la Fig. 11, en donde la implementación 1100 comprende un medio legible por ordenador 1116 (por ejemplo, un CD-R, DVD-R, o un disco de una unidad de disco duro), en el que se codifican datos legibles por ordenador 1114. Estos datos legibles por ordenador 1114 a su vez comprenden un conjunto de instrucciones de ordenador 1112 configuradas para operar según uno o más de los principios expuestos en la presente memoria. En una realización 1100 tal, las instrucciones de ordenador ejecutables por procesador 1112 se pueden configurar para realizar un método 1110, tal como al menos alguno del método 100 ejemplar de la Fig. 1, al menos alguno del método 200 ejemplar de la Fig. 2, al menos alguno del método 400 ejemplar de la Fig. 4, al menos alguno del método 500 ejemplar de la Fig. 5, al menos alguno del método 700 ejemplar de la Fig. 7, y/o al menos alguno del método 800 ejemplar de la Fig. 8, por ejemplo. En otra realización tal, las instrucciones ejecutables por procesador 1112 se pueden configurar para implementar un sistema, tal como al menos alguno del sistema 300 ejemplar de la Fig. 3, al menos alguno del sistema 600 ejemplar de la Fig. 6, al menos alguno del sistema 900 ejemplar de la Fig. 9, y/o al menos alguno del sistema 1000 ejemplar de la Fig. 10, por ejemplo. Muchos de tales medios legibles por ordenador se pueden idear por los expertos en la técnica que están configurados para operar según las técnicas presentadas en la presente memoria.

Aunque la materia objeto se ha descrito en un lenguaje específico a las características estructurales y/o los actos metodológicos, se ha de entender que la materia objeto definida en las reivindicaciones adjuntas no está necesariamente limitada a las características o actos específicos descritos anteriormente. Más bien, las características y actos específicos descritos anteriormente se describen como formas de ejemplo de implementación de las reivindicaciones.

Como se usa en esta solicitud, los términos "componente", "módulo", "sistema", "interfaz" y similares se pretende de manera general que se refieran a una entidad relacionada con ordenador, o bien hardware, una combinación de hardware y software, software, o bien software en ejecución. Por ejemplo, un componente puede ser, pero no está limitado a ser, un proceso que se ejecuta en un procesador, un procesador, un objeto, un ejecutable, un subproceso de ejecución, un programa y/o un ordenador. A modo de ilustración, tanto una aplicación que se ejecuta en un controlador como el controlador puede ser un componente. Uno o más componentes pueden residir dentro de un proceso y/o subproceso de ejecución y un componente se puede localizar en un ordenador y/o distribuir entre dos o más ordenadores.

Además, la materia objeto reivindicada se puede implementar como un método, aparato o artículo de fabricación que usa técnicas de programación y/o ingeniería estándar para producir software, microprograma, hardware o cualquier combinación de los mismos para controlar un ordenador para implementar la materia objeto descrita. El término “artículo de fabricación”, como se usa en la presente memoria, se pretende que abarque un programa de ordenador accesible desde cualquier dispositivo, soporte o medios legibles por ordenador. Por supuesto, los expertos en la técnica reconocerán que se pueden hacer muchas modificaciones a esta configuración sin apartarse del alcance o espíritu de la materia objeto reivindicada.

La Fig. 12 y la siguiente discusión proporcionan una breve descripción general de un entorno informático adecuado para implementar realizaciones de una o más de las disposiciones expuestas en la presente memoria. El entorno operativo de la Fig. 12 es solamente un ejemplo de un entorno operativo adecuado y no se pretende que sugiera ninguna limitación en cuanto al alcance de uso o la funcionalidad del entorno operativo. Ejemplos de dispositivos informáticos incluyen, pero no se limitan a, ordenadores personales, ordenadores servidores, dispositivos de mano o portátiles, dispositivos móviles (tales como teléfonos móviles, asistentes digitales personales (PDA), reproductores de medios y similares), sistemas multiprocesadores, electrónica de consumo, mini ordenadores, ordenadores centrales, entornos informáticos distribuidos que incluyan cualquiera de los anteriores sistemas o dispositivos, y similares.

Aunque no se requiere, las realizaciones se describen en el contexto general de “instrucciones legibles por ordenador” que se ejecutan por uno o más dispositivos informáticos. Las instrucciones legibles por ordenador pueden ser distribuidas a través de medios legibles por ordenador (tratado a continuación). Las instrucciones legibles por ordenador se pueden implementar como módulos de programa, tales como funciones, objetos, Interfaces de Programación de Aplicaciones (API), estructuras de datos y similares, que realizan tareas particulares o implementan tipos de datos abstractos particulares. Típicamente, la funcionalidad de las instrucciones legibles por ordenador se puede combinar o distribuir como se desee en diversos entornos.

La Fig. 12 ilustra un ejemplo de un sistema 1210 que comprende un dispositivo informático 1212 configurado para implementar una o más realizaciones proporcionadas en la presente memoria. En una configuración, el dispositivo informático 1212 incluye al menos una unidad de procesamiento 1216 y una memoria 1218. Dependiendo de la configuración exacta y del tipo de dispositivo informático, la memoria 1218 puede ser volátil (tal como RAM, por ejemplo), no volátil (tal como ROM, memoria rápida, etc., por ejemplo) o alguna combinación de las dos. Esta configuración se ilustra en la Fig. 12 mediante la línea discontinua 1214.

En otras realizaciones, el dispositivo 1212 puede incluir características y/o funcionalidad adicionales. Por ejemplo, el dispositivo 1212 también puede incluir almacenamiento adicional (por ejemplo, extraíble y/o no extraíble) incluyendo, pero no limitado a, almacenamiento magnético, almacenamiento óptico y similares. Tal almacenamiento adicional se ilustra en la Fig. 12 mediante el almacenamiento 1220. En una realización, las instrucciones legibles por ordenador para implementar una o más realizaciones proporcionadas en la presente memoria pueden estar en el almacenamiento 1220. El almacenamiento 1220 también puede almacenar otras instrucciones legibles por ordenador para implementar un sistema operativo, un programa de aplicación y similares. Las instrucciones legibles por ordenador se pueden cargar en la memoria 1218 para su ejecución por la unidad de procesamiento 1216, por ejemplo.

