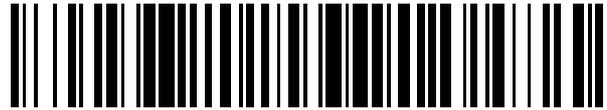


19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 520 941**

51 Int. Cl.:

**G06F 9/46** (2006.01)

**H04L 12/58** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **18.05.2009 E 09305450 (0)**

97 Fecha y número de publicación de la concesión europea: **30.07.2014 EP 2254046**

54 Título: **Método y sistema para gestionar el orden de mensajes**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**12.11.2014**

73 Titular/es:

**AMADEUS S.A.S. (100.0%)  
485 Route du Pin Montard, Les Bouillides, BP 69  
06902 Sophia Antipolis Cedex, FR**

72 Inventor/es:

**CADORET, MARION;  
PENAUD, AGNES y  
POULOUIN, DAVID**

74 Agente/Representante:

**SUGRAÑES MOLINÉ, Pedro**

**ES 2 520 941 T3**

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

## DESCRIPCIÓN

Método y sistema para gestionar el orden de mensajes

5 Campo de la invención

La presente invención se refiere a un método y a un sistema para gestionar el orden de los mensajes, particular pero no exclusivamente con respecto a la secuenciación de mensajes a sistemas externos en un entorno de múltiples procesadores.

10

Antecedentes de la invención

Existen sistemas de mensajería en una multitud de dominios diferentes y se formulan en una multitud de maneras diferentes. Los mensajes están relacionados a menudo con otros mensajes que o bien han salido antes o bien todavía están por llegar, en otras palabras a menudo hay una secuencia para los mensajes. Debido a la manera en la que ahora funcionan muchos sistemas de mensajería, no es raro que se reciban mensajes en un entorno de procesamiento central fuera de secuencia. Éste es el caso particularmente en entornos de múltiples procesadores en los que los procesos pueden producirse a velocidades diferentes, lo que conduce a una pérdida de secuenciación. Entonces, a menudo se requiere que estos mensajes se transmitan a un destino. Algunos destinos no pueden hacer frente a mensajes que se reciben fuera de secuencia puesto que entonces no pueden procesar los mensajes apropiadamente.

Hay muchos sistemas de mensajería conocidos, que intentan superar algunos de los problemas identificados anteriormente. Por ejemplo, el documento WO 2007/062510 describe un sistema que usa un secuenciador para procesar mensajes de manera concurrente. Los mensajes tienen un identificador único y un indicador de secuencia. En este sistema, las condiciones son tales que es posible determinar a priori un identificador de secuencia para el o cada mensaje. Esto facilita el uso de afinidad: los mensajes que comparten el mismo identificador de secuencia se distribuyen mediante el mismo proceso. En el documento WO2007/062510, la secuenciación se gestiona bloqueando los mensajes en una cola de salida.

El documento US 2007/0124398 da a conocer un entorno de múltiples procesadores en el que un sistema usa una combinación de identificador de secuencia y número de secuencia para garantizar que los mensajes se procesan mediante el mismo recurso. Este método requiere hilos de mensajes en espera que requieren acuses de recibo antes de que puedan procesarse. También existe una elección de cola dependiendo del número de secuencia y las colas pueden ordenarse.

El documento US 2003/110230 da a conocer un entorno de procesamiento paralelo en el que se mantiene la ordenación de mensajes procesando mensajes desde la misma fuente con un recurso común o hilo común. De nuevo, esto requiere el uso de hilos y procesos en espera o una cola ordenada de mensajes para su transmisión posterior.

El documento US 2006/0146848 da a conocer un método y un sistema para poner en cola mensajes con el fin de procesar mensajes en un orden predeterminado. El sistema comprende un identificador único para un grupo de mensajes. El sistema también incluye mensajes adicionales para determinar si los mensajes se procesaron satisfactoriamente y un parámetro de ID de correlación para correlacionar grupos separados de mensajes.

El documento US2003/0158883 da a conocer un método para procesar mensajes en una cola en el que cada mensaje tiene un identificador en relación con el origen del mensaje. El método también da a conocer una etapa de limitar la cantidad del tiempo de procesamiento para optimizar la gestión de los mensajes en la cola.

Los métodos de la técnica anterior mencionados anteriormente funcionan en cierta medida, aunque dependen mucho de poner en cola mensajes lo que conduce a una complejidad y al uso de grandes cantidades de memoria y potencia de procesamiento. El principal problema con este sistema es el mantenimiento y el hecho de que no siempre es posible que la secuenciación se base en una correlación de secuencia única. Las dependencias pueden ser mucho más complicadas que sólo el orden de un número de secuencia dentro del mismo identificador de secuencia. Los métodos anteriores funcionan pero presentan problemas de mantenimiento y fiabilidad. Tener un hilo en espera en una memoria no es persistente: así, si se produce un problema en la máquina se perderá la secuenciación. La gestión de la secuenciación en colas ordenadas da lugar a un problema de escalabilidad. Es más difícil añadir más colas, o debe haber afinidad entre procesos y colas. La afinidad también es más compleja de gestionar. Añade un punto de fallo porque no hay redundancia en términos de servicio. Además, la técnica anterior no enseña un historial centralizado, por tanto, no es posible manejar una secuenciación compleja. En otras palabras, cuando no hay un identificador de secuencia único.

65 Sumario de la invención

La presente invención proporciona un método y un sistema tal como se expone en las reivindicaciones adjuntas.

Según una realización, se proporciona un método para ordenar una pluralidad de mensajes recibidos de un remitente que van a enviarse a un receptor en una secuencia equivalente a un orden de mensajes de la pluralidad de mensajes en el que el orden de mensajes se determina en tiempo real, comprendiendo el método las etapas de: recibir uno o más mensajes de un flujo de mensajes y almacenarlos en una base de datos; identificar una característica (orden de clave P) de cada mensaje que es común a un grupo de mensajes; identificar una secuencia que corresponde al orden de mensajes (1, 2, 3, 4, etc.) para el grupo de mensajes a partir de un parámetro del mensaje; revisar un mensaje almacenado particular en la base de datos para determinar si el mensaje almacenado puede enviarse; determinando si existe un mensaje previo en la secuencia antes del mensaje almacenado y determinando un estado del mensaje previo; actualizando el estado del mensaje almacenado basándose en el estado del mensaje previo; y enviar el mensaje almacenado según la secuencia una vez que se ha enviado el mensaje previo.

La presente invención proporciona un sistema muy escalable, que puede extenderse fácilmente para hacer frente a un aumento de tráfico. La presente invención también garantiza que los mensajes se envíen en el orden correcto en un entorno de múltiples procesadores. Las dependencias de los mensajes son estrictamente las requeridas. El sistema no introduce ninguna dependencia. La ordenación no es total: los mensajes que son independientes pueden enviarse en paralelo. Por tanto, si falla un enlace de receptor, no influirá sobre ningún otro receptor a menos que los mensajes estén enlazados funcionalmente. Además, el uso de los órdenes correctos es muy importante puesto que la desincronización entre procesadores diferentes puede conducir a muchos problemas. Por ejemplo, en el entorno de una línea aérea, si hay problemas en relación con la sincronización de los mensajes, a los pasajeros se les podría denegar un asiento cuando de hecho hay espacio disponible en la aeronave. Esto conduce a una pérdida de dinero en la línea aérea y también a otras desventajas.

El proceso según la presente invención se proporciona mediante una implementación sencilla puesto que los hilos y la afinidad no forman parte del proceso. En particular, no hay afinidad entre ningún elemento en cola y de procesamiento del sistema puesto que no están presentes. Además, hay una manera flexible en la que pueden procesarse errores. Esto es particularmente importante cuando se maneja un mensaje que falta en una secuencia. Usando una base de datos, se mantiene un historial completo de los mensajes y las transmisiones durante una ventana de tiempo específica. La ventana de tiempo también proporciona un umbral. Antes del umbral, el mensaje puede estar en un estado y después el estado puede cambiar. El umbral es particularmente útil para temas de mantenimiento y para reducir la cantidad de espacio requerido para almacenar mensajes e información asociada con los mismos. Además, se minimiza la puesta en cola y se hace más fácil la escalabilidad.

Según una segunda realización, se proporciona un sistema de mensajes para ordenar una pluralidad de mensajes recibidos de un remitente que van a enviarse a un receptor en una secuencia equivalente a un orden de mensajes de la pluralidad de mensajes, comprendiendo el método las etapas de: un módulo receptor para recibir uno o más mensajes de un flujo de mensajes y almacenarlos en una base de datos; un módulo de gestión de mensajes para identificar una característica (orden de clave P) de cada mensaje que es común a un grupo de mensajes; para identificar una secuencia que corresponde al orden de mensajes (1, 2, 3, 4, etc.) para el grupo de mensajes a partir de un parámetro del mensaje; y para revisar un mensaje almacenado particular en la base de datos para determinar si el mensaje almacenado puede enviarse: determinando si existe un mensaje previo en la secuencia antes del mensaje almacenado y determinando un estado del mensaje previo; y actualizando el estado del mensaje almacenado basándose en el estado del mensaje previo; un módulo de transmisión para enviar el mensaje almacenado según la secuencia una vez que se ha enviado el mensaje previo.

Breve descripción de los dibujos

Ahora se hará referencia, a modo de ejemplo, a los dibujos adjuntos, en los que:

la figura 1 es una representación sencilla de un sistema, según una realización de la invención, proporcionada a modo de ejemplo,

la figura 2 es una representación de un caso normal de secuenciación de mensajes, según una realización de la invención, proporcionada a modo de ejemplo,

la figura 3 es una representación detallada del sistema de la figura 1, según una realización de la invención, proporcionada a modo de ejemplo,

la figura 4 es parte de una tabla para identificar la nomenclatura usada según una realización de la invención, proporcionada a modo de ejemplo,

la figura 5 es un diagrama de flujo que muestra la generación de mensajes, según una realización de la invención, proporcionada a modo de ejemplo,

la figura 6 es un diagrama de flujo que muestra la secuencia de cálculo para un mensaje, según una realización de la invención, proporcionada a modo de ejemplo,

la figura 7 es un diagrama de flujo que muestra el “estado V” para un mensaje, según una realización de la invención, proporcionada a modo de ejemplo,

5 la figura 8 es un diagrama de flujo que muestra la puesta en cascada y secuenciación de una rellamada, según una realización de la invención, proporcionada a modo de ejemplo,

la figura 9 es una tabla de dependencias de mensajes, según una realización de la invención, proporcionada a modo de ejemplo,

10 la figura 10 es un ejemplo de un sobre de registro de nombre de pasajero, según una realización de la invención, proporcionada a modo de ejemplo,

la figura 11 es un diagrama de flujo que muestra que las necesidades de secuenciación dividen en primer lugar el entorno del caso, según una realización de la invención, proporcionada a modo de ejemplo; y

15 la figura 12 es un árbol de dependencias para un único objetivo, según una realización de la invención, proporcionada a modo de ejemplo.

