

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 805 755**

51 Int. Cl.:

G06F 16/23 (2009.01)

H04L 29/08 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **10.03.2011** **E 11305256 (7)**

97 Fecha y número de publicación de la concesión europea: **29.04.2020** **EP 2501106**

54 Título: **Sistema y método para sincronización de sesión con sistemas externos independientes**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
15.02.2021

73 Titular/es:

AMADEUS S.A.S. (100.0%)
485 Route du Pin Montard, Sophia Antipolis
06410 Biot, FR

72 Inventor/es:

BRIET, CÉCILE;
MIKAELIAN, JÉRÔME;
PIERLOT, LOÏCK y
ALBEROLA, BERTRAND

74 Agente/Representante:

SUGRAÑES MOLINÉ, Pedro

ES 2 805 755 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

Sistema y método para sincronización de sesión con sistemas externos independientes

5 Campo de la invención

La presente invención se refiere en general a sistemas de procesamiento de datos y en particular a arquitecturas de software de cliente-servidor o servidor web usados para acceso a base de datos. Aún más específicamente, la presente invención se refiere a un método y un sistema para optimizar sincronización de datos desde sistemas externos independientes y posiblemente heterogéneos, en una transacción operada en un sistema de referencia, mientras se minimiza el riesgo de pérdida de datos.

Un sistema externo independiente debe entenderse en este punto como un sistema externo controlado independientemente del sistema de referencia. Un agente o usuario puede controlar el sistema de referencia, o puede ser automático.

Un sistema externo heterogéneo debe entenderse en este punto como un sistema externo con procesos, aplicaciones y estructuras de datos definidos en el mismo que pueden diferir de los del sistema de referencia.

Un ejemplo de aplicación se refiere a la reserva de viajes y emisión de billetes en la industria de las aerolíneas. El sistema externo puede ser un mecanismo automático, por ejemplo, un robot operado por un sistema de distribución global (GDS) tal como Amadeus, una empresa con sede en Madrid, España. El mecanismo automático puede disponerse, por ejemplo, para actualizar registros, tales como registros de nombres de pasajeros (PNR), directamente en el almacenamiento. Las actualizaciones pueden consistir en cancelar automáticamente reservas no emitidas en billetes.

Según otro ejemplo, un agente de viajes accede al sistema externo y permite la reserva en un sistema independiente del sistema de referencia. El sistema externo puede permitir, por ejemplo, la reserva de eventos de entretenimiento mientras que el sistema de referencia permite la reserva de billetes de viaje.

Antecedentes de la invención

La sincronización de datos puede definirse como la tarea de mantener datos consistentes entre diversos componentes de un sistema que trabajan en datos de referencia dados y duplicados de tales datos. Para ese propósito, se asigna normalmente un número de versión a cada elemento de datos. Un elemento de datos puede definirse mediante una estructura de datos y un identificador único.

El problema de sincronización de datos en un sistema multiusuario en un entorno de datos compartidos es un problema general bien conocido que se ha abordado en diversos contextos que exige soluciones específicas según las arquitecturas de sistemas, aplicaciones soportadas y cualquier aspecto específico y requisitos de tales de sistemas.

La mayoría de métodos de sincronización encontrados en la técnica relacionada se basan en un componente central que actúa como un coordinador entre diversos usuarios, aplicaciones, procesos o sistemas que necesitan trabajar en datos sincronizados. Este es, de hecho, un concepto muy genérico que sugiere tener un gestor único a cargo de la coordinación requerida.

Basándose en esta solución, el documento US 7.539.778 divulga una arquitectura de cliente-servidor para sincronización de datos entre diferentes clientes, desplegando un servidor de sincronización central entre clientes y un almacenamiento de datos de extremo final. Se divulga un método de sincronización particular, basándose en una memoria caché para almacenar en memoria intermedia de forma permanente actualizaciones entrantes en un almacenamiento permanente asignando un identificador de memoria caché único a cada una de las mismas. Los conflictos de escritura entre el servidor de sincronización que escribe nuevas entradas en la memoria caché y actualizaciones replicadas desde el extremo final a la memoria caché se resuelven usando un mecanismo de bloqueo que se basa en los identificadores de memoria caché. Esta solución de sincronización se basa, por lo tanto, en un componente de memoria caché que realiza dos tareas: resolver conflictos de escritura en actualizaciones y minimizar la conexión y el transporte al almacenamiento de datos de extremo final.

Una característica del documento US 7.539.778 relevante a sincronización de datos es el tratamiento de conflictos de escritura, usando identificadores para entradas de memoria caché. Sin embargo, este método no tiene como objetivo proporcionar sincronización de datos en un sistema basándose en un almacenamiento de datos y otros sistemas externos, independientes y posiblemente heterogéneos.

Para gestionar el problema de acceso concurrente a un almacenamiento de datos debido a un proceso local y procesos adicionales de sistemas externos independientes y posiblemente heterogéneos, debe abordarse un número de cuestiones específicas.

Por ejemplo, en un sistema de procesamiento de transacciones, tal como una plataforma de comercio electrónico, podrían existir una amplia gama de productos y múltiples proveedores de tales productos. Datos relativos a productos y sus proveedores se mantienen normalmente en una instalación de almacenamiento proporcionada en el sistema de procesamiento. Sin embargo, algunos de estos productos podrían estar disponibles en un paquete que contiene productos adicionales externalizados de sistemas externos.

Además, varios usuarios y varias aplicaciones de transacciones pueden acceder a todo el sistema al mismo tiempo. Esto es especialmente cierto en sistemas con acceso concurrente masivo intrínseco tales como plataformas de comercio electrónico de mercado masivo. Datos relativos a una sesión de transacción abierta por un usuario dado podría volverse obsoleta con respecto a disponibilidad, precio, etc. debido a actualización de gestión de producto y debido a transacciones operadas por otros usuarios o por otras aplicaciones.

Podrían existir razones adicionales de obsolescencia de datos tales como eventos no esperados que afectan a la disponibilidad de productos y precio, modificaciones de políticas de negocio decididas por proveedores, etc.

El problema resultante al que se enfrenta un sistema de este tipo se resume en la Figura 1A. El diagrama de secuencia de la Figura 1A ilustra un conflicto de versiones entre un expediente de usuario en almacenamiento y sesión de usuario a guardar. Algunos datos en el expediente en almacenamiento se han modificado por otra aplicación, posiblemente desde un sistema externo, mientras el usuario A estaba trabajando en datos recuperados de una versión previa de dichos expedientes en almacenamiento. Supóngase que el usuario A decide guardar su trabajo. Los datos a guardar se basan en una versión obsoleta de dicho expediente en almacenamiento. Es probable que tales datos sean inválidos.

En sistemas de procesamiento de transacciones clásicos, se pueden definir tres tipos de métodos de tratamiento de versiones para hacer frente al problema de conflicto de versiones.

Una primera opción es rechazar conflicto de versiones. Este es un método simple y robusto que consiste en rechazar cualquier orden de guardar si la versión de expediente de usuario a guardar por la sesión de usuario no coincide con la versión actual de expediente de usuario en almacenamiento. Esta política asegura que nunca hay ningún problema de acceso concurrente ya que la única diferencia entre el expediente en almacenamiento y el que tiene que almacenarse es la actualización que resulta de la sesión de usuario. Aunque este método es muy simple de implementar y robusto, su limitación obvia es que los datos a guardar de la sesión de usuario o de cualquier sistema externo pueden rechazarse y el trabajo del usuario podría perderse. En caso de conflicto, la única solución permitida es ignorar el trabajo actual, recuperar la nueva versión de almacenamiento, rehacer el trabajo e intentar de nuevo guardar, esperando que esta vez el expediente en almacenamiento no se haya actualizado mientras tanto. Este enfoque es muy agresivo e inefectivo en modelos de negocio con muchos accesos concurrentes.

Una segunda opción es tratar el conflicto de versiones en un enfoque de guardar todo. En sistemas actuales que usan sesiones y sincronización con otros sistemas, existe únicamente una fuente de datos sincronizada. Añadir datos desde un sistema externo implica fusionar todos los datos en la sesión y, a continuación, decidir qué hacer; esto es viable si, y únicamente si, la sesión de usuario puede ser consciente, a través de medios de señalización, de que algunos datos se hicieron disponibles mediante un sistema externo. La primera elección posible es guardar esta sesión fusionada directamente en el almacenamiento. No existe ningún mecanismo perfecto desde una perspectiva de usuario ya que pueden existir algunos datos erróneos o temporales en la sesión de usuario, debido a trabajo de usuario en progreso, y estos datos se guardarán en el expediente en almacenamiento fuera del control del usuario. Permite que la sesión consiga una vista actualizada del almacenamiento.

Como alternativa, una tercera opción es tratar el conflicto de versiones en un enfoque de guardar nada. Esto significa que los datos de un sistema externo se ponen en sesión sin haberse guardado, pendiente de la decisión del usuario para guardar. Esto aumenta el riesgo de pérdida de datos. Si el usuario finalmente omite guardar, datos de un sistema externo pueden no recuperarse nunca de nuevo y se perderían. Para simplificar este problema, sistemas reales normalmente piden al usuario que guarde los datos de sesión antes de sincronización con sistemas externos, pero esto es una fuerte restricción para el usuario. Un segundo problema es que nuevos datos en el expediente de usuario almacenado pueden no volverse visibles al usuario.

Los inconvenientes y beneficios relativos de los tres métodos de tratamiento de conflicto de versiones se resumen en la Figura 1B.

El documento US7.539.778 B2 describe sincronización de datos entre diferentes clientes usando un servidor de sincronización central enlazado a un almacenamiento de datos de extremo final que proporciona adicionalmente una memoria caché para almacenar en memoria intermedia de forma permanente actualizaciones entrantes en un almacenamiento permanente asignando un identificador de memoria caché único. El documento US 2007/0118572 A1 concierne a la detección de cambios concurrentes en datos. Si una primera versión no es la misma que una segunda versión, un proceso de resolución de concurrencia resolverá las inconsistencias en datos.

65 Objeto de la invención

Por lo tanto, es un objeto amplio de la invención divulgar un método y un sistema para mejorar la sincronización de datos en una arquitectura de procesamiento de datos que incluye un almacenamiento de datos, sistemas de cliente-servidor o servidor web para acceder a dicho almacén de datos, y una pluralidad de sistemas independientes y posiblemente heterogéneos externos a dicho almacén de datos.

5 Más específicamente, la invención tiene por objeto resolver un problema técnico importante relacionado con la recuperación y agregación de datos de fuentes externas independientes y posiblemente heterogéneas en un almacenamiento de datos.