El término “medio legible por ordenador” como se usa en la presente memoria incluye medios de almacenamiento informático. Los medios de almacenamiento informático incluyen medios volátiles y no volátiles, extraíbles y no extraíbles implementados en cualquier método o tecnología para el almacenamiento de información, tal como instrucciones legibles por ordenador u otros datos. La memoria 1218 y el almacenamiento 1220 son ejemplos de medios de almacenamiento informáticos. Los medios de almacenamiento informáticos incluyen, pero no se limitan a, RAM, ROM, EEPROM, memoria rápida u otra tecnología de memoria, CD-ROM, Discos Versátiles Digitales (DVD) u otro almacenamiento óptico, casetes magnéticos, cinta magnética, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que se pueda usar para almacenar la información deseada y al que se pueda acceder mediante el dispositivo 1212. Cualquiera de los medios de almacenamiento informático puede ser parte del dispositivo 1212.

El dispositivo 1212 también puede incluir conexión o conexiones de comunicación 1226 que permiten que el dispositivo 1212 comunique con otros dispositivos. La conexión o conexiones de comunicación 1226 pueden incluir, pero no se limitan a, un módem, una Tarjeta de Interfaz de Red (NIC), una interfaz de red integrada, transmisor/receptor de radiofrecuencia, un puerto de infrarrojos, una conexión USB u otras interfaces para conectar el dispositivo informático 1212 a otros dispositivos informáticos. La conexión o conexiones de comunicación 1226 pueden incluir una conexión cableada o una conexión inalámbrica. La conexión o conexiones de comunicación 1226 pueden transmitir y/o recibir medios de comunicación.

El término “medios legibles por ordenador” puede incluir medios de comunicación. Los medios de comunicación típicamente incorporan instrucciones legibles por ordenador u otros datos en una “señal de datos modulada”, tal como una onda portadora u otro mecanismo de transporte, e incluyen cualquier medio de entrega de información. El término “señal de datos modulada” puede incluir una señal que tenga una o más de sus características establecidas o cambiadas de una manera tal para codificar información en la señal.

- 5 El dispositivo 1212 puede incluir un dispositivo o dispositivos de entrada 1224 tales como un teclado, ratón, lápiz, dispositivo de entrada de voz, dispositivo de entrada táctil, cámaras de infrarrojos, dispositivos de entrada de video y/o cualquier otro dispositivo de entrada. También se pueden incluir en el dispositivo 1212 un dispositivo o dispositivos de salida 1222 tales como uno o más visualizadores, altavoces, impresoras y/o cualquier otro dispositivo de salida. El dispositivo o dispositivos de entrada 1224 y el dispositivo o dispositivos de salida 1222 se pueden conectar al dispositivo 1212 a través de una conexión cableada, conexión inalámbrica, o cualquier combinación de las mismas. En una realización, un dispositivo de entrada o un dispositivo de salida de otro dispositivo informático se pueden usar tal como el dispositivo o dispositivos de entrada 1224 o el dispositivo o dispositivos de salida 1222 para dispositivo informático 1212.
- 10 Los componentes del dispositivo informático 1212 se pueden conectar mediante diversas interconexiones, tales como un bus. Tales interconexiones pueden incluir una Interconexión de Componentes Periféricos (PCI), tal como PCI Express, un Bus Serie Universal (USB), firewall (IEEE 1394), una estructura de bus óptico y similares. En otra realización, los componentes del dispositivo informático 1212 se pueden interconectar mediante una red. Por ejemplo, la memoria 1218 puede estar compuesta por varias unidades de memoria física situadas en diferentes ubicaciones físicas interconectadas por una red.
- 15 Los expertos en la técnica se darán cuenta de que los dispositivos de almacenamiento utilizados para almacenar instrucciones legibles por ordenador pueden estar distribuidos a través de una red. Por ejemplo, un dispositivo informático 1230 accesible a través de una red 1228 puede almacenar instrucciones legibles por ordenador para implementar una o más realizaciones proporcionadas en la presente memoria. El dispositivo informático 1212 puede acceder al dispositivo informático 1230 y descargar una parte de o todas las instrucciones legibles por ordenador para su ejecución. Alternativamente, el dispositivo informático 1212 puede descargar partes de las instrucciones legibles por ordenador, según sea necesario, o algunas instrucciones se pueden ejecutar en el dispositivo informático 1212 y algunas en el dispositivo informático 1230.
- 20 Se proporcionan en la presente memoria diversas operaciones de realizaciones. En una realización, una o más de las operaciones descritas pueden constituir instrucciones legibles por ordenador almacenadas en uno o más medios legibles por ordenador, que si se ejecutan por un dispositivo informático, harán que el dispositivo informático realice las operaciones descritas. El orden en el que se describen algunas de o todas las operaciones no se debería interpretar como que implica que estas operaciones son dependientes necesariamente del orden. Un orden alternativo se apreciará por un experto en la técnica que tiene el beneficio de esta descripción. Además, se entenderá que no todas las operaciones están necesariamente presentes en cada realización proporcionada en la presente memoria.
- 25 Además, la palabra “ejemplar” se usa en la presente memoria para indicar que sirve como ejemplo, caso o ilustración. Cualquier aspecto o diseño descrito en la presente memoria como “ejemplar” no se ha de interpretar necesariamente como ventajoso sobre otros aspectos o diseños. Más bien, el uso de la palabra ejemplar se pretende que presente conceptos de una forma concreta. Como se usa en esta solicitud, el término “o” se pretende que signifique un “o” inclusivo en lugar de un “o” exclusivo. Es decir, a menos que se especifique de otro modo, o esté claro a partir del contexto, “X emplea A o B” se pretende que signifique cualquiera de las permutaciones inclusivas naturales. Es decir, si X emplea A; X emplea B; o X emplea tanto A como B, entonces “X emplea A o B” se cumple en cualquiera de los casos anteriores. Además, los artículos “un”, “uno” y “una” como se usan en esta solicitud y las reivindicaciones adjuntas se pueden interpretar de manera general como “uno o más”, a menos que se especifique de otro modo, o esté claro a partir del contexto que se dirige a una forma singular. También, al menos uno de A y B y/o similar significa de manera general A o B o tanto A como B.
- 30 También, aunque la descripción se ha mostrado y descrito con respecto a una o más implementaciones, a otros expertos en la técnica se les ocurrirán alteraciones y modificaciones equivalentes en base a una lectura y comprensión de esta especificación y los dibujos anexos. La descripción incluye todas de tales modificaciones y alteraciones y está limitada solamente por el alcance de las siguientes reivindicaciones. En particular con respecto a las diversas funciones realizadas por los componentes descritos anteriormente (por ejemplo, elementos, recursos, etc.), los términos usados para describir tales componentes se pretende que correspondan, a menos que se indique de otro modo, a cualquier componente que realice la función especificada del componente descrito (por ejemplo, que es funcionalmente equivalente), aún cuando no sea estructuralmente equivalente a la estructura descrita que realiza la función en las implementaciones ejemplares ilustradas en la presente memoria de la descripción. Además, mientras que una característica particular de la descripción se puede haber descrito con respecto a solamente una de varias implementaciones, tal característica se puede combinar con una o más de otras características de las otras implementaciones, como se pueda desear y ser ventajosa para cualquier aplicación dada o particular. Además, en la medida en que los términos “incluye”, “que tiene”, “tiene”, “con”, o variantes de los mismos, se usan o bien en la descripción detallada o bien en las reivindicaciones, tales términos se pretende que sean inclusivos de una manera similar al término “que comprende”.
- 35
- 40
- 45
- 50
- 55