#### Descripción detallada de las realizaciones preferidas

20 En el nivel más alto, la presente invención se refiere a la ordenación de mensajes de un sistema a otro con el fin de garantizar la sincronización. Esto se muestra a modo de ejemplo en la figura 1. La figura 1 muestra un sistema de reserva (sistema A) 100 que está conectado a un módulo de gestión de mensajes 102. El módulo de gestión de mensajes gestiona los mensajes entrantes tal como se describirá en mayor detalle a continuación, antes de enviar mensajes sincronizados a un sistema de línea aérea (sistema B) 104. El sistema de reserva 100 puede ser de cualquier tipo apropiado e incluye, a modo de ejemplo, un sistema de distribución global (GDS) y otros recursos centrales de cualquier otro tipo apropiado. El sistema de reserva puede hacerse funcionar mediante un proceso web en línea tal como un navegador, agentes de viaje o cualquier otra entidad apropiada. El sistema de línea aérea 104 puede estar asociado con una o más líneas aéreas. El módulo de gestión de mensajes 102 incluye varios procesos diferentes tales como filtrado, ordenación, formateo, etc.

En cualquier tipo de sistema de mensajería puede haber un requisito para que los mensajes satisfagan una necesidad de secuenciación particular. Éste es el caso particularmente en un entorno en el que los clientes compran billetes de línea aérea en una línea aérea de cliente apropiada. En el caso del ejemplo de comprar billetes de línea aérea, los mensajes se denominan registros de nombres de pasajeros (PNR). Un registro de nombre de pasajero incluye todos los detalles de un viaje específico para un pasajero incluyendo gastos extraordinarios y cualquier servicio adicional que el pasajero haya solicitado, junto con cualquier cambio o modificación para el viaje que pueda producirse.

40 La figura 2 muestra un ejemplo de por qué la secuenciación de mensajes es importante. La figura 2 muestra el módulo de gestión de mensajes 102 y un flujo entrante de mensajes 200. El flujo de mensajes incluye mensajes con relación a dos registros diferentes. Los mensajes con relación al primer registro están representados por líneas continuas y los mensajes con relación al segundo registro están representados por líneas discontinuas. Para cada registro hay una secuencia particular de mensajes representada por los números 0, 1, 2, 3 y 4. La secuencia puede basarse en el orden de creación de los mensajes aunque el orden puede ser o bien absoluto (0, 1, 2, 3, 4) o bien relativo ((0,), (1, 0), (2,1), (4,2), ...). Si el mensaje 202 añade un vuelo a un registro particular y el mensaje 204 cancela ese mismo vuelo del mismo registro, es necesario que el cliente de la línea aérea 206 reciba el mensaje 202 antes del mensaje 204 para que la secuencia o el contenido tenga sentido. Si el cliente de la línea aérea ve el mensaje 204 antes del mensaje 202, la línea aérea intentará cancelar un vuelo que todavía no se ha reservado según la secuencia de mensajes. El módulo de gestión de mensajes 102 garantiza que el mensaje 202 se recibe por la línea aérea de cliente 206 antes del mensaje 204 independientemente del orden en el que los mensajes llegan al módulo de gestión de mensajes. En otras palabras, el orden de mensajes se calcula “sobre la marcha” sin referencia al orden entrante. Esto se describirá ahora más abajo en mayor detalle.

55 Haciendo referencia a la figura 3, se muestra un diagrama más detallado del módulo de gestión de mensajes. El módulo de gestión de mensajes recibe una entrada Pi 300, que se genera en uno o más flujos de entrada mediante un módulo de flujo de entrada 302. Los mensajes que forman Pi pueden estar en un orden no definido debido al procesamiento paralelo u otros temas, aunque los mensajes se generan por un sistema fiable y se supone en este caso que no se pierde ningún mensaje. Los flujos de entrada se usan entonces por un módulo generador de mensajes 304 para generar los mensajes incluidos en el flujo de entrada. Los mensajes se almacenan entonces en un módulo de almacenamiento 306. El módulo generador de mensajes también está conectado a un temporizador 308 y a un módulo de flujo de salida 310. El propósito del temporizador es garantizar que después de un periodo de tiempo dado, un mensaje dado experimentará un reprocesamiento si otros eventos no lo han provocado de antemano. El módulo de flujo de salida 310 está conectado a un módulo de envío de mensajes 312 que emite una salida 314 a un sistema externo 316. El módulo de envío de mensajes también puede recibir un acuse de recibo 318 del sistema externo 316 para indicar que se ha recibido la salida (generalmente un mensaje particular). El acuse de

recibo se devuelve al módulo de almacenamiento tal como se muestra por la flecha 320. El acuse de recibido se usa para determinar qué mensajes deben enviarse a continuación, en caso de haberlos. Los números en las casillas 1 a 7 corresponden a las etapas en las figuras 5 y 8; éstos se describirán en mayor detalle a continuación.

5 Con el fin de entender más completamente las etapas del proceso que se describirán a continuación, es importante entender varios puntos clave asociados con los mensajes: la naturaleza y el tipo de los propios mensajes, su nomenclatura y su relación con otros mensajes.

10 La invención se refiere a mensajes que tienen una dependencia de uno o más mensajes. Se considera que un mensaje que tiene una dependencia de otro forma parte de una familia de mensajes. En la mayoría de los casos, los mensajes llegarán al módulo de gestión de mensajes 102 en aproximadamente el orden correcto. Sin embargo, en entornos de múltiples procesamientos no es raro que los mensajes se reciban desordenados. Éste es uno de los problemas abordados por la presente invención. Cuando el mensaje está listo para enviarse, se enviará lo más rápido posible, independientemente de cualquier otro mensaje. Esto supone que el mensaje en cuestión se ha liberado tal como se describirá a continuación. Cuando no puede enviarse un mensaje inmediatamente, generalmente hay un motivo para ello. Un posible motivo es que todavía no se ha recibido un mensaje delante del mensaje en cuestión. Un motivo alternativo es que se ha enviado el mensaje previo y todavía no se ha recibido un acuse de recibo. También puede haber circunstancias en las que no se ha determinado completamente el orden de mensajes; en este caso, puede ser que los mensajes tengan que esperar hasta que se determine el orden correcto.

20 En cualquier situación en la que un mensaje no puede enviarse inmediatamente, puede haber un criterio adicional (por ejemplo uso de un temporizador) que determine la planificación del mensaje en cuestión. Por ejemplo, si no se ha recibido un mensaje previo y hay un "hueco en el número de secuencia".

25 Los mensajes emplean un tipo específico de nomenclatura. Esto se usa junto con diversas referencias y dependencias en una base de datos para determinar el orden o la secuencia de los mensajes en el módulo de gestión de mensajes. Haciendo referencia a la figura 4, ahora se describirán determinados aspectos de la nomenclatura de mensaje. La tabla en la figura 4 muestra el número de línea en la base de datos, una referencia de mensaje y cualquier dependencia. La base de datos muestra dos mensajes: M9(j) y M10(j). Si el mensaje M10(j) depende de M9(j), entonces cuando se recibe un acuse de recibo de la recepción del mensaje M9(j), el proceso consulta la base de datos para encontrar cualquier línea de dependencia almacenada que dependa de M9(j) para determinar si hay alguna información de dependencia. De esta manera, se identifica la línea M10(j). En el ejemplo, la línea N+1 tiene una dependencia de la línea N, por tanto una vez que se acusa recibo de M9(j), ahora puede enviarse la línea N+1 (correspondiente a M10(j)).

35 La nomenclatura adicional se refiere al origen y destino de los mensajes. A este respecto, "i" es la etiqueta o índice de secuencia original y "j" es el objetivo de mensaje. Un generador de mensajes envía mensajes indicados por  $P_i$ , donde  $i$  indica el orden de los mensajes. No se conoce el objetivo  $j$  cuando se recibe un mensaje en el sistema. El sistema relacionado con la presente invención recibe los mensajes  $P_i$  y genera los mensajes  $M_i(j)$  donde  $i$  es el orden del mensaje (igual que  $P_i$ ) y  $j$  es el usuario final que debe recibir el mensaje  $M_i(j)$ , el usuario final también se conoce como objetivo o cliente.

40 En el entorno en el que se supone que no se pierde ningún mensaje,  $P_i$  conduce a la generación de mensajes  $M_i(1)$ ,  $M_i(2)$ , ...  $M_i(n)$ .  $P_i$  da lugar a una familia de mensajes definidos por una orden de clave  $P$  que tiene que seguir el orden definido por  $i$  y por el objetivo o cliente  $j$ . A continuación se expone la manera en la que pueden enviarse los mensajes. Se reciben familias de mensajes  $P_0$ ,  $P_1$ ,  $P_2$  y se supone que hay dos objetivos:  $j=0$  y  $1$ . Podrían enviarse mensajes en uno de los siguientes órdenes  $M_0(0)$ ,  $M_0(1)$ ,  $M_1(0)$ ,  $M_1(1)$ ,  $M_2(0)$ ,  $M_2(1)$ ; o  $M_0(0)$ ,  $M_1(0)$ ,  $M_0(1)$ ,  $M_2(0)$ ,  $M_1(1)$ ,  $M_2(1)$ . En cualquier caso, se respeta el orden de mensajes puesto que cada objetivo recibe los mensajes en el orden correcto. Puede haber muchos otros órdenes que también serían aceptables dependiendo del número de mensajes, objetivos, etc. A modo de ejemplo, la siguiente secuencia de mensajes es incorrecta para el objetivo 1 y correcta para el objetivo 0 y 2:  $M_1(1)$ ,  $M_0(0)$ ,  $M_1(0)$ ,  $M_0(1)$ ,  $M_2(0)$ ,  $M_2(1)$ . A continuación se describirá en mayor detalle el método para ordenar los mensajes.