10 Es un objeto más específico adicional de la invención que esto debe ajustar arquitecturas de procesamiento de datos tales como los divulgados para sistemas de reserva de aerolíneas, y caracterizado por un alto nivel de transacciones desde el lado de cliente y actualizaciones muy frecuentes, por ejemplo, de la tarifa y bases de datos de disponibilidad proporcionados por aerolíneas y de otros productos y servicios relacionados con la industria del viaje y turismo.

15 Objetos, características y ventajas adicionales de la presente invención se harán evidentes para los expertos en la materia tras el examen de la siguiente descripción en referencia con los dibujos adjuntos.

Sumario de la invención

20 La invención se define mediante las reivindicaciones independientes. A continuación se divulgan un método y un sistema para mejorar sincronización de datos en una arquitectura de procesamiento de datos que incluye un almacenamiento de datos, sistemas de cliente-servidor o servidor web para acceder a dicho almacén de datos, y una pluralidad de sistemas independientes y posiblemente heterogéneos externos a dicho almacén de datos.

25 Esto trae una solución a un problema técnico principal relacionado con la recuperación y agregación de datos de fuentes externas independientes y posiblemente heterogéneas en un almacenamiento de datos.

30 Adicionalmente, esto proporciona un mecanismo de fusión de versiones que mejora métodos de gestión de conflicto de versiones encontrados en la técnica anterior. El método trata la actualización concurrente de datos de usuario dados en dicho almacén de datos que se producen cuando dichos datos se modifican simultáneamente por el usuario y por otros procesos que operan en dichos sistemas externos. Para ese propósito, se propone definir y aplicar conjuntos de reglas para determinar a qué fuente de información se da prioridad en caso de que dos canales quieran actualizar los mismos datos en un expediente al mismo tiempo.

35 Esto resulta en la eliminación de riesgo de pérdida de datos provocada por conflicto de versiones entre datos recuperados y modificados por una sesión de usuario y una versión actualizada de los datos almacenados que no coinciden con la versión recuperada. Más específicamente, una ventaja proporcionada por la invención es que datos relacionados con usuario, incluyendo datos de sistemas externos, se guardan sin riesgo de pérdida, incluso si el usuario ha iniciado una sesión de usuario y aún está trabajando en los mismos.

40 Una ventaja adicional obtenida es una forma de actualización de sesión de usuario con datos de los sistemas externos que no fuerzan que se guarden datos de usuario, evitando, por lo tanto, que se guarden datos aún no validados.

45 Más específicamente, la invención divulga un método de actualización de una sesión de usuario con datos externos recibidos desde sistemas externos, comprendiendo el método las siguientes etapas realizadas con al menos un procesador de datos:

50 - recibir, en un sincronizador en comunicación con la sesión de usuario y un almacenamiento de datos, una solicitud desde la sesión de usuario para recuperar un expediente de usuario almacenado del almacenamiento de datos, en donde un expediente es un conjunto de datos;

55 - recuperar, en el sincronizador, el expediente de usuario almacenado del almacenamiento de datos, crear, en el sincronizador, a partir del expediente de usuario almacenado un expediente intermedio que comprende al menos el contenido del expediente de usuario almacenado y distribuir el expediente intermedio a la sesión de usuario, como un expediente de sesión, para operaciones por un usuario;

- almacenar temporalmente, en una cola de eventos en comunicación con el sincronizador, datos externos recibidos y/o actualizar, en el almacenamiento de datos, el expediente de usuario almacenado con datos externos recibidos;

60 - recibir, en el sincronizador, una actualización de usuario realizada en el expediente de sesión, actualizar el expediente de sesión con la actualización de usuario recibida y devolver el expediente de sesión actualizado a la sesión de usuario;

- tras recibir, en el sincronizador, una solicitud de actualización enviada por la sesión de usuario:

65 - recuperar, en el sincronizador, el expediente de usuario almacenado actualizado,

- recuperar, en el sincronizador, todos los eventos en la cola de eventos,

- fusionar, en el sincronizador, el expediente de usuario almacenado actualizado recuperado con actualizaciones desde todos los eventos recuperados de la cola de eventos y guardar el expediente fusionado en el almacenamiento de datos, y

- fusionar, en el sincronizador, el expediente fusionado con el expediente de sesión actualizado y devolver el resultado a la sesión de usuario, como el expediente de sesión; y

- tras recibir, en la sesión de usuario, una instrucción para cerrar la sesión, hacer una pregunta al usuario a través de interfaz de usuario de la sesión de usuario acerca de guardar o no guardar el expediente de sesión:

- tras recibir una instrucción desde el usuario para guardar el expediente de sesión, actualizar el expediente de sesión a través de la fusión de datos externos recibidos, el expediente almacenado y el expediente de sesión, y guardar el mismo en el almacenamiento de datos; y

- tras no recibir una instrucción para guardar el expediente de sesión o tras recibir una instrucción para borrar el expediente de sesión, no guardar el expediente de sesión y no actualizar el expediente almacenado en el almacenamiento de datos.

Por lo tanto, la sesión recibe un expediente intermedio actualizado que comprende los datos enviados por el sistema externo y el expediente almacenado de última versión. Por lo tanto, el usuario es además capaz de trabajar en una versión actualizada del expediente intermedio. En esta etapa, el expediente almacenado no ha sido modificado por el usuario. Mientras todos los datos recibidos por sistemas externos se guardan en el expediente almacenado, todos los cambios realizados por la sesión pueden, por lo tanto, guardarse además en el almacenamiento de datos o borrarse sin modificar el expediente almacenado en el almacenamiento de datos. Por lo tanto, la invención facilita significativamente el trabajo hecho por el usuario evitando la pérdida de datos de sistemas externos sin forzar al usuario a guardar actualizaciones de su sesión.

Ventajosamente, el sistema externo puede ser heterogéneo con el almacenamiento de datos. Esto es posible gracias a la función de fusión realizada por el sincronizador.

Según la invención, el almacenamiento de datos recibe dichos datos externos desde el sistema externo. Tras la recepción de dichos datos externos por el almacenamiento de datos, el almacenamiento de datos realiza la etapa de modificar el expediente almacenado de modo que comprende dichos datos externos y la etapa de guardar el expediente almacenado en el almacenamiento de datos.

La etapa en la que el sincronizador actualiza el expediente intermedio, de modo que el expediente intermedio comprende dichos datos externos, se produce después de que las etapas de modificar el expediente almacenado, de modo que comprende dichos datos externos, y de guardar el expediente almacenado en el almacenamiento de datos y comprende las siguientes etapas: el sincronizador recupera el expediente almacenado del almacenamiento de datos y actualiza el expediente intermedio con el expediente almacenado.

Ventajosamente, las etapas de modificar el expediente almacenado, de modo que comprende dichos datos externos, y de guardar el expediente almacenado en el almacenamiento de datos se producen tras la recepción de los datos externos por el almacenamiento de datos. Después de recibir en el sincronizador una solicitud de actualización enviada por la sesión y antes de la etapa en donde el sincronizador envía el expediente intermedio a la sesión para sustituir el expediente de sesión, a continuación el sincronizador actualiza el expediente intermedio con el expediente almacenado.

Según la invención, dichos datos externos del sistema externo se reciben en una cola de eventos en comunicación con el sincronizador. La etapa en la que el sincronizador actualiza el expediente intermedio, de modo que el expediente intermedio comprende dichos datos externos, comprende las siguientes etapas: el sincronizador recupera dichos datos externos de la cola de eventos, recupera el expediente almacenado del almacenamiento de datos y, a continuación, actualiza el expediente intermedio a través de fusionar dichos datos externos y el expediente almacenado.

El sincronizador puede recuperar primero dichos datos externos de la cola de eventos y a continuación recuperar el expediente almacenado del almacenamiento de datos. Como alternativa, el sincronizador primero recupera el expediente almacenado del almacenamiento de datos y a continuación recupera dichos datos externos de la cola de eventos.

Para realizar las etapas de modificar el expediente almacenado, de modo que comprende dichos datos externos, y de guardar el expediente almacenado en el almacenamiento de datos, el sincronizador guarda el expediente intermedio en el almacenamiento de datos.

El sincronizador a continuación inicia la etapa en donde el almacenamiento de datos guarda el expediente intermedio.

Tras la recepción de dichos datos externos desde el sistema externo, la cola de eventos envía a dicho sistema externo una confirmación de que dichos datos se reciben correctamente.

5 Después de la etapa de recibir en el sincronizador una solicitud de actualización enviada por la sesión y antes de la etapa en donde el sincronizador envía el expediente intermedio a la sesión para sustituir el expediente de sesión, el método realiza las etapas en donde el sincronizador actualiza el expediente intermedio a través de fusionar dichos datos externos y el expediente almacenado.

10 La etapa de recibir datos externos desde el sistema externo comprende recibir datos externos por el almacenamiento de datos y recibir datos externos adicionales por una cola de eventos en comunicación con el sincronizador. Tras la recepción de dichos datos externos por el almacenamiento de datos, a continuación el almacenamiento de datos realiza la etapa de modificar el expediente almacenado de modo que comprende dichos datos externos y la etapa de guardar el expediente almacenado en el almacenamiento de datos. El sincronizador recupera dichos datos externos adicionales de la cola de eventos, recupera el expediente almacenado del almacenamiento de datos, actualiza el expediente intermedio a través de fusionar dichos datos externos adicionales y el expediente almacenado y sustituye el expediente almacenado con el expediente intermedio.

20 Por lo tanto, en esta etapa, el expediente almacenado y el expediente intermedio comprenden los datos externos recibidos en el almacenamiento de datos y los datos externos recibidos adicionales en la cola de eventos.

A continuación el sincronizador puede actualizar el expediente intermedio a través de fusionar dichas actualizaciones de usuario y el expediente intermedio para proporcionar al usuario con un expediente intermedio que comprende los datos externos recibidos en el almacenamiento de datos, los datos externos recibidos adicionales en la cola de eventos y las actualizaciones de usuario.

30 Tras la actualización del expediente intermedio por el sincronizador a través de fusionar dichos datos externos adicionales y el expediente almacenado, el sincronizador guarda en el almacenamiento de datos el expediente intermedio. Por lo tanto, en esta etapa, el expediente almacenado comprende los datos externos recibidos en el almacenamiento de datos y los datos externos recibidos adicionales en la cola de eventos.

35 El sincronizador recibe de la sesión actualizaciones de usuario adicionales, a continuación el sincronizador recupera el expediente almacenado del almacenamiento de datos, actualiza el expediente intermedio a través de fusionar el expediente almacenado con las actualizaciones de usuario adicionales de la sesión y envía el expediente intermedio a la sesión para sustituir el expediente de sesión.

