

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 548 087**

51 Int. Cl.:

G06F 12/16 (2006.01)

G06F 11/14 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **14.10.2003 E 03770166 (1)**

97 Fecha y número de publicación de la concesión europea: **15.07.2015 EP 1559019**

54 Título: **Método, sistema y software para registrar diariamente objetos de sistema**

30 Prioridad:

14.10.2002 NZ 52198302

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

13.10.2015

73 Titular/es:

**MAXIMUM AVAILABILITY LIMITED (100.0%)
46 MULGAN WAY, BROWNS BAY
AUCKLAND, NZ**

72 Inventor/es:

**TARBELL, JAMES SCOTT y
GARVEY, DAVID JOHN**

74 Agente/Representante:

VALLEJO LÓPEZ, Juan Pedro

ES 2 548 087 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, sistema y software para registrar diariamente objetos de sistema

5 **Campo de la invención**

La presente invención se refiere a un método, sistema y software para registrar diariamente la creación, cambio y borrado de objetos de sistema. Más particularmente, pero no exclusivamente, la presente invención se refiere a un método, sistema y software para registrar diariamente cambios a objetos de sistema de OS/400 generales (incluyendo objetos de programa, objetos de configuración, colas y objetos mapeados de espacio/memoria) para replicar estos cambios en el mismo o en un sistema remoto.

Antecedentes de la invención

15 Shultz analiza en el documento 'An approach and mechanism for auditable and testable advanced-transaction processing systems', IEEE Transactions on Software Engineering, Vol. SE-13, N° 6, junio de 1987, un método para usar un módulo de sonda para llevar a cabo auditoría de un sistema. La Patente de Estados Unidos 5.307.498 analiza un sistema que usa enganches en software para capturar datos.

20 El sistema operativo IBM OS/400 proporciona registro diario de cambios de base de datos mediante funciones de sistema integradas. Esta función de registro diario está principalmente orientada a grabar cambios de nivel de registro de base de datos para recuperación de sistema, control de entrega (por ejemplo para asegurar límites de transacción), auditoría y para soportar la repetición de cambios a una base de datos de réplica (remota o local). Otros objetos de sistema, tales como programas y objetos de configuración, no se soportan mediante la función de registro diario de base de datos. Para estos otros objetos de sistema se mantiene un *Registro Diario de Auditoría de Sistema* independiente. La creación, borrado y cambio de objetos de sistema puede grabarse en el Registro Diario de Auditoría para el fin principal de proporcionar una auditoría de actividad relacionada con estos objetos. Cuando se observa con la intención de proporcionar replicación de estos objetos a un remoto o local (copia) el Registro Diario de Auditoría tiene varias desventajas significativas, en concreto:

30 1. Las entradas del Registro Diario de Auditoría se depositan y ponen a disposición de otros procesos en el sistema únicamente después de que se ha realizado la actividad del objeto asociado. No hay manera de capturar la actividad del objeto "como ocurre". Esto hace difícil el procesamiento adicional del objeto mediante otro proceso de sistema puesto que el proceso, que está realizando la actividad del objeto, es probable que bloquee y/o use el objeto antes de que pueda obtenerse la entrada de Registro Diario de Auditoría relacionada mediante algún proceso de monitorización.

35 2. Las entradas del Registro Diario de Auditoría son en "todo el sistema" - hay un registro diario para todos los objetos en todo el sistema. Esto requiere que cualquier proceso de monitorización recupere todas las entradas de registro diario incluso si únicamente se requiere un pequeño subconjunto para la replicación.

40 3. Las entradas del Registro Diario de Auditoría no contienen suficiente información para realizar la actividad de objeto asociado en otro sistema (o copia local). Esto requiere que cualquier proceso de monitorización debe intentar localizar y bloquear el objeto asociado en un intento para hacer una copia antes de que se realice otro cambio en el objeto. Si no puede realizarse una copia del estado exacto del objeto, este estado se "pierde" y un proceso de replicación no podrá proporcionar el estado de objeto correcto a un sistema remoto (o copia local).