45 La presente invención se refiere al método y al sistema para procesar mensajes para garantizar que llegan a un objetivo en el orden correcto. Esto se consigue por medio de un proceso de secuenciación basándose en una implementación de software y una base de datos. La base de datos tiene forma de tabla persistente, tal como se describió anteriormente, que contiene datos históricos para llevar a cabo el proceso de secuenciación tal como ya se describió anteriormente. Los datos históricos pueden incluir lo siguiente:

- 50 • un número de identificación (por ejemplo un número de línea);
- 60 • una clave: orden de clave  $P$ , por ejemplo AAAAAA; BBBBBB; etc.;
- una etiqueta de secuencia numérica ( $i$ );
- 65 • un objetivo ( $j$ );

- un identificador de dependencias DEP-id que permite una referencia cruzada correspondiente a un número de identificación de otra línea;

5 • puede enviarse (S) un campo de estado (por ejemplo, Mi(j), o anularse (V) si no hay ningún mensaje para el objetivo, o (D) cuando depende de otro mensaje, N cuando no depende (= listo para serlo) y U cuando es desconocido cuando la dependencia todavía no se ha identificado, cuando falta un mensaje correspondiente;

- el mensaje generado Mi(j);

10 • otros datos: por ejemplo mensajes de advertencia para manejo de errores, etc., si se requiere.

El ejemplo descrito anteriormente se basa en la situación ideal en la que no se pierde ningún mensaje. En esta situación, los mensajes pueden retardarse el tiempo necesario para esperar a los mensajes dependientes. Si un mensaje no tiene ninguna dependencia, se le asigna el estado N. Esto significa que el mensaje no tiene ninguna dependencia y puede ponerse en cola inmediatamente para su transmisión al sistema externo. Una vez que se le ha proporcionado al mensaje el estado N y se pone en cola para enviarse al objetivo, el estado del mensaje puede fijarse a S. El estado S significa que el mensaje se ha enviado satisfactoriamente y que ahora pueden liberarse todos los mensajes que dependen de este mensaje. Esto significa que los mensajes posteriores cambiarán de estado de D a N (en el orden relevante) y se pondrán en cola para una transmisión de salida en forma de cascada al objetivo.

En realidad, pueden perderse algunos mensajes, particularmente en sistemas complejos y/o en el caso de problemas de enlace, en caso de que el sistema colapse u otras situaciones. En estas circunstancias, el método descrito anteriormente puede adaptarse para incluir cualquier mecanismo de "mejores esfuerzos". Un ejemplo de un mecanismo de mejores esfuerzos es mediante el uso del temporizador 308 en la figura 1.

El uso del temporizador o cualquier técnica restante equivalente es una parte importante de la presente invención. Esto se debe al hecho de que en arquitecturas de múltiples procesadores pueden perderse o retardarse determinados mensajes, lo que puede afectar a la secuencia de mensajes que van a transmitirse al cliente. Con el fin de volver a tratar los mensajes que no pueden enviarse inmediatamente, deben almacenarse en primer lugar durante un periodo de tiempo. Esto permite que se reciban mensajes nuevos en el módulo generador de mensajes 304 lo que puede influir en cualquier mensaje retardado. Después de pasar por el temporizador 308, el mensaje se devuelve al módulo generador de mensajes para reprocesarse según el proceso de secuenciación (que se describe a continuación). Si se han recibido mensajes mientras que el mensaje retardado ha estado en el temporizador, puede proporcionarse un estado diferente al mensaje retardado y, por tanto, puede ser posible una acción diferente. Por ejemplo, el estado puede cambiar, si se ha recibido un mensaje del que depende el mensaje retardado. El mensaje que entra potencialmente en el temporizador es un mensaje nuevo que no ha tenido una dependencia determinada. Se revisará cualquier mensaje en el temporizador para actualizar el estado (a D o N, por ejemplo). Si no hay ningún cambio de estado de este tipo, el mensaje permanecerá en el temporizador. Puede ajustarse la duración del retardo al proceso particular y/o al número de mensajes que se reciben durante un determinado retardo de tiempo. La duración del retardo también puede estar enlazada a la ventana activa en la tabla persistente de la base de datos. Esta ventana activa se describe más abajo en mayor detalle con referencia a la figura 10.

En un determinado punto en el tiempo, puede enviarse el mensaje retardado, independientemente de su estado. Alternativamente, puede tomarse una decisión para borrar el mensaje, o puede llevarse a cabo otra acción tal como el envío de una advertencia a una línea directa, etc. El resultado preciso dependerá de los ajustes del sistema y la lógica empresarial predeterminada.

Haciendo referencia a la figura 5, ahora se describirá un proceso para la generación de mensajes que tiene en cuenta el hecho de que pueden perderse mensajes. Los números en las casillas corresponden a las flechas en la figura 3. Inicialmente, en la etapa 500, se recibe la entrada Pi y luego se pone en cola por medio de un proceso o conocido como demonio. Estos demonios se denominan generalmente, según la presente invención, módulos generadores de mensajes. Estos módulos generadores de mensajes generarán entonces mensajes Mi(j) para su transmisión posterior a un objetivo en la etapa 502. Los datos relacionados con el orden de clave P y el objetivo (j) se recuperan de la base de datos y se bloquean en la etapa 504. Todas las líneas que contienen una clave P se bloquean en la base de datos hasta que se produce la transmisión. Esto permite una lectura coherente de la base de datos e impide actualizaciones concurrentes en los datos de secuenciación. En la etapa 506, se lleva a cabo un análisis de secuencias para calcular la secuencia de los mensajes. Este proceso se describirá en mayor detalle con referencia a la figura 6. En la etapa 508, la base de datos se actualiza en aquellas filas correspondientes a las que se recuperaron anteriormente en la etapa 504 y se libera el bloqueo. Esta actualización de las filas identifica las dependencias de los mensajes y garantiza que las filas se actualicen con los valores de secuenciación correctos.

En la etapa 510, se realiza una determinación de si un mensaje está listo para enviarse. En caso afirmativo, el estado del mensaje se fija a N en la etapa 512. Esto significa que el mensaje no tiene ninguna dependencia y que puede ponerse en cola inmediatamente para su transmisión al sistema externo en la etapa 514. Si la respuesta en la etapa 510 es no, la dependencia del mensaje se determina en la etapa 516. Un mensaje está listo para enviarse

siempre que sea el primero de la secuencia (primer mensaje real o mensaje recibido fuera de la 'ventana activa'); o el mensaje previo se ha enviado satisfactoriamente. Se mantiene un historial de base de datos durante una ventana de tiempo específica con el fin de reducir la cantidad de espacio requerido. El valor de tiempo específico puede definirse por la distancia máxima entre dos mensajes que han "cambiado" de lugar entre sí. Por ejemplo: el mensaje 1 llega en  $t_1$ , el mensaje 2 llega en  $t_0$  con  $t_0 < t_1$ . El tiempo específico será el máximo teórico de  $(t_1 - t_0)$  en una 'situación razonable'. La situación razonable depende del sistema. En una situación normal, el mensaje 2 debe esperar la llegada del mensaje 1. Sin embargo, después de haber alcanzado el tiempo específico, el mensaje 2 puede declararse libre de su dependencia del mensaje 1. Si se identifica una dependencia, el estado se actualiza a D con el identificador de dependencia relevante. Esto se muestra en la etapa 518. El estado D indica que hay una dependencia entre el presente mensaje y otro. Por tanto, con el fin de enviar el presente mensaje, debe haberse recibido un acuse de recibo del mensaje del que depende el presente mensaje. Si en la etapa 516 no se identifica ninguna dependencia (es decir no se ha recibido un mensaje previo) el estado se fija a U en la etapa 520. El estado U significa desconocido. En este estado, no pueden enviarse mensajes puesto que no se ha calculado todavía ninguna dependencia. Si el estado del mensaje es U, el mensaje se inserta entonces en un temporizador en la etapa 522. Este temporizador retarda el mensaje dejando de ese modo tiempo para recibir entradas adicionales lo que puede permitir determinar una dependencia del mensaje en un análisis posterior tal como se describió anteriormente.

Con referencia a la figura 6, ahora se describirán las etapas asociadas con el cálculo de la secuencia de mensajes. En la etapa 600, se recuperan las filas relevantes con respecto al objetivo  $j$  y el orden de clave P de la tabla de base de datos persistente y se bloquean. En la etapa 602, se realiza una determinación de si se genera un mensaje de estado nulo. En caso afirmativo (604), el estado del mensaje se fija a nulo en la etapa 606. Si no se genera ningún mensaje nulo (608), se realiza una determinación de si hay o no mensajes previos en la etapa 610. Si no hay ningún mensaje previo (NO, 612) el estado del mensaje se fija a desconocido (estado "U"). En este punto, (etapa 616) el mensaje se inserta en el temporizador para esperar la llegada del mensaje previo. Si hay un mensaje previo en la etapa 610 (SÍ, 618) se realiza una determinación de si se ha enviado o no el mensaje previo en la etapa 620. Si no se ha enviado el mensaje previo (NO, 622) el estado del mensaje en cuestión se fija a dependiente (D) y se pone en cola cuando se acusa recibo de que se envió el mensaje previo, tal como se muestra en la etapa 624. Si se ha enviado el mensaje previo (SÍ, 626) el estado del mensaje se fija a no dependiente (N) y se pone en cola para su transmisión posterior en la etapa 628.