40 Tras recibir una instrucción desde el usuario para guardar el expediente de sesión, la sesión proporciona al usuario una posibilidad de guardar el expediente intermedio en el almacenamiento de datos. Por lo tanto, la sesión proporciona al usuario con una posibilidad de actualizar el expediente intermedio a través de la fusión de datos externos recibidos desde el sistema externo, el expediente almacenado y el expediente de sesión y, a continuación, guardar el expediente intermedio como actualizado en el almacenamiento de datos. Por lo tanto, el trabajo posiblemente realizado por el usuario a través de la sesión y todos los datos posiblemente recibidos desde el al menos un sistema externo se guardan en el almacenamiento de datos. Por lo tanto, la siguiente vez que el sincronizador recupere el expediente almacenado, esa versión recuperada comprenderá actualizaciones de la sesión y datos del al menos un sistema externo.

50 Según la invención, la posibilidad es una pregunta hecha al usuario a través de una interfaz de usuario de la sesión acerca de guardar o no guardar el expediente de sesión, haciéndose dicha pregunta cuando la sesión recibe una instrucción para cerrar la sesión.

55 La sesión recibe una instrucción de guardar el expediente de sesión y envía al sincronizador una solicitud de guardar, en donde tras la recepción de la solicitud de guardar el sincronizador actualiza el expediente intermedio a través de la fusión de datos recibidos desde el sistema externo, el expediente almacenado y el expediente de sesión, y guarda el expediente intermedio como actualizado en el almacenamiento de datos.

Como alternativa, si la sesión no recibe una instrucción de guardar el expediente de sesión o si la sesión recibe una instrucción para borrar el expediente de sesión, a continuación las actualizaciones de usuario no se guardan y la sesión no actualiza el expediente almacenado en el almacenamiento de datos.

60 El método se realiza como resultado de ejecución de un software informático por al menos un procesador de datos, almacenándose el software informático en un medio de memoria legible por ordenador no transitorio.

65 Otra materia objeto de la invención es un sincronizador para actualizar una sesión de usuario con datos externos recibidos en un almacenamiento de datos de sistemas externos configurados para realizar el método descrito anteriormente.

Breve descripción de los dibujos

La Figura 1A muestra un ejemplo de acceso concurrente en almacenamiento de datos.

5 La Figura 1B describe inconvenientes y beneficios relativos de métodos existentes de tratamiento de conflicto de versiones.

La Figura 2 muestra un ejemplo de arquitectura de software de alto nivel de sistema a la que se aplica la invención ventajosamente.

10 La Figura 3 muestra un ejemplo de arquitectura de sistema detallada a la que se aplica la invención ventajosamente.

La Figura 4 muestra un ejemplo de arquitectura de sistema detallada a la que se aplica la invención ventajosamente, con flujos de comunicación para una secuencia de tareas típicas a realizar.

15 La Figura 5 proporciona un ejemplo de diagrama secuencial de las diversas etapas típicas que resultan del procesamiento de solicitudes de usuario final y acciones desde sistemas externos.

La Figura 6 proporciona una vista general del proceso de fusión.

20 La Figura 7 explica cómo se aplica un conjunto de reglas en intersección de datos entre datos externos y datos de sesión, y para dos expedientes en general.

25 La Figura 8 muestra un ejemplo de arquitectura de sistema detallada a la que se aplica la invención ventajosamente, con flujos de comunicación para una secuencia de tareas típicas a realizar en el caso de tratamiento de eventos asíncrono mejorado por una tarea planificada.

30 La Figura 9 proporciona un ejemplo de diagrama secuencial de las diversas etapas típicas que resultan del procesamiento de solicitudes de usuario final y acciones desde sistemas externos en el caso de tratamiento de eventos asíncrono mejorado por una tarea planificada.

35 La Figura 10 muestra un ejemplo de arquitectura de sistema detallada a la que se aplica la invención ventajosamente, con flujos de comunicación para una secuencia de tareas típicas a realizar en el caso de únicamente una actualización de almacenamiento.

La Figura 11 proporciona un ejemplo de diagrama secuencial de las diversas etapas típicas que resultan del procesamiento de solicitudes de usuario final y acciones desde sistemas externos en el caso de únicamente una actualización de almacenamiento.

40 La Figura 12 muestra un ejemplo de arquitectura de sistema detallada a la que se aplica la invención ventajosamente, con flujos de comunicación para una secuencia de tareas típicas a realizar en el caso de una actualización asíncrona por un sistema externo y sin sesión de usuario.

45 La Figura 13 proporciona un ejemplo de diagrama secuencial de las diversas etapas típicas que resultan del procesamiento de acciones desde sistemas externos en el caso de una actualización asíncrona por un sistema externo y sin sesión de usuario.

Descripción detallada

50 La invención se define mediante las reivindicaciones independientes.

La Figura 2 muestra un ejemplo de arquitectura de sistema de alto nivel de la presente invención, en la que un componente del sincronizador 4 se ubica entre un almacenamiento de datos 1 y una sesión de usuario 3 y recibe datos desde al menos un sistema externo 5, 5'. La función de este sincronizador 4 es agregar y fusionar datos. Para cualquier expediente recuperado del almacenamiento de datos 1 para una sesión de usuario dada 3, y para recibir actualizaciones desde al menos un sistema externo 5, 5' y desde la sesión de usuario 3, el sincronizador 4 crea o actualiza un expediente intermedio. El expediente intermedio puede actualizarse fusionando el expediente intermedio actual con el expediente de sesión y con actualizaciones de expediente recibidas desde sistemas externos, si tales actualizaciones están disponibles. El sincronizador 4 actualiza el expediente intermedio según a un conjunto de reglas predefinidas o avisando al usuario siempre que sea necesario.

La Figura 3 muestra un ejemplo de arquitectura de sistema detallada a la que se aplica la invención ventajosamente. Esta arquitectura se usará para ilustrar diferentes escenarios de interés y, para ese propósito, se detallarán los flujos de comunicación para cada uno de tales escenarios. Por consiguiente, todos los elementos mostrados en **la Figura 3** se describen en posteriores figuras relacionados con escenarios de interés.

La Figura 4 muestra un ejemplo de arquitectura de sistema detallada a la que se aplica la invención ventajosamente. También se muestran una secuencia de tareas típicas a realizar por el sistema y flujos de comunicación relacionados. Esta secuencia corresponde al siguiente escenario:

- 5 En la etapa 1, un usuario 2 está trabajando en una sesión de usuario 3 y, en este sentido, se recupera un expediente de usuario del almacenamiento de datos 1 y se visualiza. Para esa etapa, el sincronizador 4 recupera en primer lugar el expediente almacenado del almacenamiento de datos 1, como un expediente intermedio. El expediente intermedio se envía a continuación a la sesión de usuario 3 como un expediente de sesión. Esto se hace a través de un manejador de expedientes 12 y un manejador de almacenamiento 13. El manejador de almacenamiento está a cargo de recuperar expedientes almacenados del almacenamiento de datos 1 y guardar expedientes actualizados en el almacenamiento de datos 1. El manejador de expedientes está a cargo de recibir solicitudes y actualizaciones desde la sesión de usuario 3, enviar el expediente intermedio a la sesión de usuario 3 y gestionar todas las tareas internas del sincronizador 4 relacionadas con la recuperación y el guardado del expediente intermedio, y con operaciones de actualización-fusión.
- 10
- 15 En la etapa 2, para hacer una transacción que opera en un sistema externo 5', tal como reservar un paquete, el usuario cambia a ese sistema externo 5', por ejemplo abriendo una pestaña proporcionada en la interfaz de usuario, y realiza la transacción.
- 20 En la etapa 3, el sistema externo 5' envía los datos que corresponden a la transacción y la referencia al expediente actual a un punto de entrada externo 6. El punto de entrada externo puede ser cualquier canal adecuado que conecta el sistema externo 5' y el sincronizador 4. Puede proporcionarse por un servicio web o cualquier medio específico. Los datos enviados por el sistema externo 5' se almacenan temporalmente en una cola de eventos 9 a través de un manejador de notificaciones 7 y un manejador de eventos 8. La cola de eventos 9 es una instalación que mantiene datos enviados por sistemas externos. El manejador de notificaciones es una instalación a cargo de recibir notificaciones desde sistemas externos. El manejador de eventos es una instalación a cargo del tratamiento de eventos, que incluyen tales tareas como: poner eventos en la cola, recuperar eventos, encontrar eventos a recuperar para un expediente dado.
- 25
- 30 En la etapa 4, el usuario desencadena una acción, tal como cambiar la pestaña de expediente en la interfaz de usuario, guardar expediente, actualizar expediente..., que actualiza el expediente de sesión. Para ese propósito, el expediente intermedio se actualiza y devuelve a la sesión de usuario 3. Esto se hace en las etapas 5 a 9.
- 35 En la etapa 5, el sincronizador 4 comprueba la cola de eventos 9 para cualquier evento a procesarse para el expediente actual. Se recuperan todos los eventos en cola.
- En la etapa 6, el sincronizador 4 recupera el último expediente guardado en el almacenamiento de datos 1.
- 40 En la etapa 7, el sincronizador 4 fusiona el último expediente con las actualizaciones desde todos los eventos anteriormente recuperados. Esto se hace por un fusionador 11 con reglas suministradas por un motor de reglas 10. La función realizada por el fusionador se explicará en la descripción de **La Figura 6**. La función realizada por el motor de reglas se explicará en la descripción de **La Figura 7**.
- 45 En la etapa 8, el sincronizador 4 guarda el expediente fusionado en el almacenamiento de datos 1 para evitar pérdida de datos.
- En la etapa 9, el sincronizador 4 fusiona el expediente fusionado con el expediente de sesión y devuelve el resultado al usuario. Este es el expediente visualizado al usuario.
- 50 Después de la etapa 9, el usuario puede continuar trabajando y, tras la finalización del proceso según para lo que sea la sesión, el usuario puede guardar el expediente para validación final de la transacción; o incluso antes de finalizar el proceso, el usuario puede guardar el expediente y trabajar de nuevo en el mismo más tarde. Por lo tanto, la sesión continuaría rehaciendo las etapas anteriores desde la etapa 2 o desde la etapa 4, hasta que el expediente de sesión se guarda finalmente, o hasta que el usuario decide cerrar la sesión sin guardar ningún cambio.
- 55 Por lo tanto, la invención proporciona un método de agregación de datos particularmente adecuado para un sistema orientado a transacciones que comprende un sistema principal y otros sistemas externos, independientes y posiblemente heterogéneos.
- 60 Ese método se basa en una arquitectura de sistema y una función de fusión que mejora los métodos ya conocidos.
- La función de fusión depende de un mecanismo que elimina el riesgo de pérdida de datos provocado por conflicto de versiones entre datos recuperados y modificados en la sesión de usuario 3 y una versión actualizada de los datos almacenados que no coincide con la versión recuperada.
- 65 Más específicamente, datos relacionados con usuario, incluyendo datos de sistemas externos, se guardan sin riesgo

de pérdida incluso si el usuario ha iniciado una sesión de usuario 3 y aún está trabajando en los mismos.