45 4. La única manera de poner en serie las entradas de Registro Diario de Auditoría con un registro diario de base de datos es intentar usar las indicaciones de tiempo de entrada de registro diario para fusionar las entradas del Registro Diario de Auditoría con las entradas de registro diario de base de datos asociadas. Esto puede producir tara significativa al procesar las entradas para replicación. Adicionalmente, cuando un sistema tiene múltiples procesadores las indicaciones de tiempo contenidas en registros diarios independientes pueden no reflejar con precisión la secuencia exacta de operaciones.

50 Dadas estas desventajas, no está disponible un método infalible para sincronizar cambios de objeto de sistema con cambios de base de datos asociados usando el Registro Diario de Auditoría. Un método para capturar el contenido y estado de los objetos de sistema usando el mismo registro diario de base de datos como se usa para capturar los cambios de nivel de registro de base de datos reales aseguraría que los cambios de la base de datos y de objetos pudieran replicarse con precisión a un sistema remoto (o copia local).

55 Puesto que los objetos de sistema de OS/400 se crean, cambian y borran usando un conjunto de comandos convencional (finito), la solución más evidente para obtener la información de estado (o hacer una copia de un objeto) es proporcionar comandos de sustitución o implementar un comando *salir de programa*. Existen desventajas significativas para ambos de estos enfoques.

60 El enfoque de comandos de sistema de sustitución tiene las siguientes desventajas:

1. Las interfaces de parámetros de comandos de sistema para los programas de procesamiento de comandos de OS/400 asociados pueden (y a menudo lo hacen) cambiar con cada versión de OS/400. Esto produciría dependencia significativa entre el software de replicación y una versión dada de OS/400.

5 2. El número y complejidad de los comandos que necesitarían sustituirse es alto (sobre 150 comandos, varios con listas anidadas de parámetros). Cada comando necesitaría su propio programa de procesamiento de sustitución así como esfuerzo significativo para asegurar que cada parámetro se procesa correctamente (por ejemplo como si fuera mediante el comando de OS/400 original).

10 El enfoque de programa de punto de salida de comando tiene las siguientes desventajas:

1. Ninguno de los dos puntos de salida registrados proporcionados mediante OS/400 permiten que el programa de salida asociado se active después de que se ha ejecutado el comando. Por lo tanto, en caso de la creación de objeto, cambio, comandos de borrado, el programa de salida no puede procesar el objeto resultante.

15 2. El punto de salida QIBM_QCA_CHG_COMMAND está limitado también en cuanto al número de programas de salida que pueden registrarse. Esto evitaría que algunos clientes usen este punto de salida si otro software usa el punto de salida.

20 Es un objeto de la presente invención proporcionar un método, sistema y software para registrar diariamente objetos de sistema que supera las desventajas anteriores o al menos proporcionar al público con una elección útil.

20 Sumario de la invención

25 De acuerdo con un primer aspecto de la invención se proporciona un método implementado por ordenador para registrar diariamente cambios a objetos de sistema (16) en un sistema operativo con un procesador, incluyendo el método las etapas de, y caracterizado por, el procesador:

i) ejecutar una función ficticia (6) en lugar de una función de sistema (12) cuando se solicita la función de sistema (12);

30 ii) ejecutar la función de sistema (12) bajo control de la función ficticia (6); y

iii) generar mediante la función ficticia (6), copias de objetos de sistema (16) cambiados por la ejecución de la función de sistema (12), para registro diario.

35 La función ficticia puede sustituir la función de sistema teniendo un nombre de solicitud duplicado, y adelantando la ejecución de la función de sistema.

Un punto de salida puede asociarse con la función ficticia y puede registrarse un programa de salida para el punto de salida de modo que durante la operación de la función ficticia el programa de salida puede ejecutarse.

40 El programa de salida puede manejar la ejecución de la función de sistema y capturar cambios a objetos de sistema que tienen lugar durante tal ejecución. Las copias de los cambios se generan mediante el programa de salida y pueden grabarse en disco o transmitirse en modo de flujo continuo directamente a un sistema de base de datos para registro diario.