Haciendo referencia a la figura 7, ahora se describirá la secuenciación especial V o nulo. Un mensaje nulo es uno que no es necesario enviar pero que de hecho se usa en la base de datos para evitar huecos en los números de secuencia de los mensajes. Suponiendo que un mensaje, por ejemplo  $M_i(j)$  está en el estado nulo y se recibe  $P_{i+1}$ , tal como se muestra en la etapa 700, se realiza una determinación de si  $M_{i+1}(j)$  es nulo en la etapa 702. En caso afirmativo (704), el estado para el mensaje  $M_{i+1}(j)$  se fija a nulo, tal como se muestra en la etapa 706. En caso negativo (708), el proceso se mueve a la etapa 710. En la etapa 710, se realiza una determinación de si se ha enviado el mensaje  $M_{i-1}(j)$ . En caso afirmativo (712), el estado del mensaje  $M_{i+1}(j)$  se fija a N tal como se muestra en la etapa 714. Entonces, en la etapa 716, el mensaje  $M_{i+1}(j)$  se pone en cola para su envío. Si en la etapa 710 la determinación es no (718), el estado del mensaje  $M_{i+1}(j)$  se fija a D, basándose la dependencia en  $M_{i-1}(j)$ . Las diversas etapas descritas anteriormente pueden llevarse a cabo en un mensaje entrante o un mensaje retardado que ha estado en el temporizador. Una vez que se ha puesto en cola un mensaje para su transmisión, con la recepción del acuse de recibo, se produce una segunda parte del proceso que implica una rellamada en cascada y secuenciación. Esto se muestra en la figura 8 y una vez más las figuras en las casillas corresponden con las flechas en la figura 3. En la etapa 800, se ha enviado el mensaje  $M_i(j)$  al cliente y se ha recibido un acuse de recibo. Se realiza una determinación de si el mensaje  $M_i(j)$  tiene alguna dependencia en la etapa 802. En caso afirmativo (804), se realiza una hipótesis de que  $M_{i+n}(j)$  depende del mensaje  $M_i(j)$  en la etapa 806. Entonces, la dependencia del mensaje  $M_{i+n}(j)$  se actualiza al estado N y el estado del mensaje  $M_i(j)$  se actualiza al estado S. En la etapa 808, el mensaje  $M_{i+n}(j)$  se pone en cola para su transmisión al cliente. Entonces, en la etapa 810, el estado del mensaje  $M_i(j)$  se fija a S.

Debe indicarse que una vez que se pone en cola un mensaje que va a enviarse, no es necesario comprobar adicionalmente el estado del mensaje hasta que se produzca un acuse de recibo apropiado u otra acción. Después de que se ha enviado un mensaje, no siempre es necesario un acuse de recibo para que el proceso continúe. Este proceso (de enviar el mensaje y esperar la respuesta) puede ser o bien síncrono o bien asíncrono.

La presente invención se ilustra por medio de un único módulo de gestión de mensajes y una única base de datos, sin embargo, la invención podría implementarse en múltiples módulos de gestión de mensajes y bases de datos; aunque esto puede conducir a una complejidad mayor para el sistema.

La figura 9 muestra un ejemplo de parte de una tabla de dependencias que a su vez forma parte de la tabla persistente en la base de datos. La tabla presenta un código de identificación que aumenta numéricamente en la columna 900; un orden de clave P en la columna 902; la etiqueta ordenada (i) del mensaje en la columna 904; la etiqueta de objetivo o cliente (j) en la columna 906; estado de mensaje en la columna 908; dependencia de mensaje en la columna 910 y el propio mensaje en la columna 912. Para un objetivo particular, por ejemplo  $j$  es 1, los mensajes 9, 11, 13, 15 y 16 son relevantes. Sin embargo, el mensaje 16 es parte de una familia diferente basándose

en el orden de clave P y, por tanto, no tiene ninguna conexión con los otros mensajes 9, 11, 13 y 15. Haciendo referencia ahora a los mensajes de una familia común (AAAAAA), el mensaje con ID 9 es el primero en la familia puesto que tiene un índice i de 0. Este mensaje se ha enviado tal como se ilustra por el estado S. El mensaje con ID 11 es el siguiente en la familia pero este mensaje es nulo (estado V), por tanto, no hay ningún mensaje para enviar. El mensaje con ID 13 tiene el estado para ninguna dependencia (N) puesto que ya se ha enviado el mensaje con ID 9. Por tanto, este mensaje (ID 13) está listo para enviar. El mensaje con ID 15 tiene el estado de D puesto que depende del mensaje con ID 13. El mensaje 15 se pondrá en cola y se enviará una vez que se haya enviado el ID de mensaje 13.

La base de datos que incluye los mensajes e información asociada con los mismos está en forma de tabla persistente. Esta tabla tiene una ventana activa en cualquier momento particular en el tiempo. La ventana activa es aquella parte de la tabla que actualmente está almacenada y activa. Por ejemplo, con referencia a la figura 9, la ventana activa puede ser de ID9 a ID16. Ya no es necesario almacenar ningún mensaje en la tabla anterior a éstas (es decir de ID0 a ID8) y puede borrarse si es necesario. Se ha usado y procesado cualquier información que es importante según ha sido necesario. En la tabla, ambas familias de mensajes comienzan en la ventana activa. Éste no es necesariamente el caso. Por ejemplo, la ventana activa podría moverse de modo que comience en ID11, en este caso pueden borrarse ID9 e ID10. Los mensajes que son dependientes de ID9 e ID10 ya no pueden ver ID9 e ID10 en la ventana activa y de esa manera se supone que se han enviado todos los mensajes de los que depende un mensaje si no están en la ventana activa. Se apreciará que la ventana activa continuará cambiando a medida que entren y salgan mensajes. El tamaño de la ventana activa es predeterminado de modo que puede llegar cualquier mensaje que esté fuera de secuencia y sus dependencias se determinan de manera que cualquier mensaje no enviado se mantiene en la ventana activa hasta el momento en el que puedan enviarse de manera justificada. Puede ser un valor estático que es una hipótesis del tiempo máximo a esperar para un mensaje anterior dado. Por ejemplo después de n horas el mensaje Mi todavía está esperado el mensaje Mi-1, en este caso puede ser demasiado tarde para el mensaje Mi-1. Entonces, el mensaje Mi se procesará como 'primero de la secuencia'. En este caso, la ventana activa es de n horas. No es necesario mantener un historial de más de n horas. Es posible decidir si un mensaje entrante tiene que esperar a uno previo basándose en una fecha/hora del mensaje entrante Pi(j). La fecha/hora corresponde a la fecha/hora de creación del mensaje Pi-1(j).

El tamaño de la ventana activa puede variar de cualquier manera apropiada dependiendo de, por ejemplo, el número de mensajes, requisitos del sistema u otros requisitos.

En determinadas situaciones, un mensaje puede incluir información relacionada con más de una entidad. Por ejemplo, con respecto a un registro de nombre de pasajero (PNR) puede haber dos pasajeros registrados originalmente en un vuelo específico. Si un pasajero decide cambiar su vuelo, el mensaje original debe dividirse en dos mensajes separados, que entonces podrían dar como resultado posteriormente dos familias diferentes de mensajes aunque estén enlazados. La figura 10 muestra un ejemplo de una situación de este tipo. Un registro de nombre de pasajero original se identifica como 1000. En un determinado punto en el tiempo, el segundo pasajero Jean Dupond 1002 decide tomar un vuelo diferente. En este punto, el mensaje se divide en dos familias separadas 1004 y 1006, cada una con órdenes de clave P diferentes, concretamente AAAAAA y BBBBBB, respectivamente. El registro de nombre de pasajero original se denomina a menudo padre anterior, mientras que los 2 mensajes generados recientemente se denominan padre e hijo posteriores.

La figura 11 muestra las necesidades de secuenciación, detalles de registro y dependencias de registro en el denominado caso de división, en el que un padre anterior se divide en padre e hijo.

En el lado izquierdo 1100, el mensaje A0 se divide en el mensaje A1 y B0. Tanto A1 como B0 tienen mensajes posteriores en su familia de los que se muestra uno para cada familia como A2 y B1, respectivamente. El lado derecho 1102 de la figura 11 muestra las dependencias de los mensajes diferentes. B1 depende de B0, que a su vez depende de A0 mientras que A2 depende de A1 y A1 de B0.

Los ejemplos descritos anteriormente se basan en dependencias de mensajes sencillas. Sin embargo, a menudo las dependencias entre familias diferentes de mensajes dan como resultado una relación compleja de la dependencia y, a su vez, la secuencia requerida de esos mensajes. La figura 12 es un árbol de dependencias para un único objetivo con familias diferentes de mensajes. La figura muestra secuencias diferentes de mensajes con claves diferentes, en otras palabras de familias diferentes. Las claves diferentes se indican por M, M', M" y M"" y se define un orden específico para los mensajes en cada familia mediante el número de referencia asociado. Se ha eliminado la anotación de objetivo o cliente para simplificar el diagrama. Tal como se ilustra, M', M" y M"" contienen una referencia a su padre, el mensaje M y, por tanto, existe una dependencia y las familias diferentes de mensajes lo saben. En la figura, una flecha indica que un mensaje depende de otro. Por ejemplo, M'2 depende de M"0.

Una característica de la presente invención es almacenar un registro histórico de mensajes entrantes en una base de datos durante un periodo de tiempo predeterminado. Este periodo de tiempo predeterminado depende del intervalo máximo de tiempo necesario para reordenar los mensajes que están desordenados. Si se determina que un mensaje es independiente de cualquier otro mensaje, puede ignorarse el periodo de tiempo predeterminado.

Esta invención se ha descrito con referencia a la compra de billetes en el entorno de viajes. Sin embargo, se apreciará que la invención puede aplicarse a otros entornos, por ejemplo cualquier entorno de mensajería de múltiples procesamientos, por ejemplo la banca.