Una ventaja adicional del método es una forma de actualización de sesión de usuario 3 con datos desde las fuentes externas que no fuerza que se guarden los datos de usuario, evitando, por lo tanto, que se guarden datos aún no validados.

La Figura 5 proporciona un ejemplo de diagrama secuencial que muestra las diversas etapas que resultan del procesamiento de solicitudes de usuario final y notificaciones desde los sistemas externos 5, 5' en un escenario típico. Obsérvese que, en este diagrama, la notificación recibida desde el sistema externo 5' podría desencadenarse por el usuario, por ejemplo, en el que el usuario cambia a ese sistema externo y realiza una acción que provoca que ese sistema externo envíe datos al expediente intermedio.

En las etapas 1, 2, 2', 3, 3', 4, un usuario está trabajando en una sesión de usuario 3 y, en este sentido, se recupera un expediente de usuario del almacenamiento de datos 1 y se visualiza. Para eso, el sincronizador 4 recupera en primer lugar el expediente almacenado del almacenamiento de datos 1, como un expediente intermedio. El expediente intermedio se envía a continuación a la sesión de usuario 3 como un expediente de sesión.

En las etapas 5, 6, para hacer una transacción que opera en un sistema externo 5', tal como reservar un paquete, el usuario cambia a ese sistema externo 5', por ejemplo abriendo una pestaña proporcionada en la interfaz de usuario, realiza la transacción y vuelve a la sesión abierta.

Como resultado, se enviará un evento a la cola. Este es el hecho importante en este punto.

Realmente, la actualización desde el sistema externo 5' puede desencadenarse por el usuario cambiando a ese sistema externo 5' o mediante un proceso interno independiente del sistema externo. Por lo tanto, las etapas 5 y 6 se refieren a un ejemplo particular y no restringen el alcance de la invención. Las etapas 5 y 6 son entonces únicamente opcionales y se ilustran con líneas discontinuas. Esto también se aplica a la Figura 9.

En las etapas 7, 8, el sistema externo 5' envía una notificación de cambios relacionados con datos en el expediente de usuario. En el sincronizador 4, los datos enviados por el sistema externo 5' se almacenan temporalmente en la cola de eventos 9 a través del manejador de notificaciones y el manejador de eventos.

En las etapas 9, 10, un sistema externo 5 actualiza directamente el expediente de usuario en el almacenamiento de datos 1.

En las etapas 11, 12, 13, 14, la sesión de usuario 3 se actualiza debido al usuario trabajando.

En las etapas 15, 16, el usuario desencadena una actualización de la sesión de usuario 3.

En las etapas 17, 18, para comenzar la actualización de la sesión de usuario 3, el sincronizador 4 recupera el último expediente guardado en el almacenamiento de datos 1.

En las etapas 19, 20, el sincronizador 4 comprueba la cola de eventos 9 para cualquier evento a procesarse para el expediente actual. Se recuperan todos los eventos en cola.

En la etapa 21, el sincronizador 4 fusiona el último expediente con las actualizaciones desde todos los eventos anteriormente recuperados.

En las etapas 22, 23, el sincronizador 4 guarda el expediente fusionado en el almacenamiento de datos 1 para evitar pérdida de datos. En este punto, las actualizaciones hechas por los sistemas externos 5, 5' se han guardado y asegurado.

En la etapa 24, el sincronizador 4 fusiona el expediente fusionado con el expediente de sesión.

En las etapas 25, 26, el sincronizador 4 devuelve el resultado al usuario. Este es el expediente visualizado al usuario. Por lo tanto, el usuario ahora trabaja en una sesión actualizada sincronizada con el expediente almacenado actualizado.

Por lo tanto, la invención proporciona una solución al problema técnico relacionado con la recuperación y agregación de datos desde fuentes externas independientes y posiblemente heterogéneas en un almacenamiento de datos 1.

La invención también permite gestionar el riesgo de pérdida de datos provocado por conflicto de versiones entre datos almacenados recuperados por una sesión de usuario 3 y una versión actualizada de los datos almacenados que no coinciden con la versión recuperada.

El método según la invención también habilita la recepción y fusión de datos desde fuentes externas sin forzar que se

guarden los datos temporales de usuario.

5 **La Figura 6** proporciona una vista general del proceso de fusión. El mecanismo de fusión de versiones de la invención tiene por objetivo específicamente evitar un riesgo de pérdida de datos provocado por conflicto de versiones entre datos recuperados y modificados en la sesión de usuario 3 y una versión actualizada de los datos almacenados que no coinciden con la versión recuperada. El principio del método de la invención es fusionar inmediatamente y guardar datos recibidos desde los sistemas externos 5, 5' y fusionar y guardar datos desde la sesión de usuario 3 únicamente tras una solicitud por la sesión de usuario 3. El proceso de fusión requiere identificar intersecciones de datos y aplicar reglas de prioridad a intersecciones de datos. Este es un proceso de dos etapas, que se efectúa como se indica a continuación.

15 La primera etapa del proceso de fusión es identificar las zonas de intersección entre datos que se cambian por el almacenamiento de datos 1 y los sistemas externos 5, 5'. Esto se hace preferentemente en el momento de configuración de sistema, es decir, cada vez que se integra un nuevo sistema externo en la arquitectura. Se definen los datos a integrar y a su vez se definen intersecciones. A continuación se definen reglas de prioridad y configuran para uso de tiempo de ejecución.

20 La segunda fase es la etapa de tiempo de ejecución del proceso de fusión. En tiempo de ejecución el sistema comprueba si existen datos en la zona de intersección de los datos intercambiados. Si existen, las reglas para cada atributo en la zona de intersección se aplican a continuación de la configuración.

25 **La Figura 7** explica cómo se aplica un conjunto de reglas en intersección de datos entre datos externos y datos de sesión, y para dos expedientes en general. Esto es cómo se aplican las reglas para una intersección de datos dada en tiempo de ejecución.

30 Se han identificado tres subconjuntos en la intersección entre datos en sesión y datos externos. Se ha definido un conjunto de reglas, que proporciona una regla para cada subconjunto. Dos reglas asignan automáticamente una fuente preferida al primer y segundo subconjuntos objetivo. No existe ninguna regla automática para el tercer subconjunto. Este subconjunto se gestiona avisando al usuario.

35 **La Figura 8** muestra un ejemplo de arquitectura de sistema detallada a la que se aplica la invención ventajosamente, con flujos de comunicación para una secuencia de tareas típicas a realizar en el caso de tratamiento de eventos asíncrono mejorado por una tarea planificada 14.

Las etapas 1 a 3 son idénticas a las etapas 1 a 3 de la Figura 4.

40 En la etapa 4, una tarea planificada 14, independiente de cualquier sesión, solicita al sincronizador 4 que compruebe la cola de eventos 9 para cualquier evento a procesarse para el expediente actual. Se recuperan todos los eventos en cola, si hay alguno.

Las etapas 5 a 7 son idénticas a las etapas 6 a 8 de la Figura 4.

La etapa 8 es idéntica a la etapa 4 de la Figura 4.

45 La etapa 9 es idéntica a la etapa 6 de la Figura 4.

La etapa 10 es idéntica a la etapa 9 de la Figura 4.

50 Por lo tanto, una ventaja adicional de la invención es la posibilidad de que los sistemas externos 5, 5' notifiquen y envíen actualizaciones al sistema principal a través de comunicación asíncrona.

55 **La Figura 9** proporciona un ejemplo de diagrama secuencial de las diversas etapas típicas que resultan del procesamiento de solicitudes de usuario final y acciones desde sistemas externos 5, 5' en el caso de tratamiento de eventos asíncrono mejorado por una tarea planificada 14. Este diagrama secuencial se refiere al escenario y flujos de comunicación ilustrados por la Figura 8.

Las etapas 1 a 14 son idénticas a las etapas 1 a 14 de la Figura 5.

60 En la etapa 15, una tarea planificada 14, independiente de cualquier sesión, solicita al sincronizador 4 que compruebe la cola de eventos 9 para cualquier evento a procesarse para el expediente actual.

En las etapas 16, 17, se recuperan todos los eventos en cola, si hay alguno.

65 Las etapas 18, 19 son idénticas a las etapas 17, 18 de la Figura 5.

Las etapas 20, 21, 22 son idénticas a las etapas 21, 22, 23 de la Figura 5.

En la etapa 23, el sincronizador 4 ha completado la tarea planificada 14 y la tarea planificada 14 se cierra.

5 Las etapas 24 a 29 son idénticas a las etapas 15 a 20 de la Figura 5. Después de la etapa 29, ya que no había más datos en cola, el sincronizador 4 únicamente tiene que fusionar los datos de sesión con el expediente almacenado recientemente recuperado.

Las etapas 30 a 32 son idénticas a las etapas 24 a 26 de la Figura 5.

10 **La Figura 10** muestra un ejemplo de arquitectura de sistema detallada a la que se aplica la invención ventajosamente, con flujos de comunicación para una secuencia de tareas típicas a realizar en el caso de únicamente una actualización de almacenamiento.

15 La etapa 1 es idéntica a la etapa 1 de la Figura 4.

En la etapa 2, independientemente del expediente intermedio, el expediente almacenado se actualiza directamente en el almacenamiento de datos 1.

20 La etapa 3 es idéntica a la etapa 4 de la Figura 4.

La etapa 4 es idéntica a la etapa 6 de la Figura 4.

La etapa 5 es idéntica a la etapa 9 de la Figura 4.

25 Por lo tanto, una ventaja adicional de la invención es la posibilidad de que los sistemas externos 5 envíen directamente actualizaciones al sistema principal, es decir, sin notificación del mismo al sincronizador 4 y sin forzar que la sesión de usuario 3 se guarde. La actualización se volverá visible cuando la sesión de usuario 3 se actualice.

30 **La Figura 11** proporciona un ejemplo de diagrama secuencial de las diversas etapas típicas que resultan del procesamiento de solicitudes de usuario final y acciones desde sistemas externos 5 en el caso de únicamente una actualización de almacenamiento. Este diagrama secuencial se refiere al escenario y flujos de comunicación ilustrados por la Figura 10.

35 Las etapas 1 a 4 son idénticas a las etapas 1 a 4 de la Figura 5.

Las etapas 5, 6 son idénticas a las etapas 9, 10 de la Figura 5.

Las etapas 7 a 10 son idénticas a las etapas 11 a 14 de la Figura 5.

40 Las etapas 11 a 16 son idénticas a las etapas 15 a 20 de la Figura 5.

En la siguiente etapa, ya que no había ningún dato en la cola de eventos 9, el sincronizador 4 únicamente tiene que fusionar datos de sesión con el expediente almacenado recientemente recuperado.