REIVINDICACIONES

1. Un método para suspender uno o más procesos asociados con una aplicación, que comprende:
- recibir (204) una notificación de que una aplicación está inactiva, por lo cual una aplicación está inactiva cuando no está siendo usada;
- 5 caracterizado por:
- atravesar (206) iterativamente una jerarquía de contenedores lógicos para identificar uno o más procesos asociados con la aplicación;
- poner (208) el uno o más procesos en un estado de suspensión;
- 10 identificar un proceso recién creado, por lo cual un proceso recién creado es un proceso creado después de que se suspendan uno o más procesos;
- determinar que el proceso recién creado se asocia con la aplicación en base a determinar que el proceso recién creado se asigna a un contenedor lógico asociado con un contenedor lógico raíz dentro de la jerarquía de contenedores lógicos; y
- poner el proceso recién creado en el estado de suspensión.
- 15 2. El método de la reivindicación 1, que comprende:
- determinar que la aplicación está inactiva en base a al menos uno de:
- determinar que la aplicación no es visible por un usuario;
- determinar que la aplicación no está siendo interactuada por el usuario; y
- determinar que la aplicación está en un estado de segundo plano.
- 20 3. El método de la reivindicación 1, en donde la aplicación se asigna al contenedor lógico raíz dentro de la jerarquía de contenedores lógicos y los procesos asociados con la aplicación se asignan a los contenedores lógicos asociados con el contenedor lógico raíz.
4. El método de la reivindicación 3, la identificación de uno o más procesos que comprende:
- 25 determinar que un primer proceso está asociado con la aplicación en base a determinar que el primer proceso se asigna a un primer contenedor lógico asociado con el contenedor lógico raíz.
5. El método de la reivindicación 4, la identificación de uno o más procesos que comprende al menos uno de:
- determinar que un segundo proceso está asociado con la aplicación en base a determinar que el segundo proceso se asigna al primer contenedor lógico; y
- 30 determinar que el segundo proceso está asociado con la aplicación en base a determinar que el segundo proceso se asigna a un segundo contenedor lógico asociado con el primer contenedor lógico.
6. Un sistema para suspender uno o más procesos asociados con una aplicación, que comprende:
- un componente de suspensión (322) configurado para:
- recibir una notificación de que una aplicación está inactiva, por lo cual una aplicación está inactiva cuando no está siendo usada;
- 35 atravesar iterativamente una jerarquía de contenedores lógicos para identificar uno o más procesos asociados con la aplicación;
- poner el uno o más procesos en un estado de suspensión;
- identificar un proceso recién creado, por lo cual un proceso recién creado es un proceso creado después de que se suspendan uno o más procesos;
- 40 determinar que el proceso recién creado está asociado con la aplicación en base a determinar que el proceso recién creado se asigna a un contenedor lógico asociado con un contenedor lógico raíz dentro de la jerarquía de contenedores lógicos; y
- poner el proceso recién creado en el estado de suspensión.

7. El sistema de la reivindicación 6, el componente de suspensión (322) configurado para:
 - determinar que la aplicación está asociada con el contenedor lógico raíz dentro de la jerarquía de contenedores lógicos; y
 - determinar que un primer proceso está asociado con la aplicación en base a determinar que el primer proceso se asigna a un primer contenedor lógico asociado con el contenedor lógico raíz.
- 5
8. El sistema de la reivindicación 7, el componente de suspensión (322) configurado para:
 - determinar que un segundo proceso está asociado con la aplicación en base a determinar que el segundo proceso se asigna a un segundo contenedor lógico asociado con el primer contenedor lógico.
9. El sistema de la reivindicación 7, el componente de suspensión (322) configurado para:
 - determinar que un segundo proceso está asociado con la aplicación en base a determinar que el segundo proceso se asigna al primer contenedor lógico.
- 10