**REIVINDICACIONES**

1. Método para ordenar una pluralidad de mensajes recibidos de un remitente que van a enviarse a un receptor en una secuencia basándose en la dependencia de un mensaje de uno o más otros mensajes, comprendiendo el método las etapas de:
- recibir (500) uno o más mensajes de un flujo de mensajes;
  - generar (502) uno o más mensajes a partir del flujo de mensajes y determinar un orden, un objetivo y un valor para un campo de estado para cada uno de los mensajes generados;
  - almacenar los mensajes generados en una base de datos;
  - identificar una característica de cada mensaje almacenado, característica que es común a un grupo de los mensajes almacenados;
  - identificar una dependencia de mensaje para los mensajes almacenados en el grupo de los mensajes almacenados de un parámetro del mensaje almacenado;
  - revisar (510) un mensaje almacenado particular en la base de datos para determinar si el mensaje almacenado puede enviarse:
    - determinando (516) si el mensaje almacenado particular depende de un mensaje almacenado previo y determinando un estado del mensaje almacenado previo leyendo un valor del campo de estado del mensaje almacenado previo; y
    - actualizando (512, 518, 520) el valor del campo de estado del mensaje almacenado particular basándose en el estado del mensaje almacenado previo;
  - enviar el mensaje almacenado particular después del acuse de recibo de que se ha enviado el mensaje almacenado previo.
2. Método según la reivindicación 1, que comprende además retardar el mensaje almacenado particular hasta que se cumpla una condición predeterminada.
3. Método según la reivindicación 1, que comprende además retardar (522) el mensaje almacenado particular hasta que se cumpla un retardo de tiempo.
4. Método según cualquier reivindicación anterior, en el que la etapa de determinar (516) si existe un mensaje previo comprende recibir un acuse de recibo de mensaje de que se ha recibido el mensaje previo por un destinatario.
5. Método según la reivindicación 4, que comprende además enviar el mensaje almacenado particular a una cola de salida.
6. Método según la reivindicación 4 o la reivindicación 5, que comprende además actualizar el valor del campo de estado del mensaje almacenado particular para de ese modo actualizar el valor del campo de estado de cualquier mensaje posterior asociado con el mensaje almacenado particular.
7. Método según cualquier reivindicación anterior, que comprende además repetir secuencialmente las etapas de revisión, actualización y envío para cada mensaje en el grupo de mensajes.
8. Método según cualquier reivindicación anterior, que comprende además actualizar la base de datos con cualquier cambio en el estado.
9. Sistema de mensajes para ordenar una pluralidad de mensajes sin un identificador de secuencia recibidos de un remitente que van a enviarse a un receptor en una secuencia equivalente a un orden de mensajes de la pluralidad de mensajes, comprendiendo el sistema:
- un módulo receptor para recibir (500) uno o más mensajes de un flujo de mensajes y para generar (502) uno o más mensajes a partir del flujo de mensajes y determinar un orden, un objetivo y un valor de un campo de estado para cada uno de los mensajes generados, y almacenar los mensajes generados en una base de datos;
  - un módulo de gestión de mensajes para identificar una característica de cada mensaje que es común a un grupo de los mensajes almacenados; para identificar una secuencia que corresponde al orden de

mensajes para el grupo de mensajes a partir de un parámetro del mensaje; y para revisar (510) un mensaje almacenado particular en la base de datos para determinar si el mensaje almacenado particular puede enviarse empleando:

- 5
- medios para determinar (516) si existe un mensaje almacenado previo en la secuencia antes del mensaje almacenado particular y determinar un estado del mensaje previo leyendo un valor del campo de estado del mensaje almacenado previo; y
- 10
- medios para actualizar (512, 518, 520) el valor del campo de estado del mensaje almacenado particular basándose en el estado del mensaje almacenado previo;
  - un módulo de transmisión para enviar el mensaje almacenado particular según la secuencia una vez que se ha enviado el mensaje almacenado previo.
- 15
10. Sistema según la reivindicación 9, que comprende además un módulo de retardo para retardar el mensaje almacenado particular hasta que se cumpla una condición predeterminada.
- 20
11. Sistema según la reivindicación 9, que comprende además un módulo de retardo de tiempo (308) para retardar (522) el mensaje almacenado particular hasta que se cumpla un retardo de tiempo.
12. Programa informático que comprende instrucciones para llevar a cabo el método según cualquiera de las reivindicaciones 1 a 8 cuando dicho programa informático se ejecuta en un aparato programable.

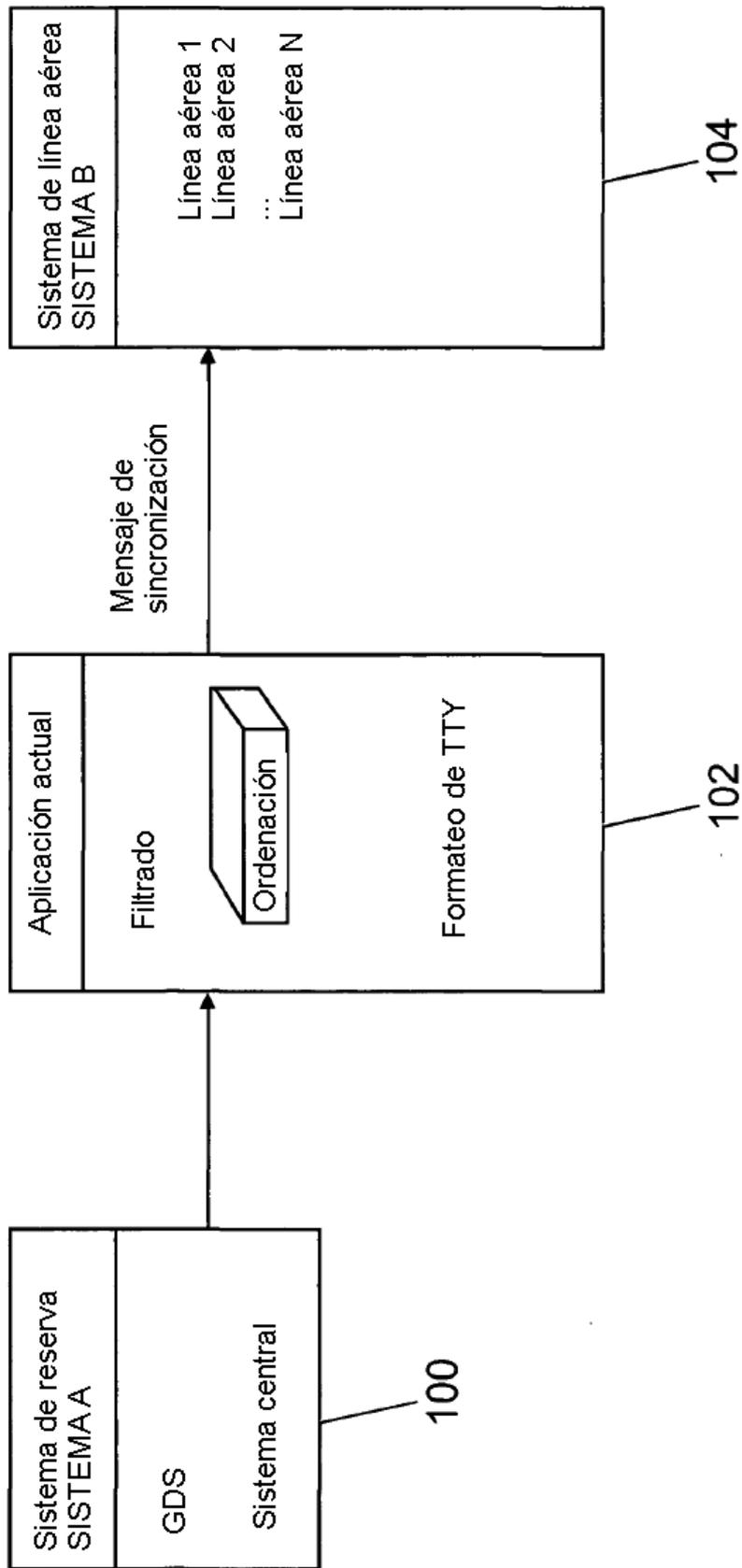


Fig. 1

### Necesidad de secuenciación - Caso normal -

Es necesario tener sobres en el orden correcto para un registro de pasajeros dado