45 Las etapas 17 a 19 son idénticas a las etapas 22 a 24 de la Figura 5.

La Figura 12 muestra un ejemplo de arquitectura de sistema detallada a la que se aplica la invención ventajosamente, con flujos de comunicación para una secuencia de tareas típicas a realizar en el caso de una actualización asíncrona por un sistema externo y sin sesión de usuario 3.

50 En la etapa 1, los datos que corresponden a una transacción se actualizan en un sistema externo. El sistema externo envía los datos que corresponden a la transacción y la referencia al correspondiente expediente al punto de entrada externo. Los datos enviados por el sistema externo se almacenan temporalmente en la cola de eventos 9 a través del manejador de notificaciones y el manejador de eventos.

55 La etapa 2 es idéntica a la etapa 4 de la Figura 8.

Las etapas 3 a 5 son idénticas a las etapas 5 a 7 de la Figura 8.

60 La Figura 12 muestra que, incluso si no hay ninguna sesión de usuario 3 activa, puede hacerse una sincronización con un sistema externo a través de una tarea planificada 14.

65 Por lo tanto, una ventaja adicional de la invención es la posibilidad de que los sistemas externos 5' notifiquen y envíen actualizaciones al sistema principal y actualicen un expediente de usuario dado en cualquier momento, tanto si hay una sesión de usuario 3 activa como si no la hay. Por lo tanto, datos desde los sistemas externos 5' se guardan automáticamente en el expediente almacenado.

5 **La Figura 13** proporciona un ejemplo de diagrama secuencial de las diversas etapas típicas que resultan del procesamiento de solicitudes de usuario final y acciones desde sistemas externos 5' en el caso de una actualización asíncrona por un sistema externo y sin sesión de usuario 3. Este diagrama secuencial se refiere al escenario y flujos de comunicación ilustrados por la Figura 12.

Las etapas 1, 2 son idénticas a las etapas 7, 8 de la Figura 9.

10 Las etapas 3 a 11 son idénticas a las etapas 15 a 23 de la Figura 9.

Por lo tanto, la invención divulgada proporciona una solución para asegurar datos recibidos desde sistemas externos y evitar guardar datos no deseados tal como datos de sesión no validados, mientras permite que actualizaciones de almacenamiento sean visibles al usuario.

REIVINDICACIONES

- 5 1. Un método de actualización de una sesión de usuario (3) con datos externos recibidos desde sistemas externos (5, 5'), **caracterizado por que** comprende las siguientes etapas realizadas con al menos un procesador de datos:
- 10 - recibir, en un sincronizador (4) en comunicación con la sesión de usuario (3) y un almacenamiento de datos (1), una solicitud desde la sesión de usuario (3) para recuperar un expediente de usuario almacenado del almacenamiento de datos (1), en donde un expediente es un conjunto de datos;
- 15 - recuperar, en el sincronizador (4), el expediente de usuario almacenado del almacenamiento de datos (1), crear, en el sincronizador (4), a partir del expediente de usuario almacenado un expediente intermedio que comprende al menos el contenido del expediente de usuario almacenado y distribuir el expediente intermedio a la sesión de usuario (3), como un expediente de sesión, para operaciones por un usuario (2);
- 20 - almacenar temporalmente, en una cola de eventos (9) en comunicación con el sincronizador (4), datos externos recibidos y/o actualizar, en el almacenamiento de datos (1), el expediente de usuario almacenado con datos externos recibidos;
- 25 - recibir, en el sincronizador (4), una actualización de usuario realizada en el expediente de sesión, actualizar el expediente de sesión con la actualización de usuario recibida y devolver el expediente de sesión actualizado a la sesión de usuario;
- 30 - tras recibir, en el sincronizador (4), una solicitud de actualización enviada por la sesión de usuario (3):
- recuperar, en el sincronizador (4), el expediente de usuario almacenado actualizado,
 - recuperar, en el sincronizador (4), todos los eventos en la cola de eventos (9),
 - fusionar, en el sincronizador (4), el expediente de usuario almacenado actualizado recuperado con actualizaciones desde todos los eventos recuperados de la cola de eventos (9) y guardar el expediente fusionado en el almacenamiento de datos (1), y
- 35 - fusionar, en el sincronizador (4), el expediente fusionado con el expediente de sesión actualizado y devolver el resultado a la sesión de usuario; y
- tras recibir, en la sesión de usuario (3), una instrucción para cerrar la sesión de usuario (3), hacer una pregunta al usuario a través de una interfaz de usuario de la sesión de usuario (3) acerca de guardar o no guardar el expediente de sesión:
- tras recibir una instrucción desde el usuario para guardar el expediente de sesión, actualizar el expediente de sesión a través de la fusión de datos externos recibidos, el expediente almacenado y el expediente de sesión, y guardar el mismo en el almacenamiento de datos (1); y
 - tras no recibir una instrucción para guardar el expediente de sesión o tras recibir una instrucción para borrar el expediente de sesión, no guardar el expediente de sesión y no actualizar el expediente almacenado en el almacenamiento de datos (1).
- 40 2. Producto de programa informático almacenado en un medio de memoria legible por ordenador no transitorio y que realiza el método según la reivindicación 1.
3. Un sincronizador para actualizar una sesión de usuario (3) con datos externos recibidos en un almacenamiento de datos (1) desde sistemas externos (5, 5') configurados para realizar el método según la reivindicación 1.

Técnica anterior

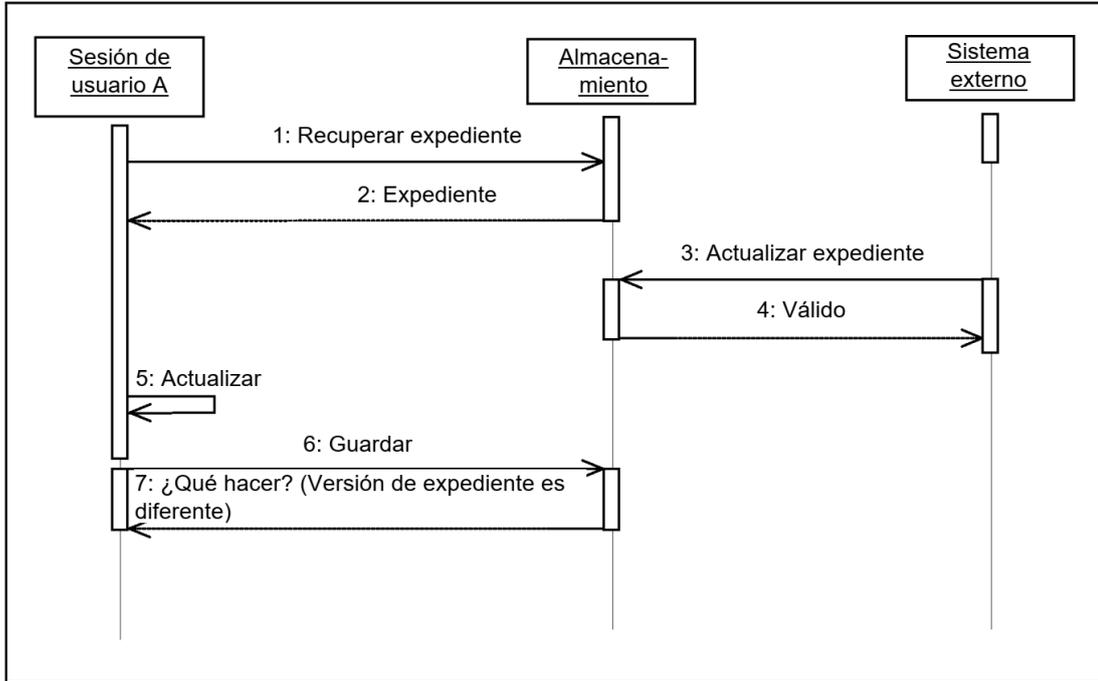


Figura 1A

	No guardar nada	Guardar todo
Tratar conflicto de versiones	Datos de sistema externo no son seguros ☹	Datos no deseados pueden guardarse ☹
	Nuevos datos de almacenamiento ni se muestran ☹	Actualización de almacenamiento es visible a usuario ☺
Rechazar conflicto de versiones	Riesgo de pérdida de datos de usuario en momento de guardado ☹ Inefectivo para muchos accesos concurrentes ☹	