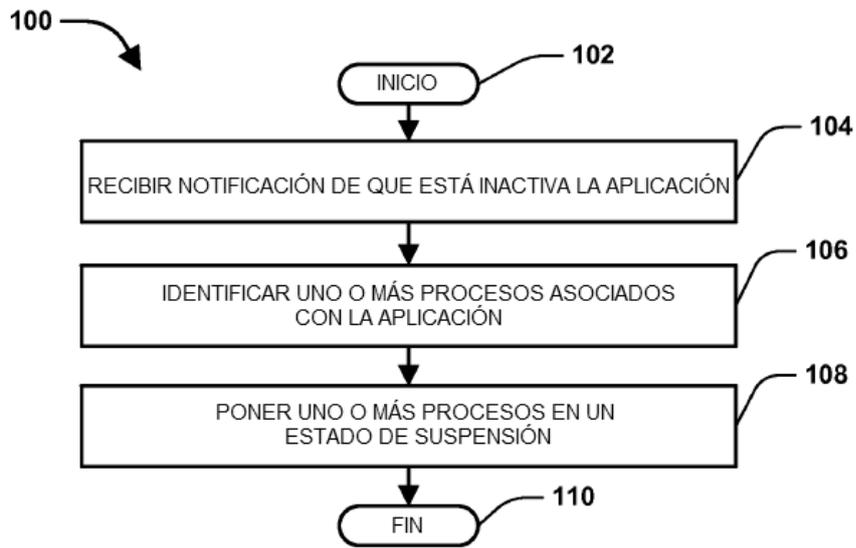


FIG. 1

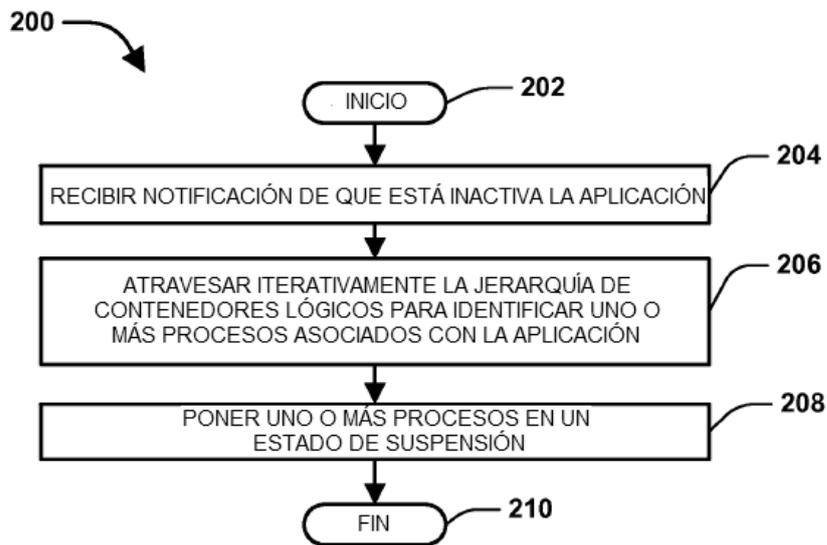


FIG. 2