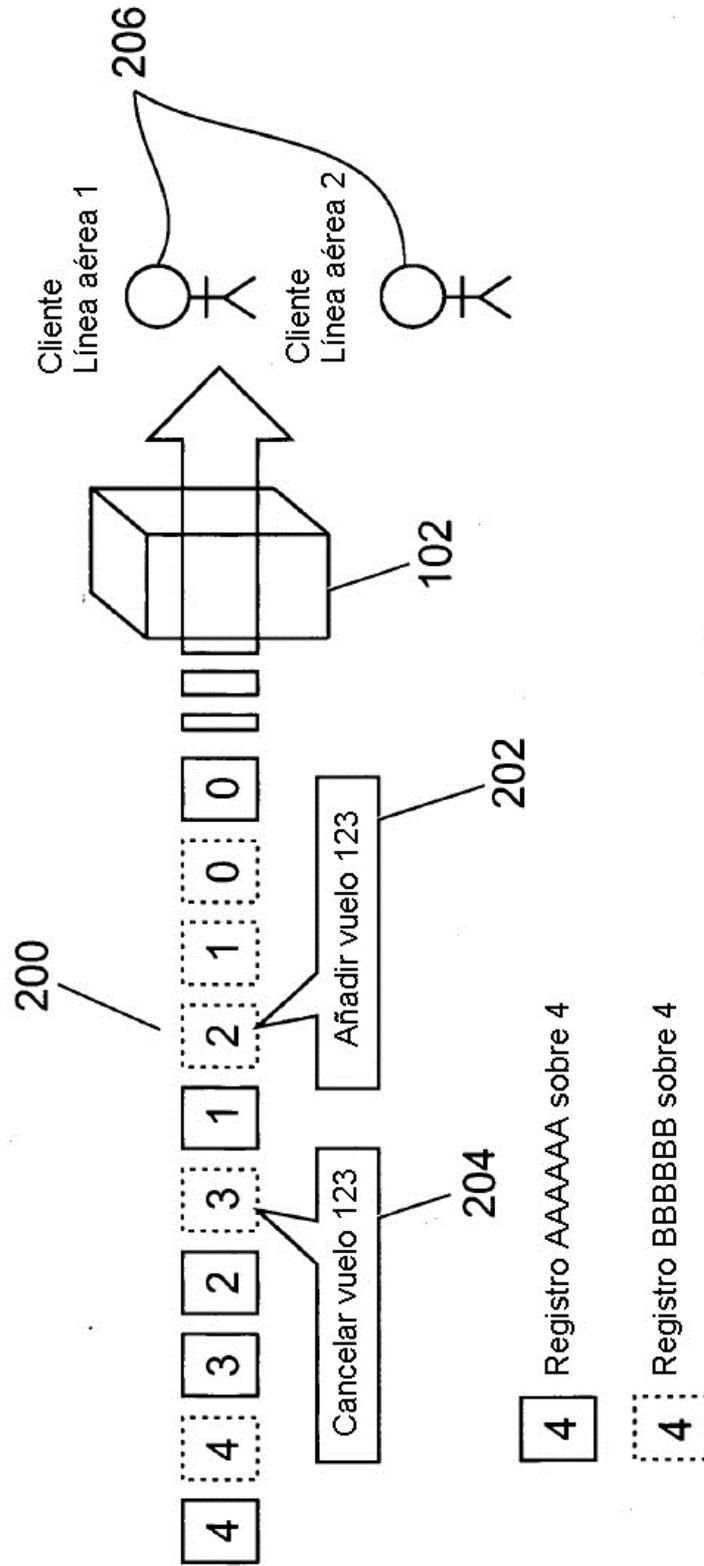


Fig. 2

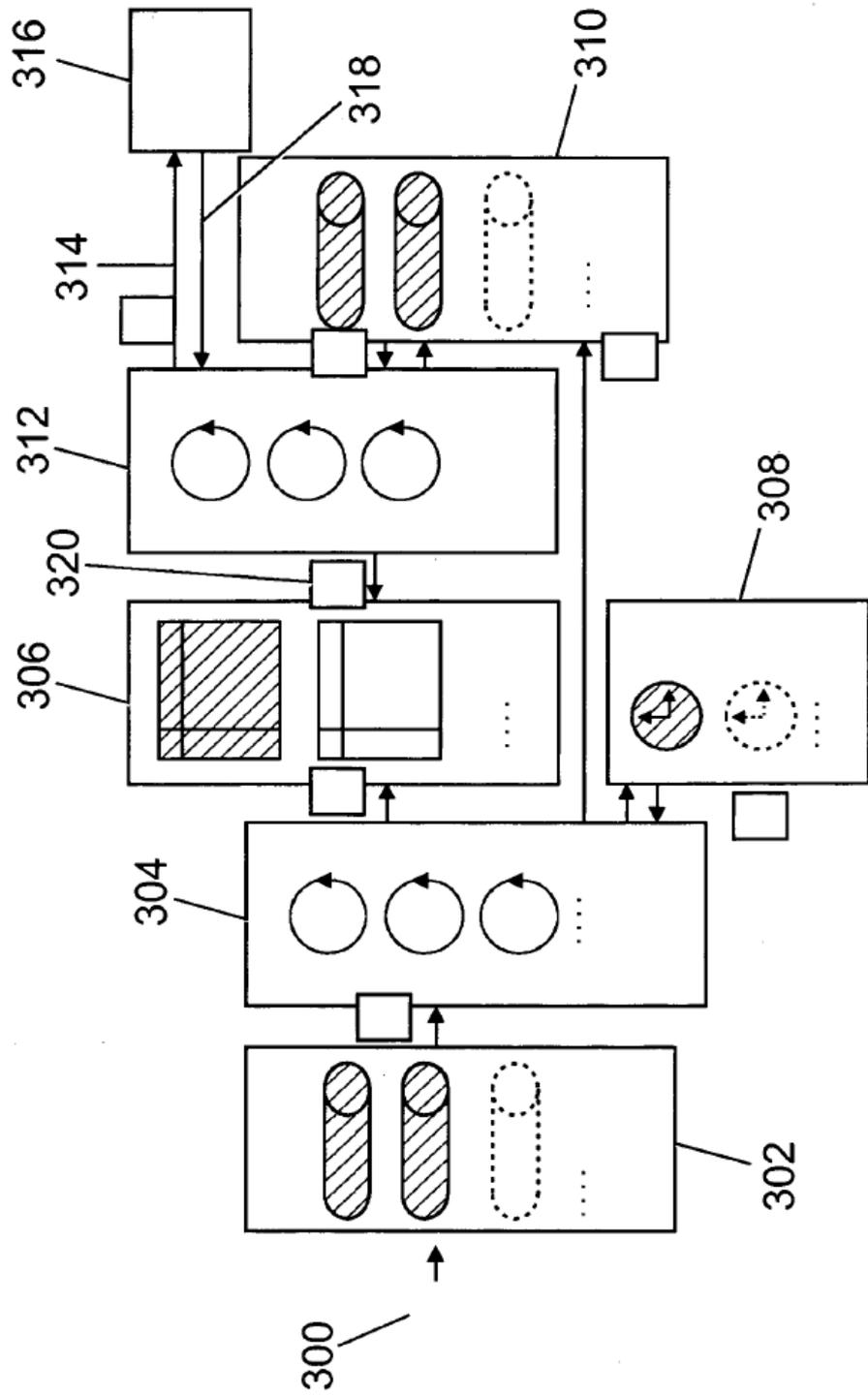


Fig. 3

Número de línea	Referencia de mensaje	Dependencia
N	M9(j)	
N + 1	M10(j)	Línea N

*Fig. 4*

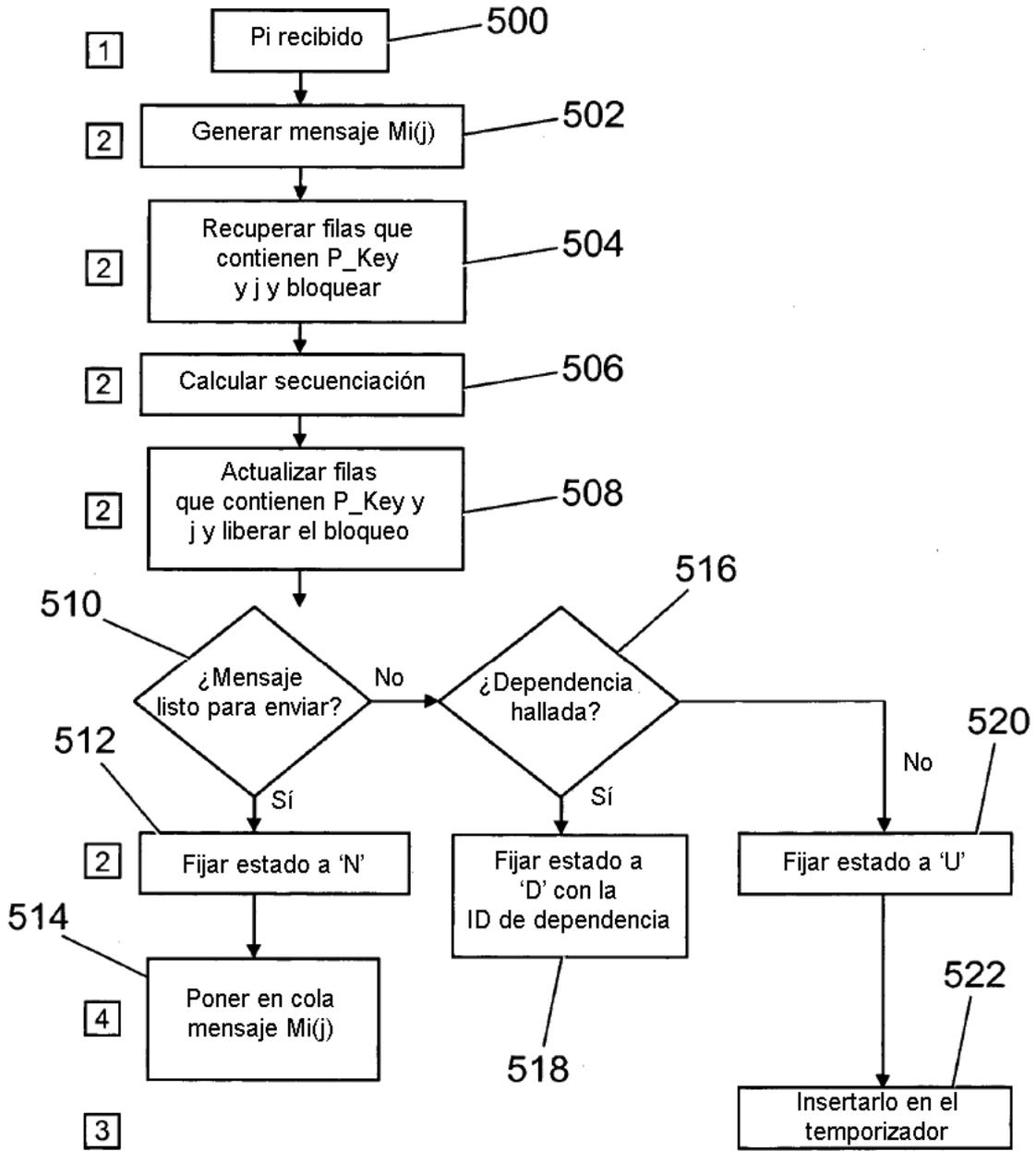


Fig. 5

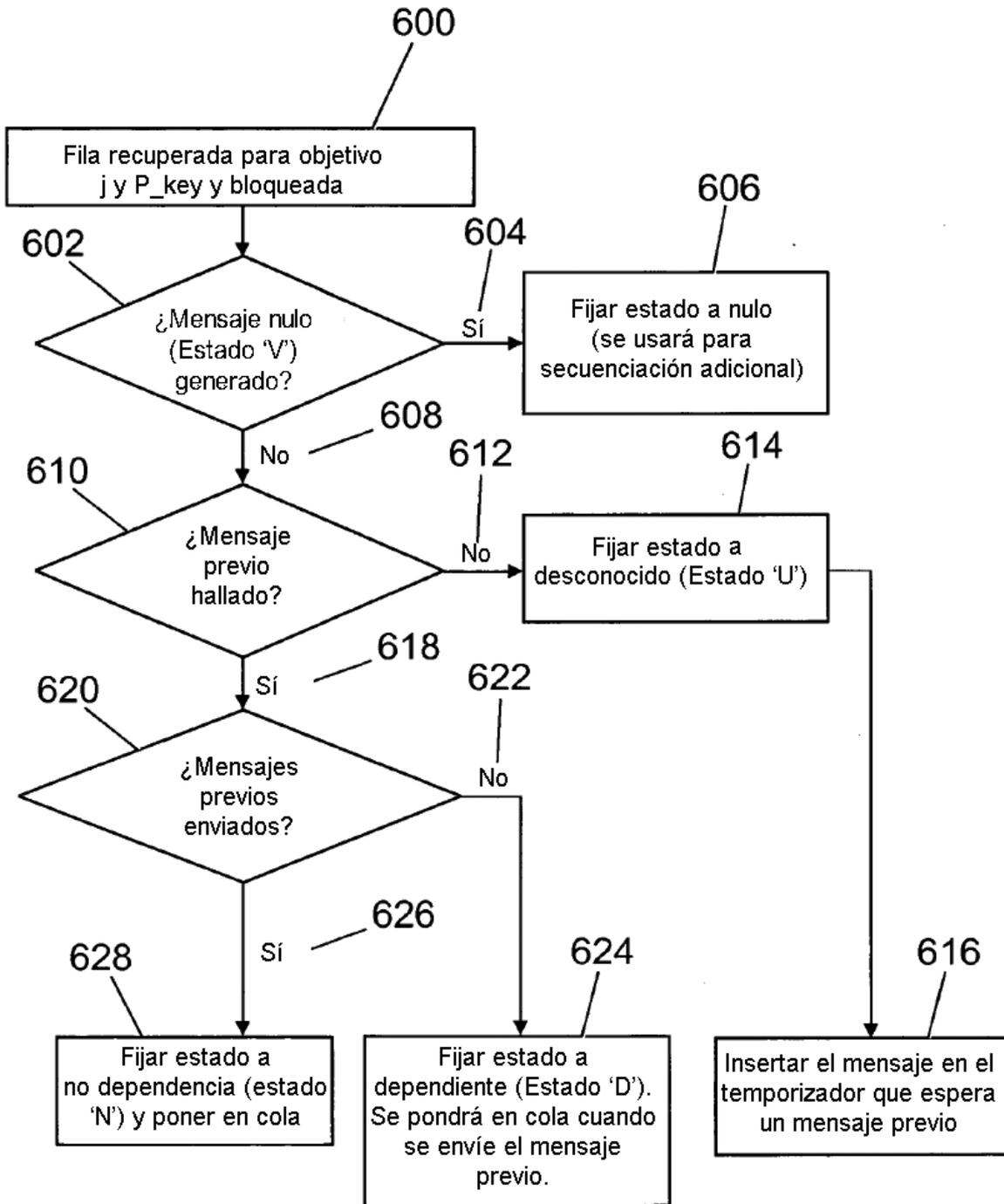


Fig. 6