Figura 1B

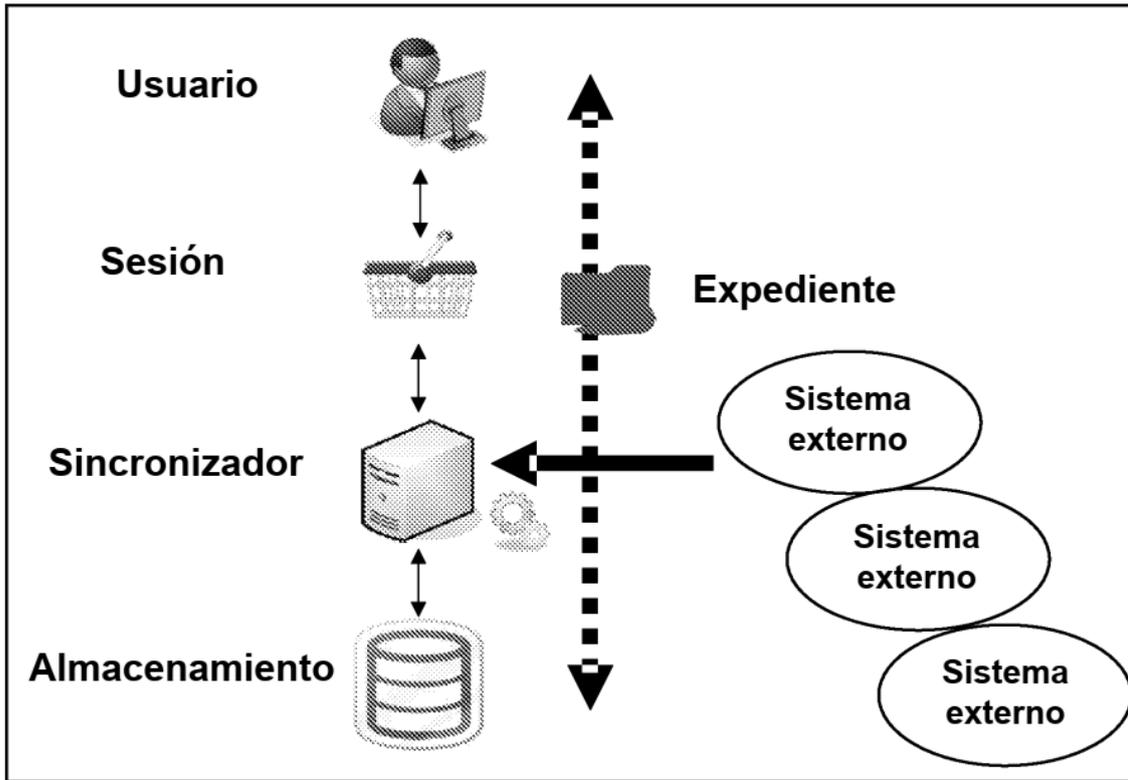


Figura 2

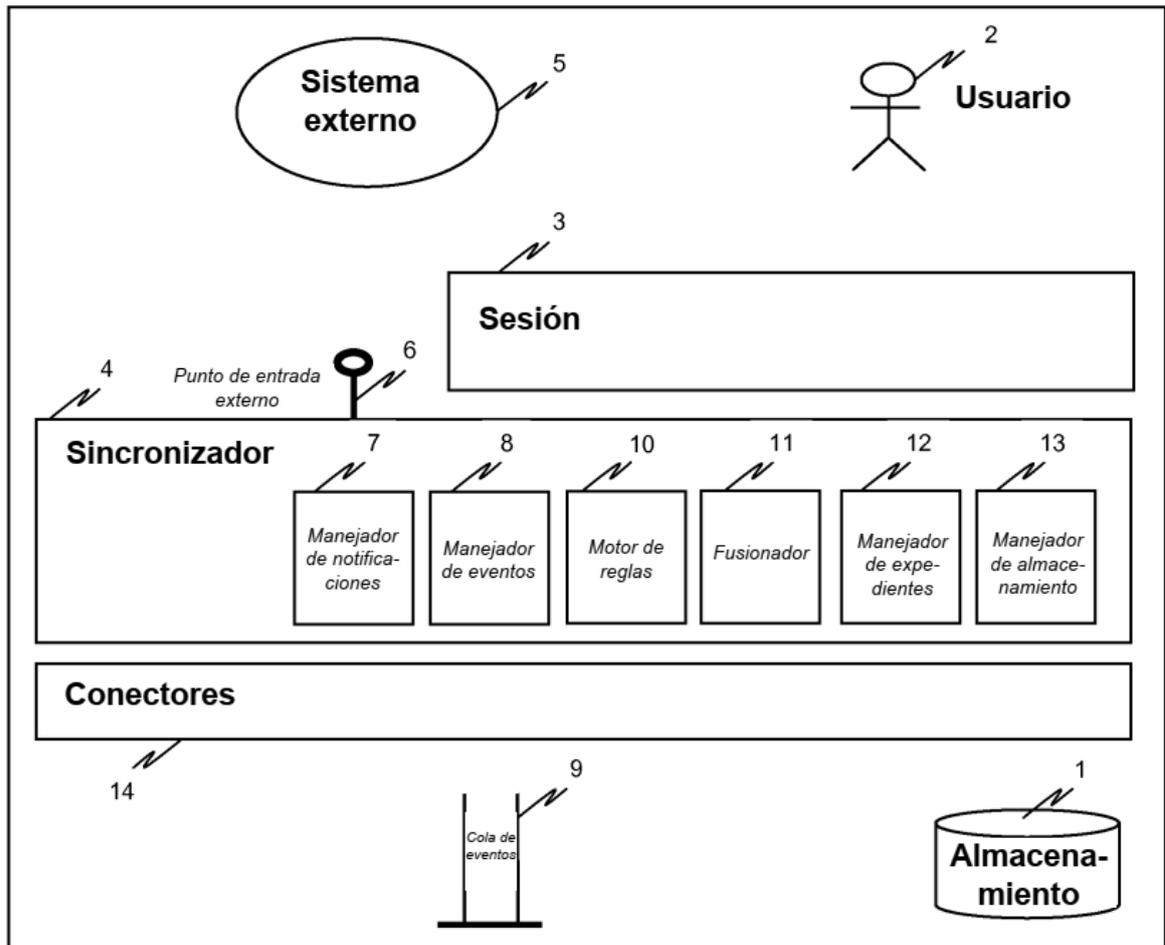


Figura 3

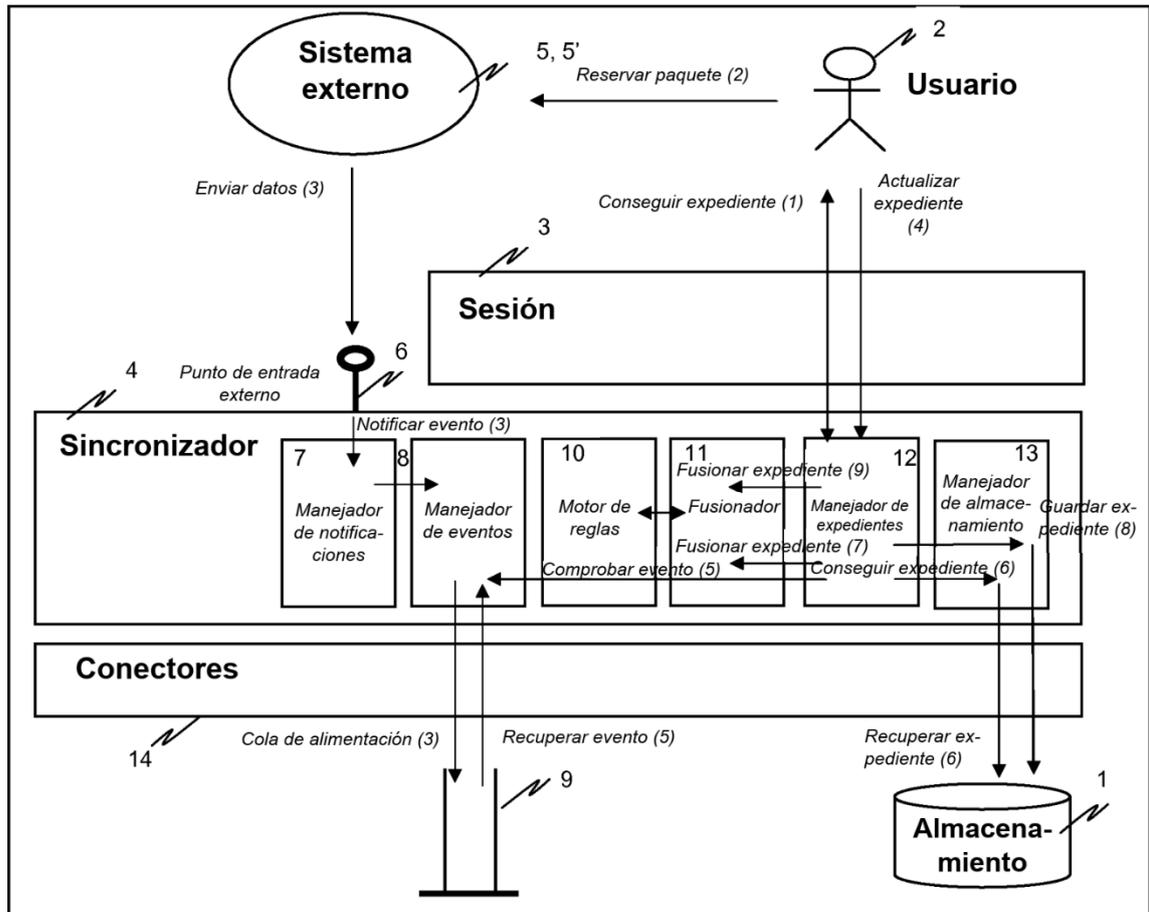


Figura 4