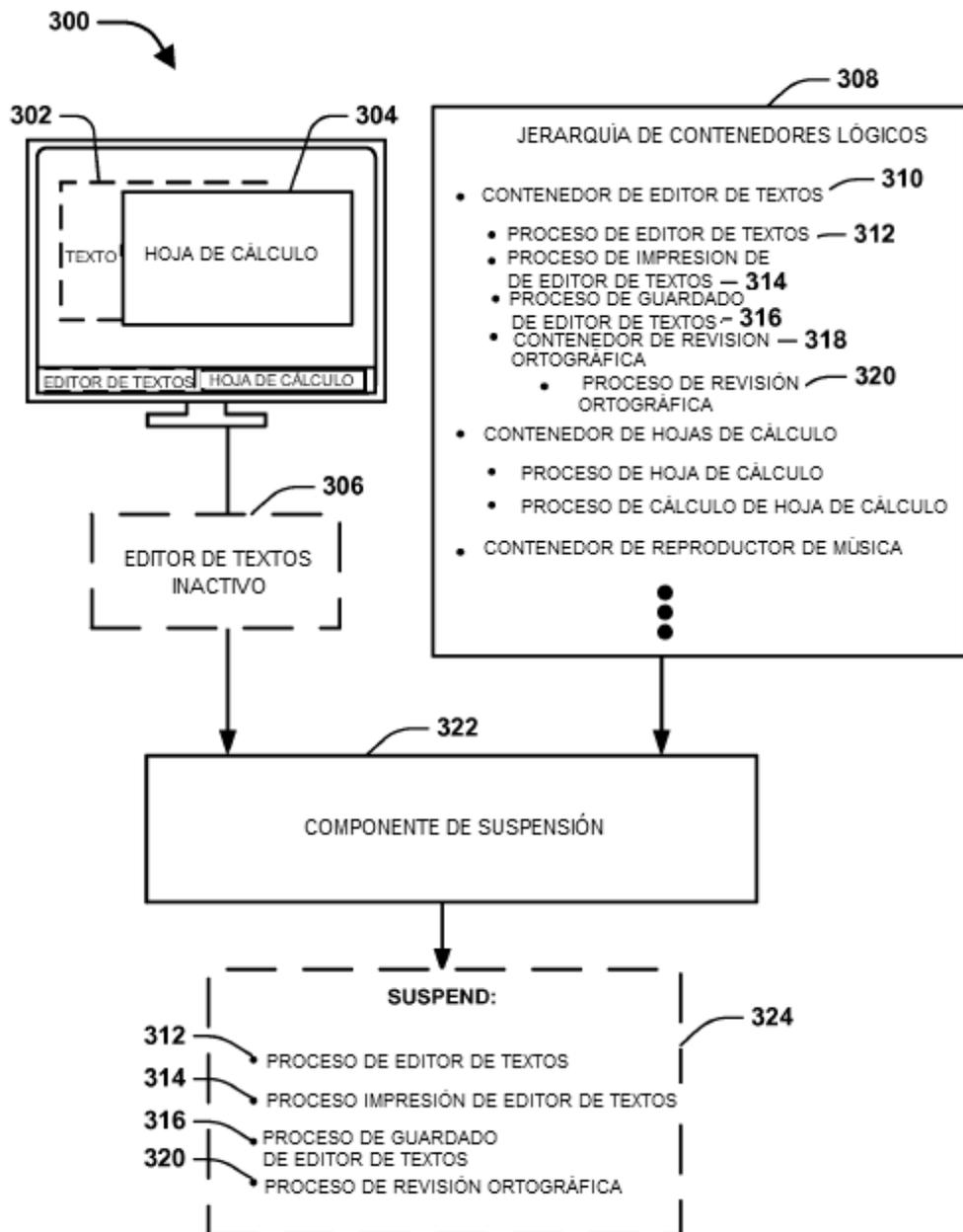


FIG. 3

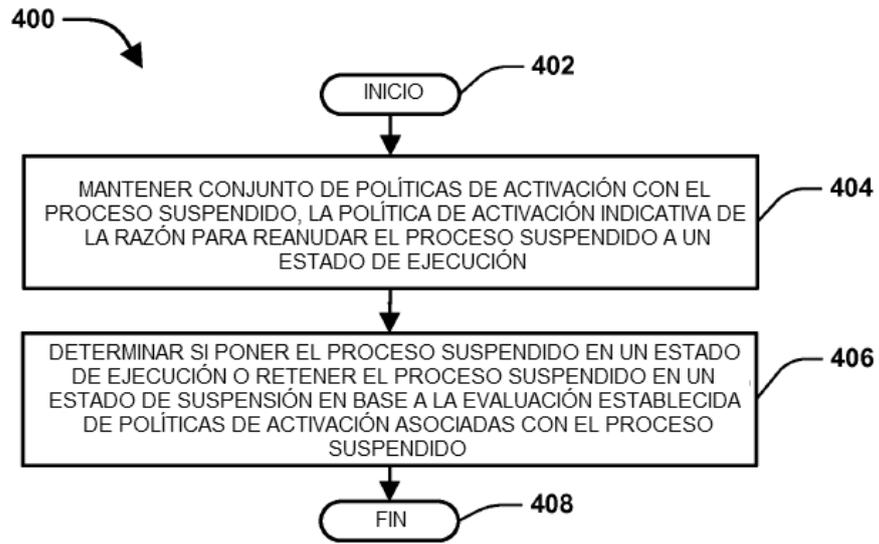


FIG. 4

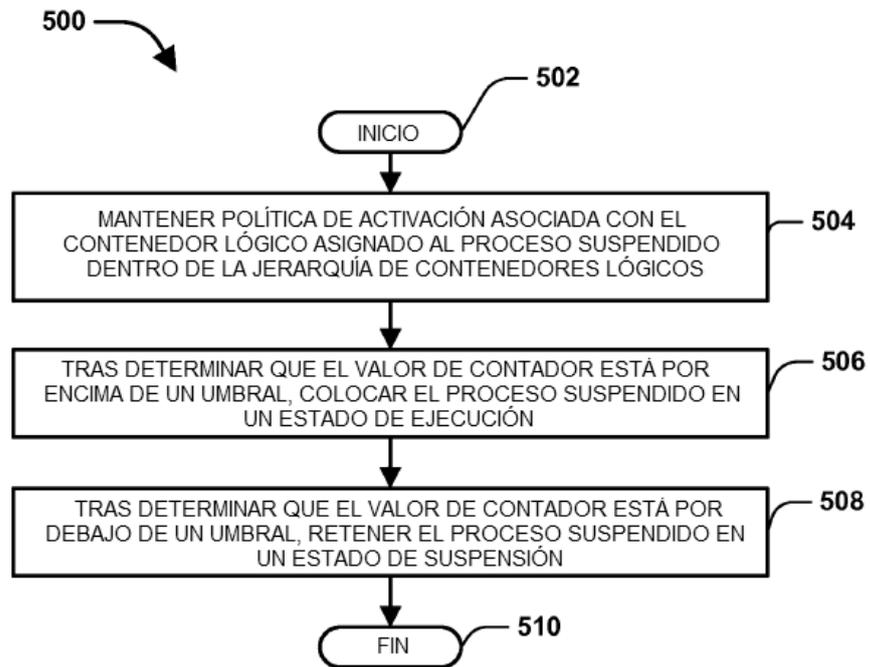