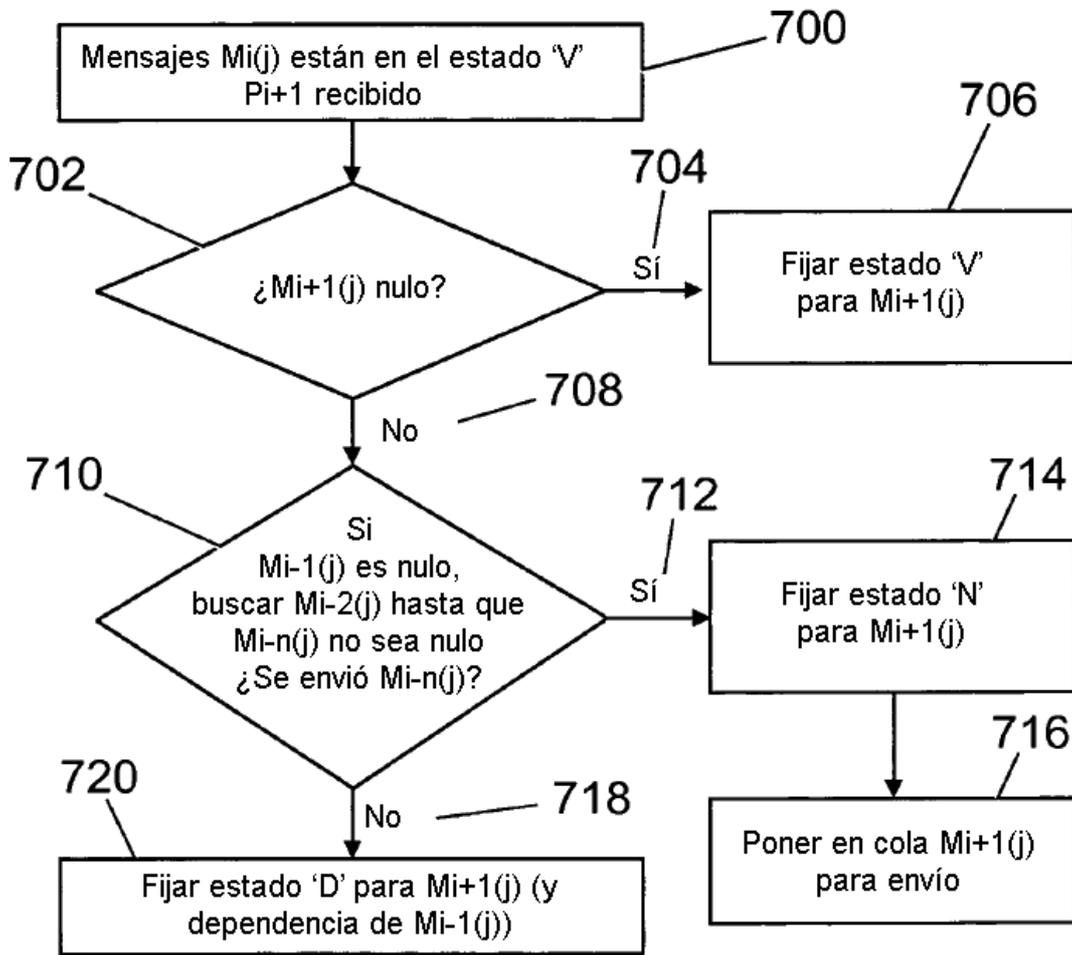


Fig. 7

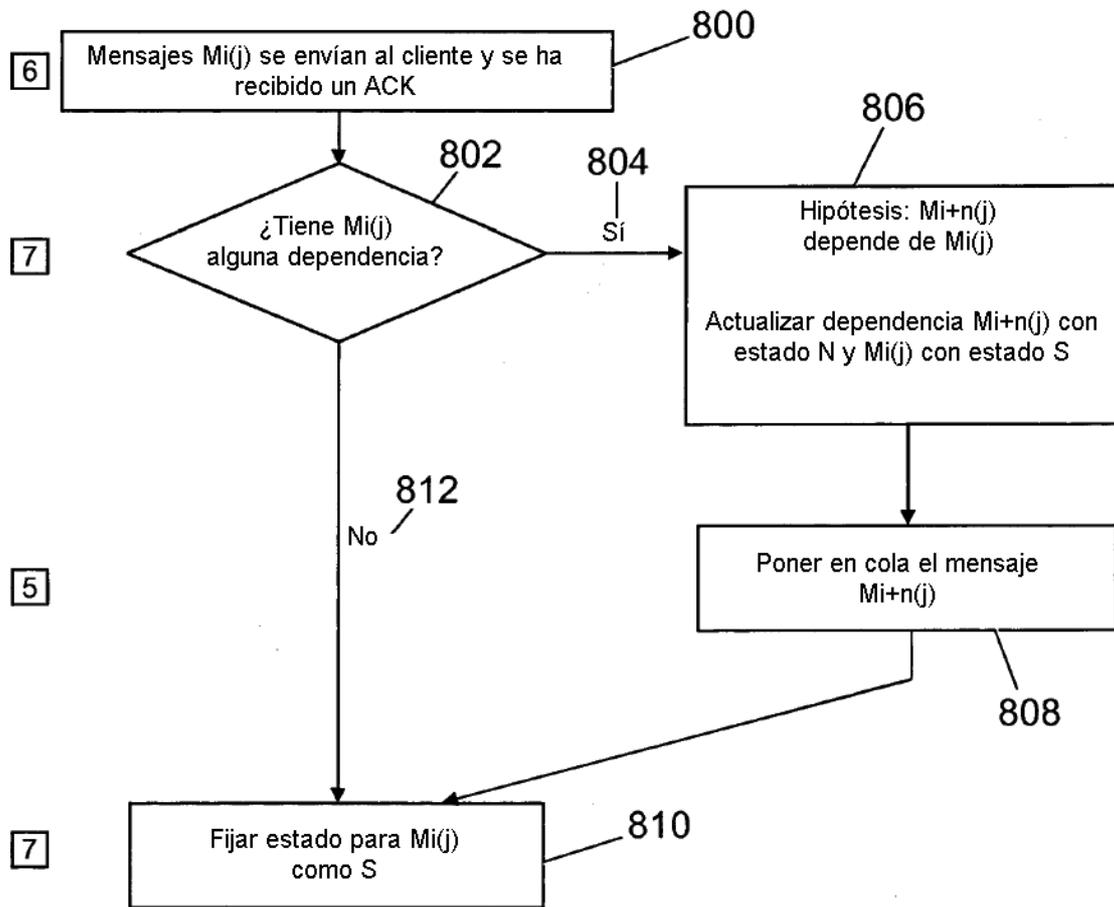


Fig. 8

ID	P_Key	i	j	Estado	Dep_Id	Mensaje
9	AAAAAA	0	1	S	∅	M0(1)
10	AAAAAA	0	2	S	∅	M0(2)
11	AAAAAA	1	1	V	∅	∅
12	AAAAAA	1	2	N	10	M1(2)
13	AAAAAA	2	1	N	9	M2(1)
14	AAAAAA	2	2	D	12	M2(2)
15	AAAAAA	3	1	D	13	M3(1)
16	BBBBBB	2	1	U	∅	M'2(1)