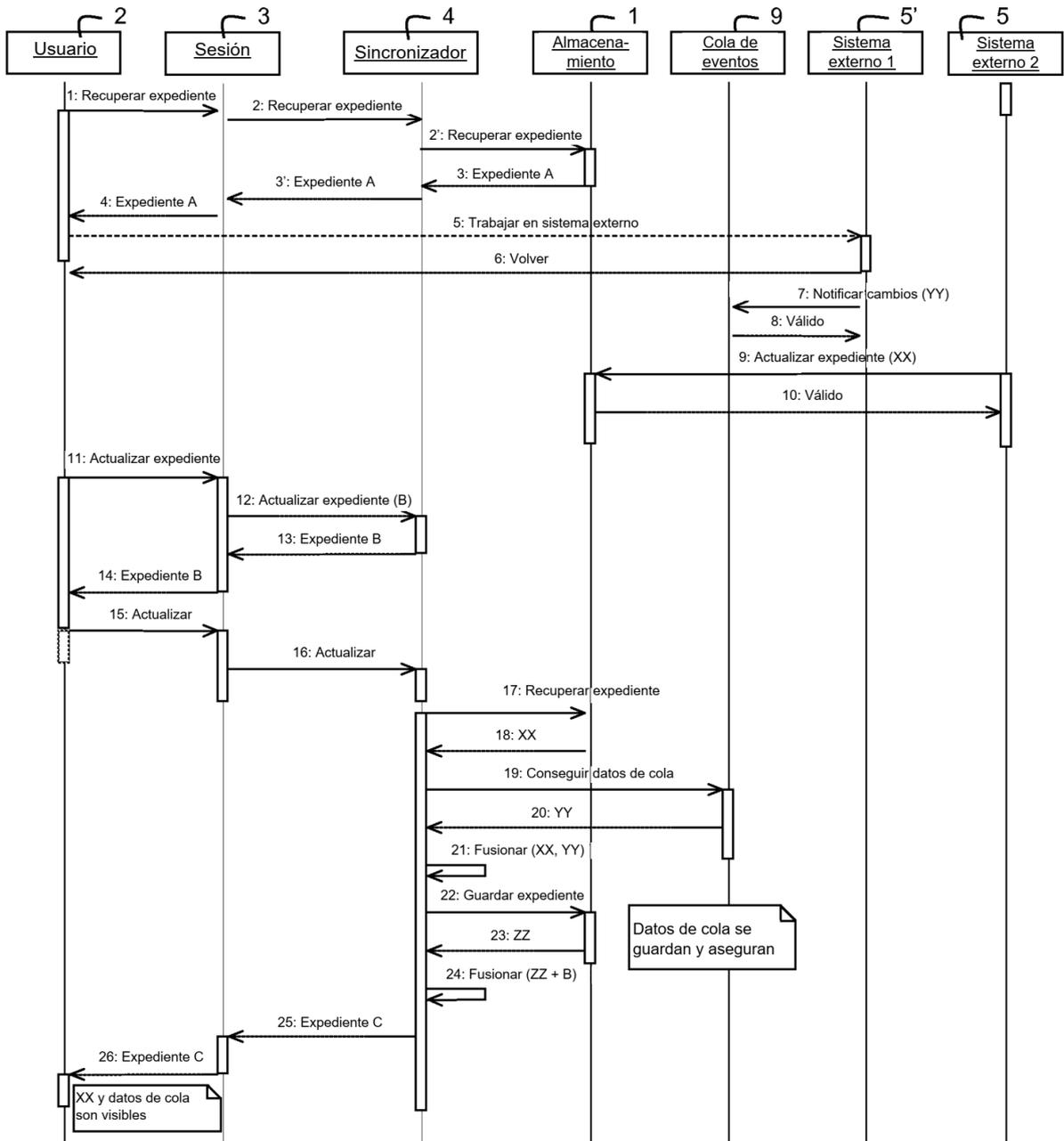


Figura 5

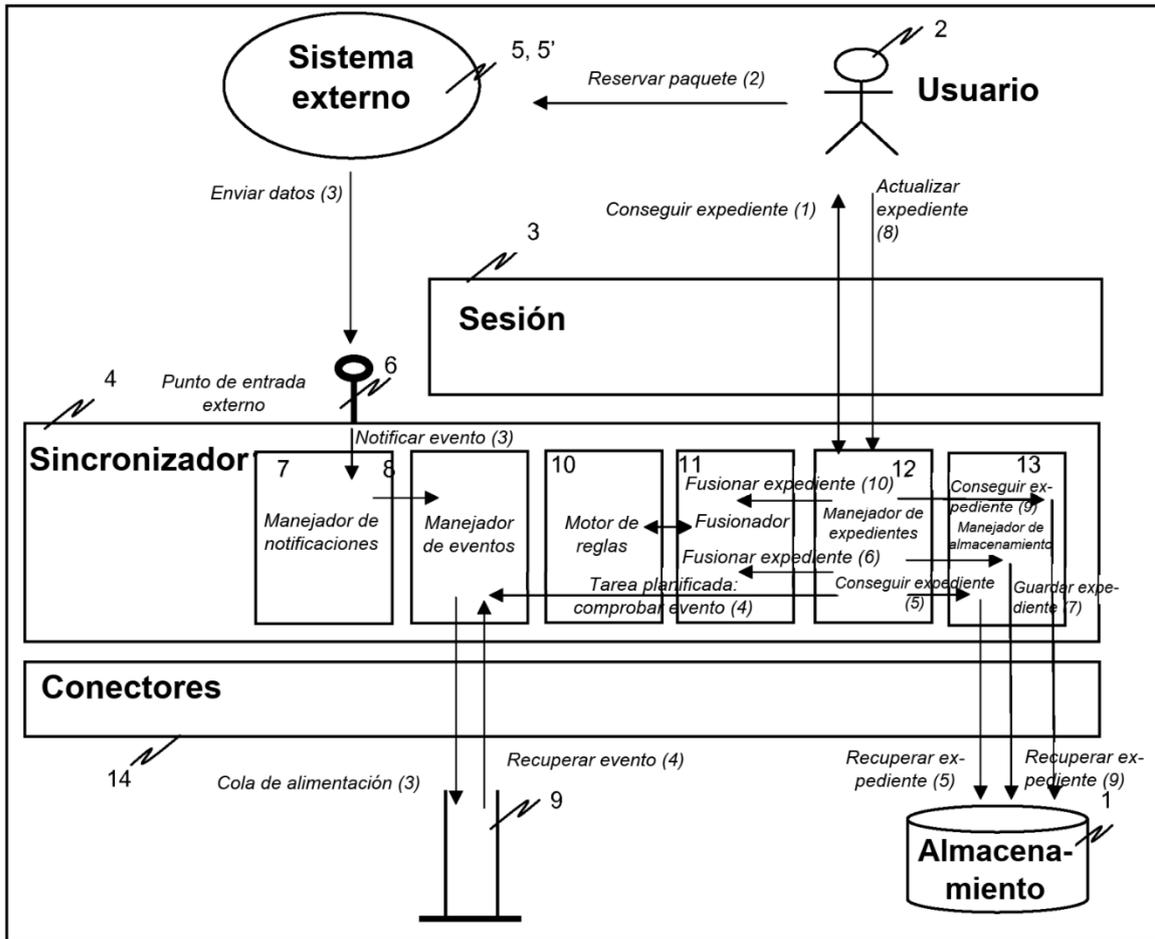


Figura 8

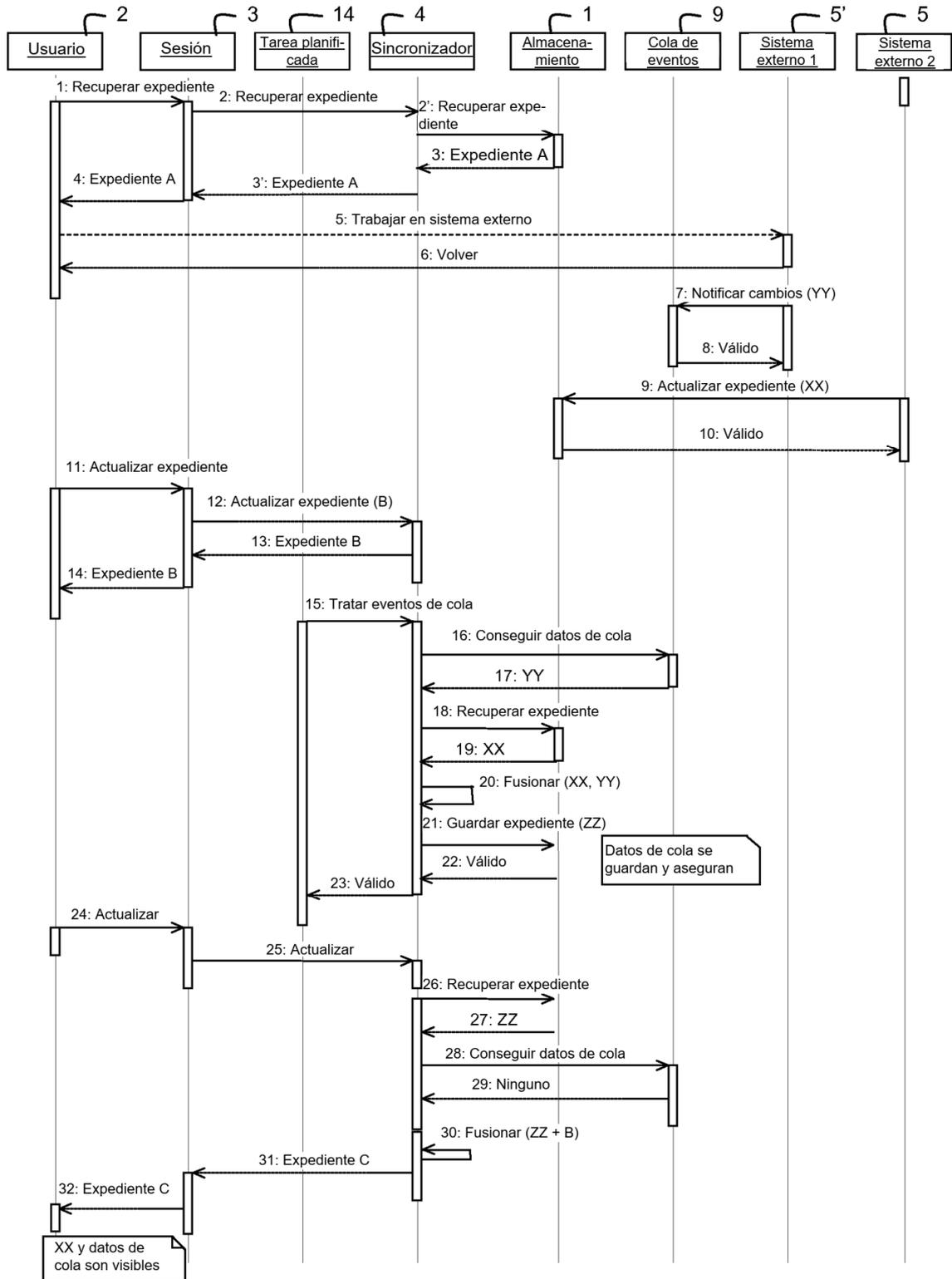


Figura 9

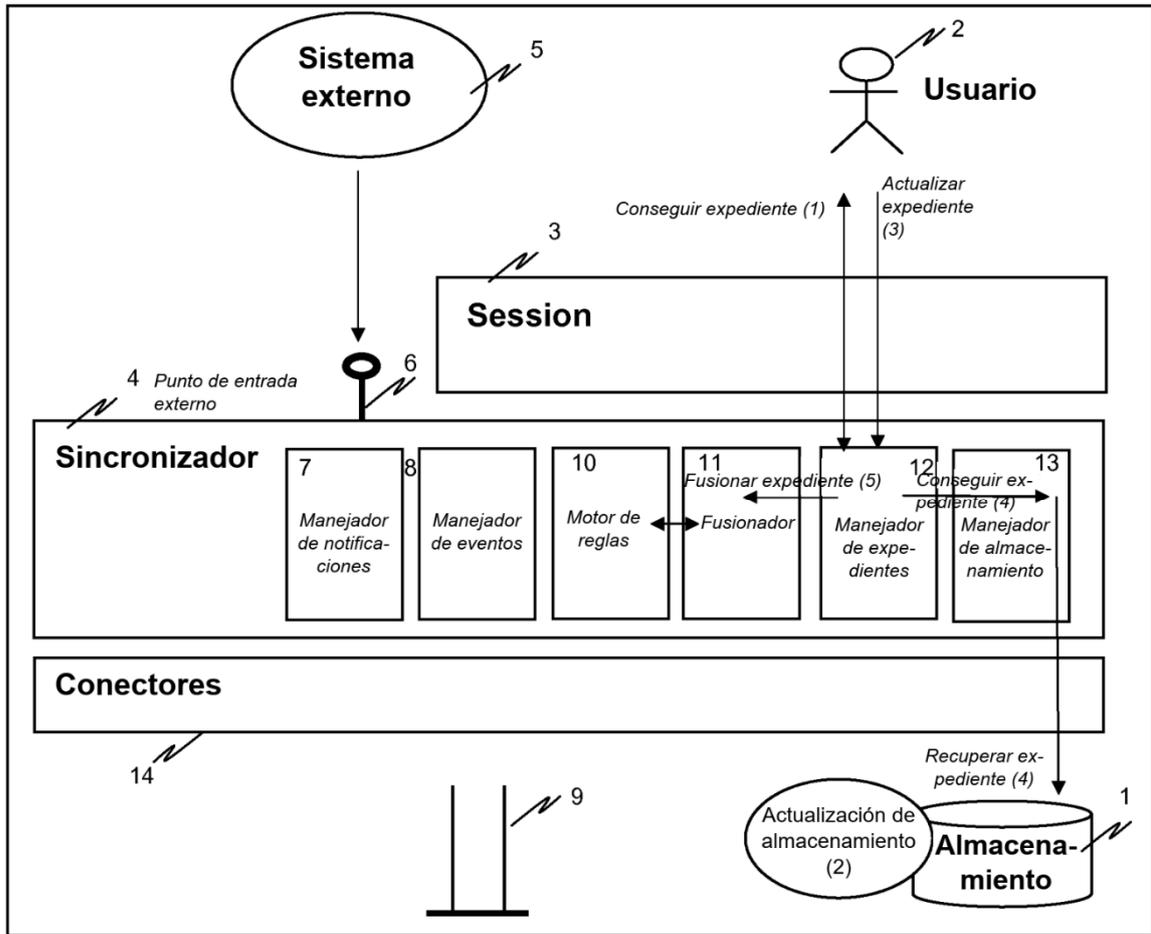