FIG. 5

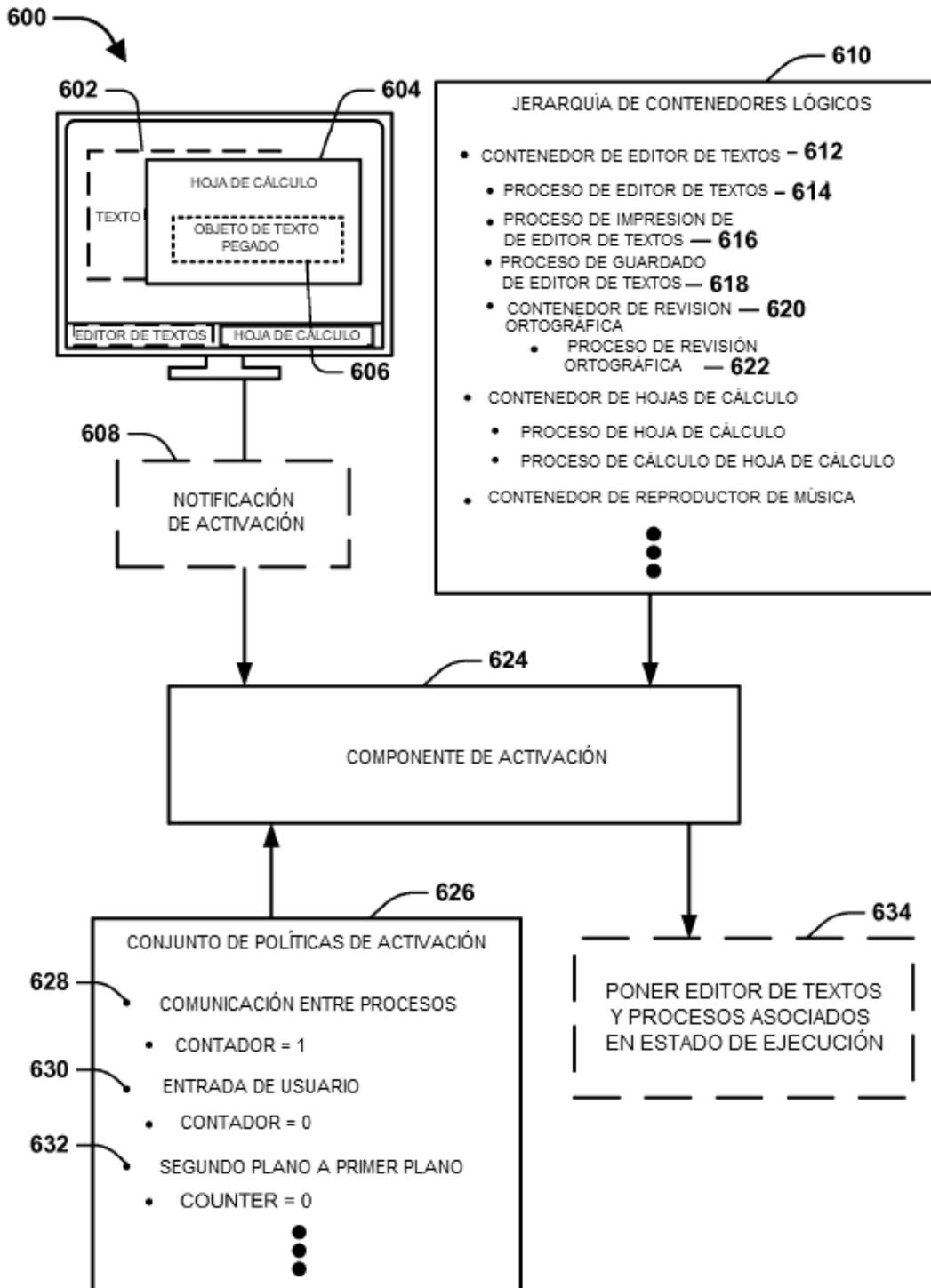


FIG. 6

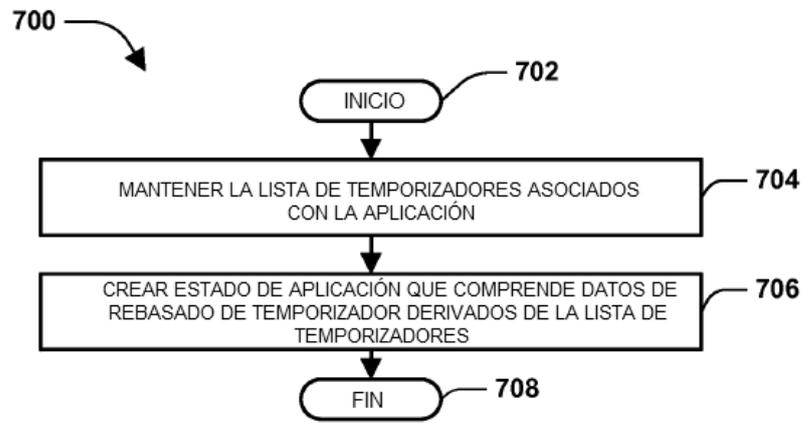


FIG. 7