*Fig. 9*

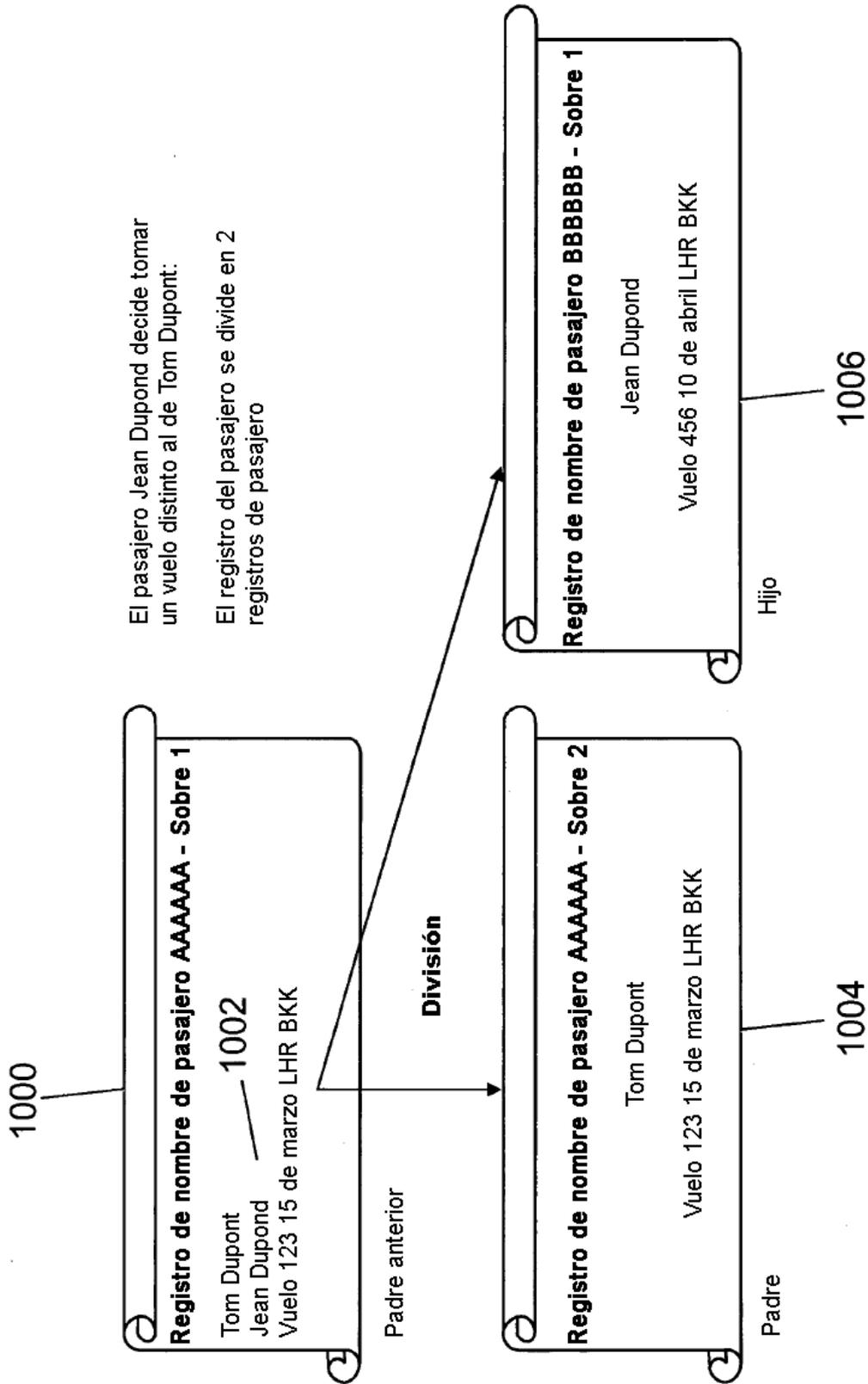


Fig. 10

**Necesidad de secuenciación - Caso de división -**  
Debe preceder un mensaje DVD (división del hijo) al mensaje TTY para su padre

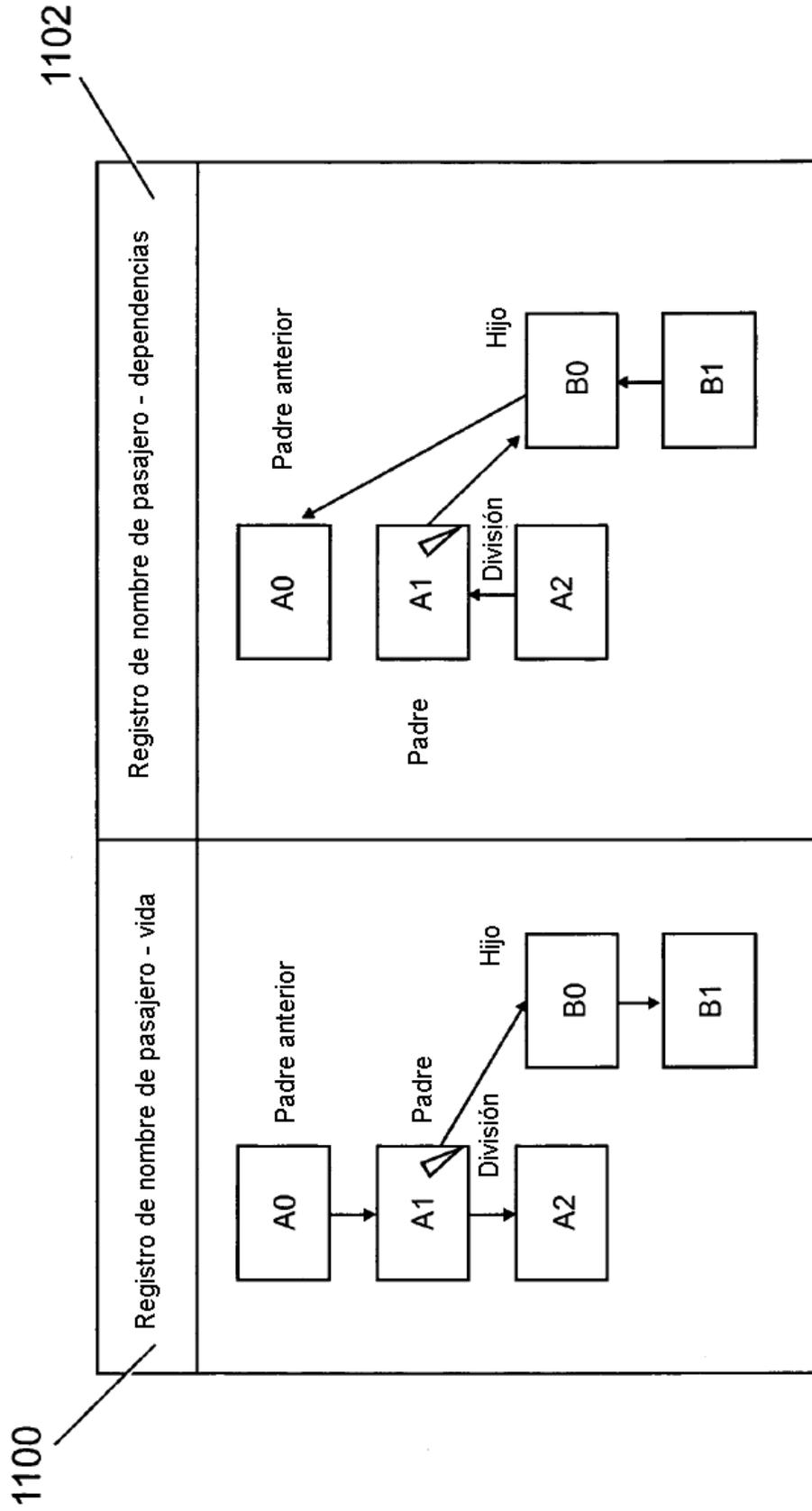
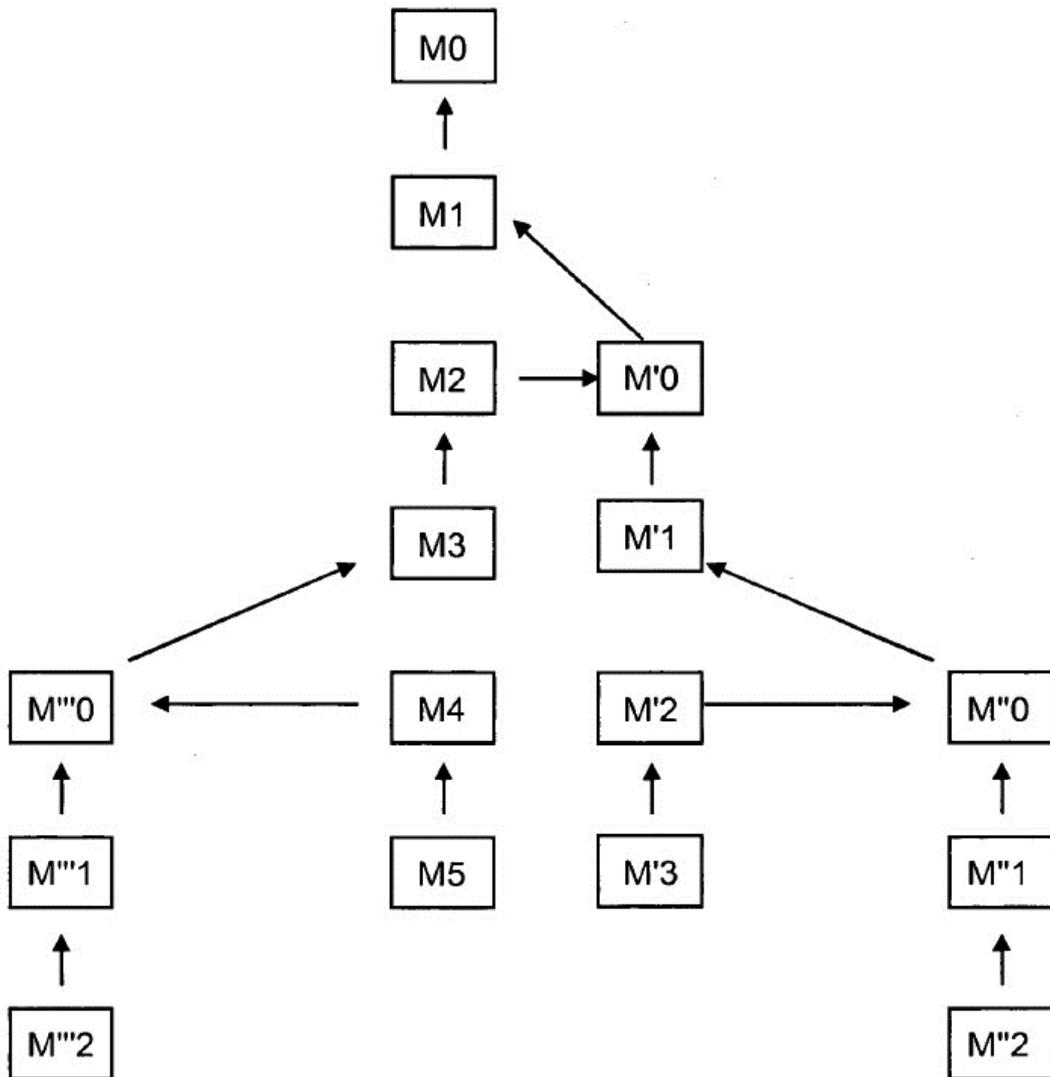


Fig. 11



Leyenda:  
**M'0** : Mensaje de familia M', índice 0.  
 En este caso, se omite el objetivo o cliente para simplificar la anotación.

*Fig. 12*