Figura 10

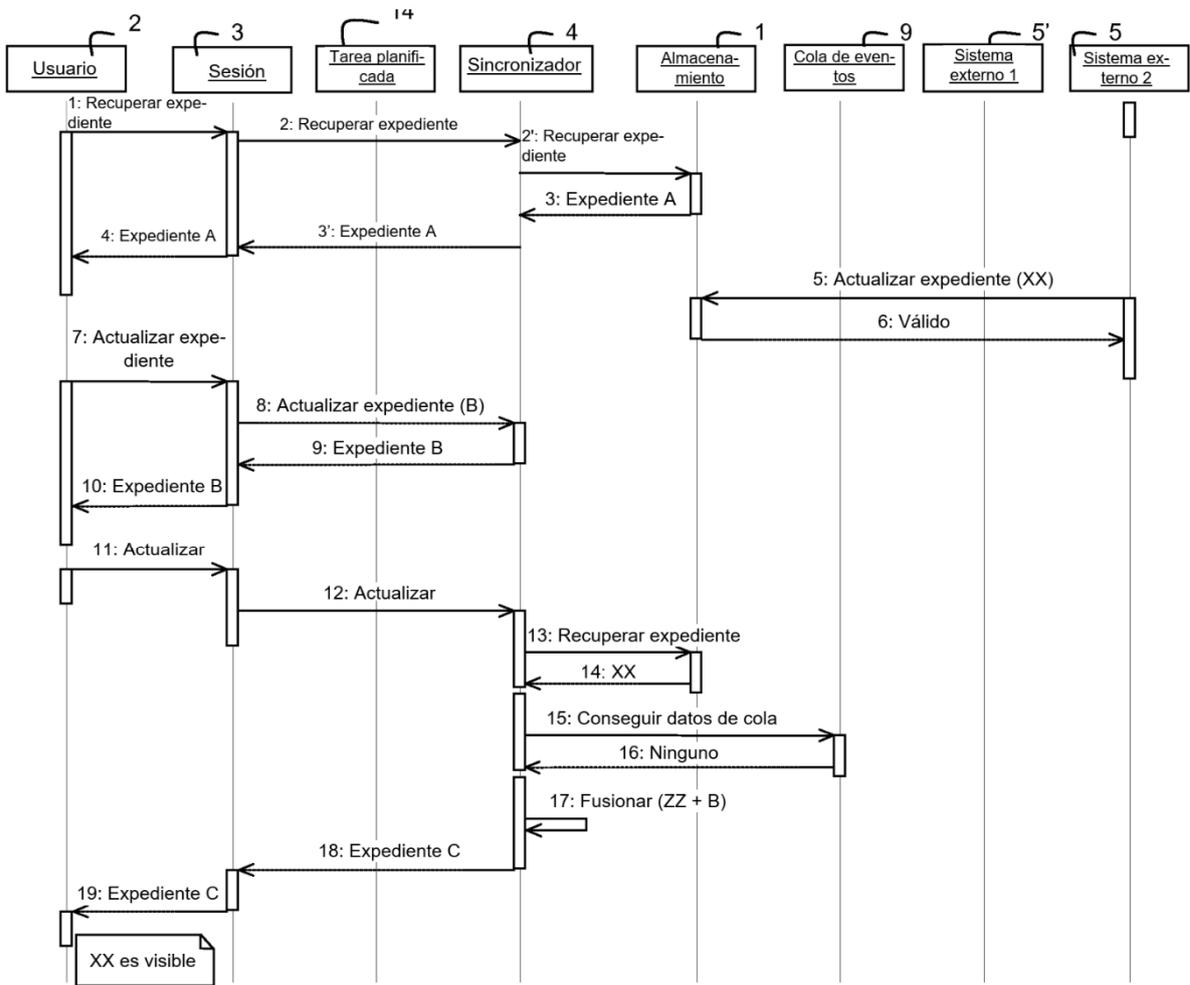


Figura 11

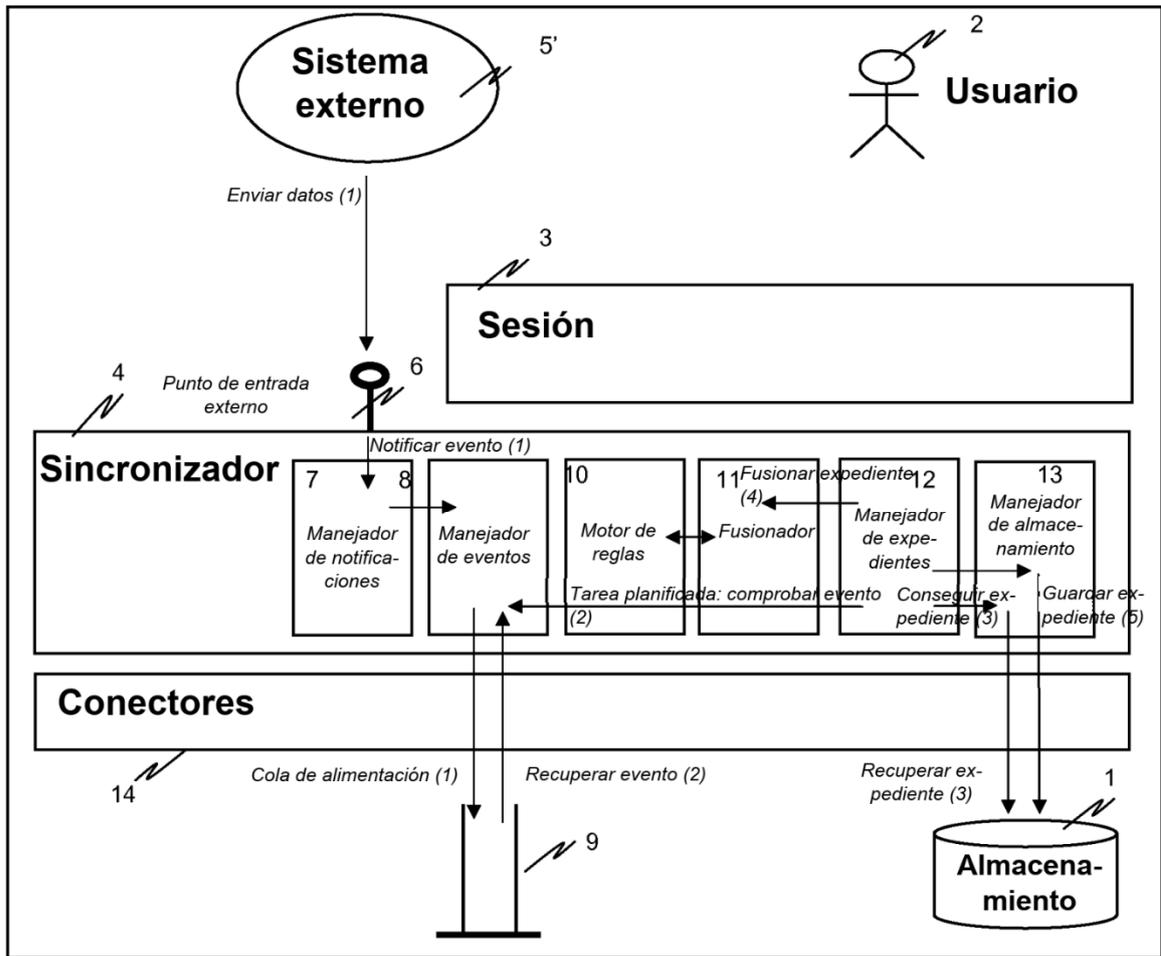


Figura 12

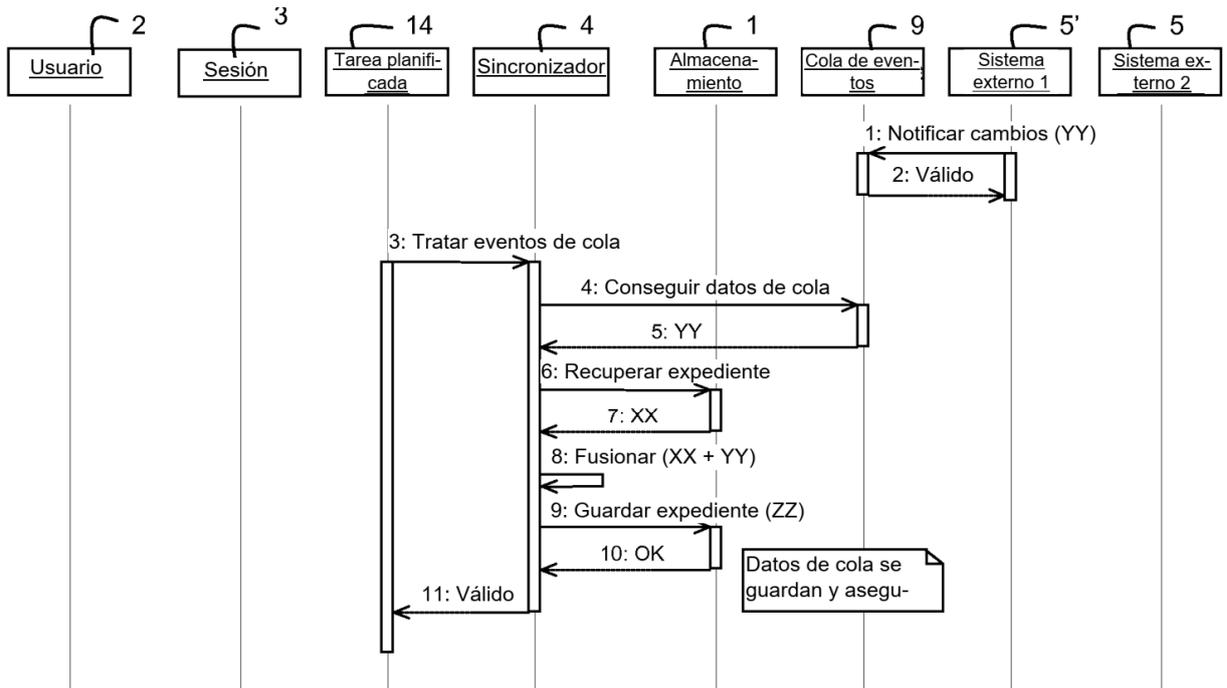


Figura 13