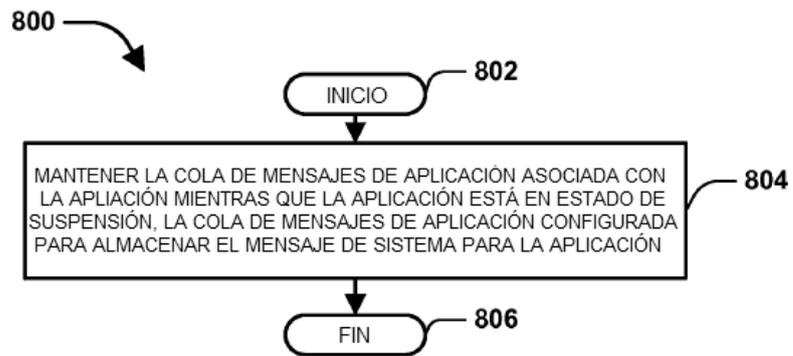


FIG. 8

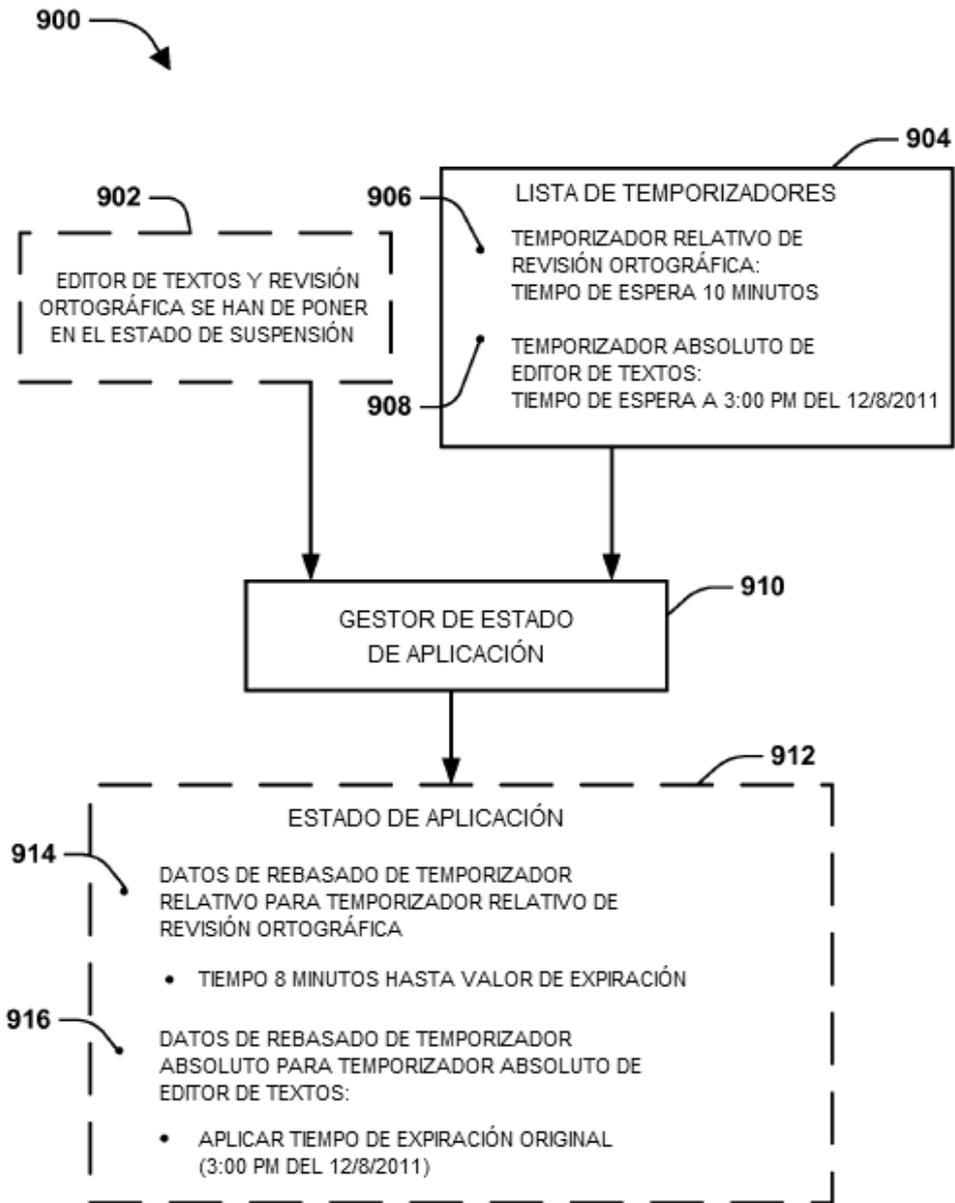


FIG. 9

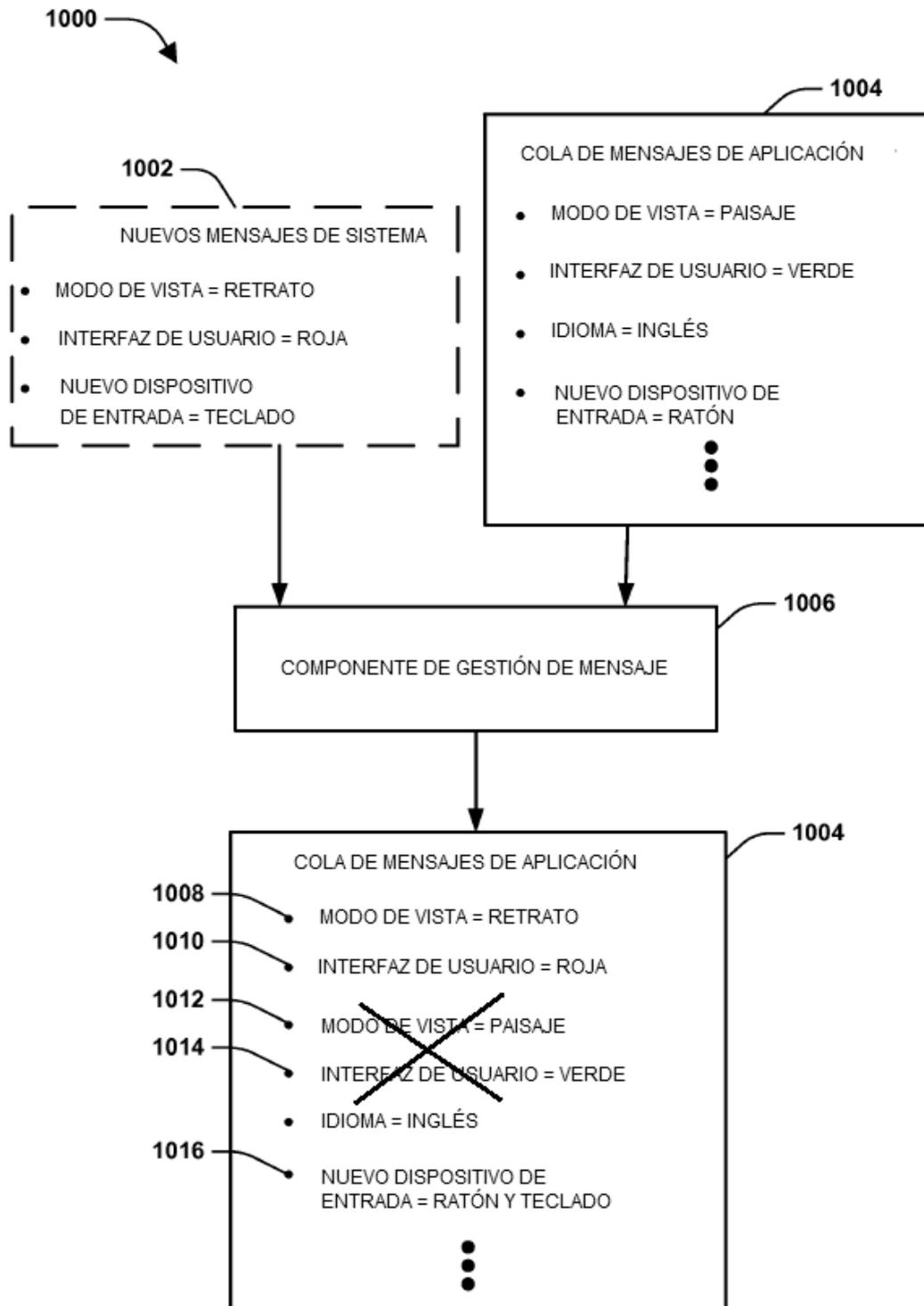


FIG. 9

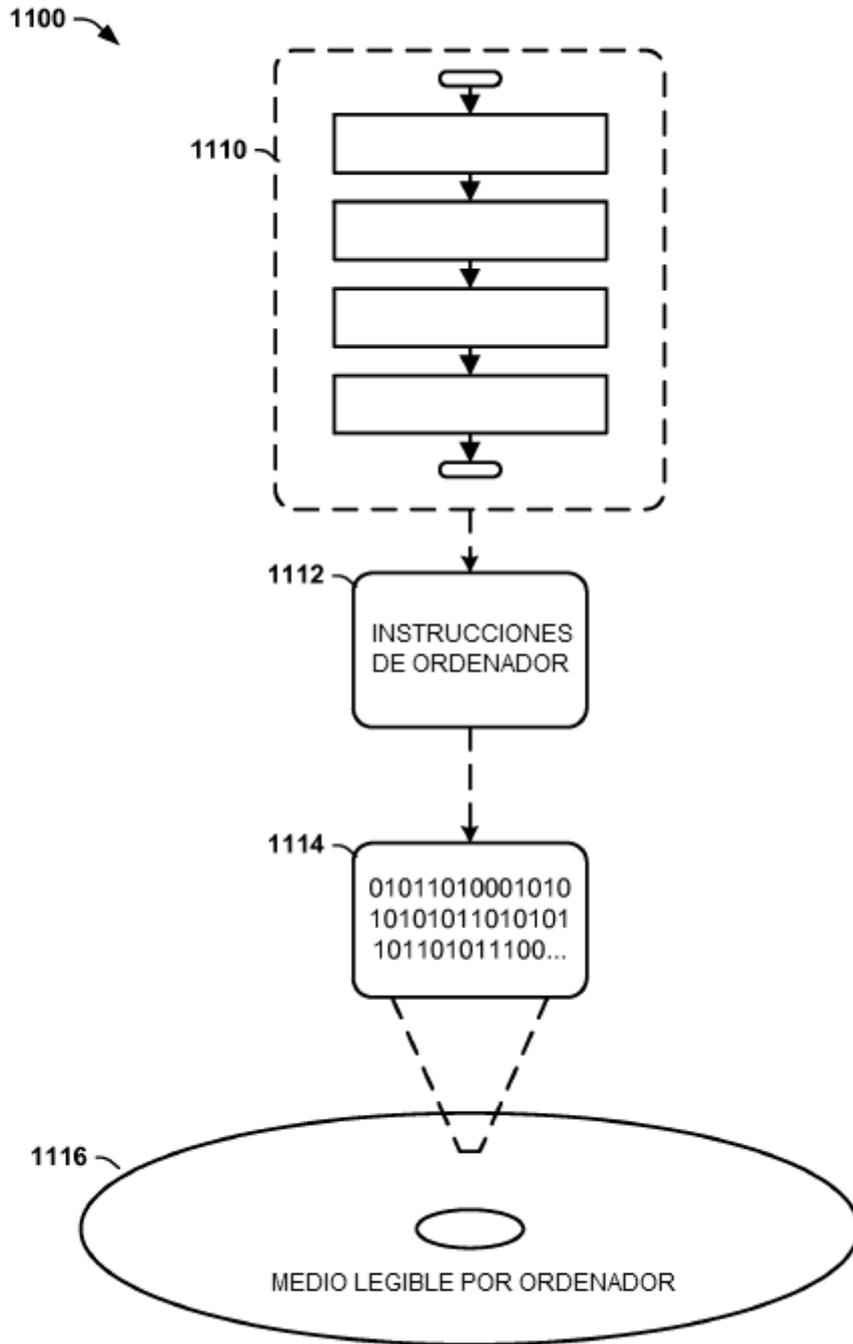


FIG. 11

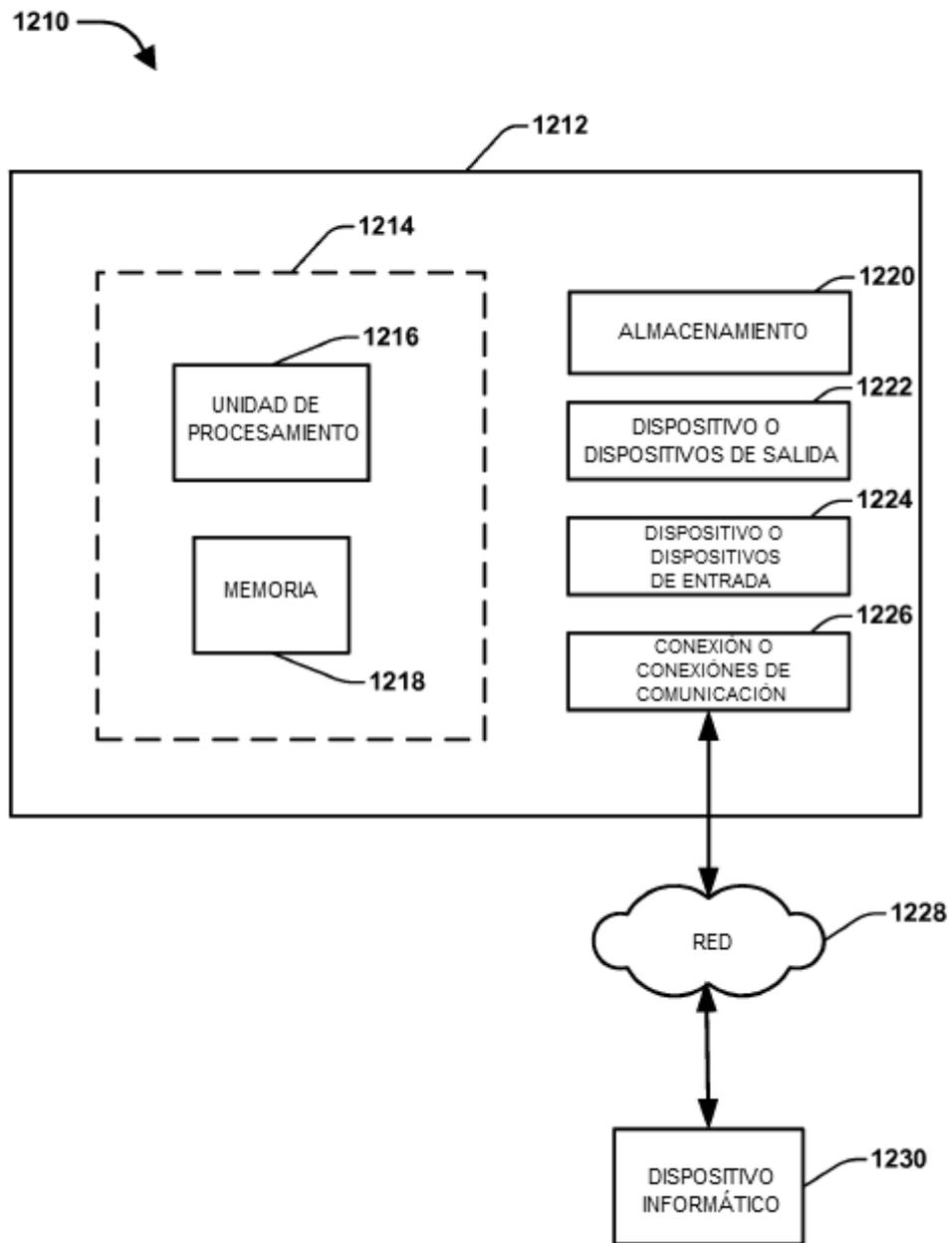


FIG. 12