45 Como alternativa, el programa de salida puede usarse para recuperar una cadena de comando para la función de sistema y transmitir esta de vuelta a la función ficticia. La función ficticia puede a continuación manejar la ejecución de la función de sistema y capturar cambios a objetos de sistema que tienen lugar durante tal ejecución. En un caso de este tipo, las copias de los cambios se generan mediante la función ficticia.

50 Preferentemente, la función ficticia incluye un comando de sustitución y un programa de procesamiento de comando de sustitución.

El sistema de base de datos puede incorporarse con un sistema de replicación y puede replicar los cambios a otras bases de datos locales o remotas.

55 Los mensajes o excepciones generados mediante la función de sistema pueden capturarse en una cola.

La función ficticia puede completar la ejecución reenviando cualquier mensaje o excepciones generadas mediante la función de sistema de vuelta al proceso que solicitó la función de sistema.

60 Los objetos de sistema incluyen objetos de programa, objetos de configuración, colas y objetos mapeados de espacio/memoria.

Los cambios a los objetos de sistema incluyen creación, cambio y borrado de objetos de sistema.

65 Preferentemente las funciones de sistema son aquellas encontradas en un procesador de OS/400.

Preferentemente el método incluye las etapas de:

- i) ejecutar la función de sistema durante la que tienen lugar los cambios a objetos de sistema; y
- ii) registrar diariamente cambios a objetos de sistema durante la ejecución de la función de sistema.

Una manera que pueden registrarse diariamente cambios a objetos de sistema durante la ejecución de la función de sistema es integrando comandos de registro diario en el código de las funciones de sistema.

Otra manera que pueden registrarse diariamente cambios a objetos de sistema durante la ejecución de la función de sistema es asociando puntos de salida con la función de sistema de modo que durante la ejecución de la función de sistema puede solicitarse un programa de salida para registrar diariamente los objetos de sistema.

De acuerdo con un segundo aspecto de la invención se proporciona un sistema para registrar diariamente cambios a objetos de sistema (16) caracterizado por que el sistema incluye:

- i) un procesador adaptado para ejecutar una función ficticia (6) en lugar de una función de sistema (12) cuando se solicita la función de sistema (12), en el que la función ficticia (6) ejecuta la función de sistema (12) y genera copias de objetos de sistema (16) cambiados por la ejecución de la función de sistema (12) para registro diario; y
- ii) memoria para uso mediante el procesador durante la ejecución.

Preferentemente el procesador es un procesador AS/400.

De acuerdo con una realización de la invención se proporciona software para efectuar el método del primer aspecto de la invención.

Breve descripción del dibujo

La invención se describirá ahora a modo de ejemplo con referencia al dibujo adjunto en que:

La Figura 1: muestra una ilustración de registro diario de objetos en relación con un objeto de clase de OS/400.

La Figura 2: muestra una ilustración de una realización alternativa de la invención.

Descripción detallada de realizaciones preferidas

La siguiente descripción describe un método de registro diario de objetos de sistema que opera bajo el sistema operativo OS/400. Se apreciará que el método es aplicable a otros sistemas con modificaciones apropiadas.

El método usa un enfoque de combinación para conseguir el resultado deseado; reduciendo significativamente las desventajas asociadas con cualquier enfoque único.

Un resumen de las etapas del método es como sigue:

- i. Se realiza un duplicado exacto de cada comando de OS/400 asociado con cambios de objeto en una nueva biblioteca.
- ii. El programa de procesamiento de comandos de cada comando duplicado se cambia a un programa proporcionado de proveedor común que no necesita procesar ninguno de los parámetros de comando (y por lo tanto no se ve afectado por cambios de parámetros de comando).
- iii. Se registra un programa de punto de salida QIBM_QCA_RTV_COMMAND para cada comando (duplicado) en la nueva biblioteca. Esto significa que el programa de salida se solicitará antes del Programa de Procesamiento de Comandos asociado (por ejemplo el programa especificado en la etapa ii).
- iv. La nueva biblioteca de comandos se coloca en la lista de búsqueda de biblioteca de sistema, *por encima* de la biblioteca de sistema de OS/400 normal (QSYS). Esto produce que los usuarios de sistema y programas de aplicación normales invoquen los comandos desde la nueva biblioteca en lugar de sus homólogos en la biblioteca de sistema de OS/400 (QSYS).
- v. El programa de punto de salida usa la cadena de comandos pasada, para ejecutar el comando especificado usando el comando de OS/400 original (en la biblioteca de sistema de OS/400). El programa de punto de salida puede a continuación realizar su propio procesamiento (para capturar los cambios de objeto) después de que se ha ejecutado el comando de OS/400. Cualquier mensaje enviado mediante el comando de OS/400 al programa de punto de salida se almacena en una cola temporal de modo que pueden "reenviarse" al solicitante original (por ejemplo el usuario y/o programa de aplicación que enviaron el comando).

vi. El Programa de Procesamiento de Comandos de sustitución (duplicado) se solicita a continuación mediante OS/400 (cuando el programa de punto de salida se ha completado). El Programa de Procesamiento de Comandos (sustitución) común reenvía simplemente cualquier mensaje contenido en la cola temporal (colocado ahí en la etapa v).

5 En una realización alternativa del método las etapas v y vi pueden sustituirse con:
v. El programa de punto de salida pasa la cadena de comando, usando una cola de datos, al Programa de Procesamiento de Comandos de sustitución (duplicado).

10 vi. El Programa de Procesamiento de Comandos de sustitución (duplicado) ejecuta el comando especificado usando el comando de OS/400 original (en la biblioteca de sistema de OS/400). El Programa de Procesamiento de Comandos (sustitución), a continuación puede realizar su propio procesamiento (para capturar los cambios de objetos) después de que se ha ejecutado el comando de OS/400. Cualquier mensaje enviado mediante el comando de OS/400 al Programa de Procesamiento de Comandos (sustitución) se almacena en una cola temporal y se "reenvía" al solicitante original.

20 El método elimina la necesidad de un programa de procesamiento de comandos de sustitución personalizado para cada comando duplicado - se usa un único programa común para cada comando. El uso de la versión duplicada de los comandos para unir el programa de punto de salida, permite al programa de punto de salida (o al Programa de Procesamiento de Comandos de sustitución en la realización alternativa) controlar el procesamiento del comando de sistema real - realizar actividades de replicación tanto antes como después de que se cree/cambie/borre el objeto asociado.

25 Haciendo referencia a la Figura 1, la invención se describirá en relación a registro diario de un objeto de sistema en un sistema principal para replicación a un registro diario remoto donde el cambio para el objeto de sistema es la creación de un objeto de Clase de OS/400 mediante una aplicación de usuario y donde se usa un programa de punto de salida para controlar el procesamiento del comando de sistema real.

30 La aplicación de usuario 1 se ejecuta con una lista de búsqueda de biblioteca de sistema 2 que coloca 3 la biblioteca de comandos duplicados 4 (MAXSYS) por encima de la biblioteca de comandos de sistema operativo 5 (QSYS).

35 Esto produce que el comando sin calificar CRTCLS (Crear Descripción de Clase) 6 se vincule 7 a la versión de biblioteca de MAXSYS 4 del comando en tiempo de ejecución.

40 El programa de replicación MXIICARTET 8 se registra como un programa de salida (usando el punto de salida QIBM QCA_RTV_COMMAND definido por sistema 9) para la versión de MAXSYS del comando CRTCLS. Esto produce que se solicite 10 el programa MXIICARTET 8 ANTES de que se solicite el Programa de Procesamiento de Comandos 11 del comando. La interfaz de punto de salida pasa el texto de cadena de comando completo, como se especifica mediante la aplicación de usuario, al programa de salida 8.

45 El programa de salida 8 realiza cualquier pre-procesamiento requerido para determinar si el objeto asociado está definido para replicación. La versión de sistema el comando especificado 12 (en este caso, el comando CRTCLS) se ejecuta 13 mediante el programa MXIICARTET 8 y cualquier mensaje generado 14 por el comando de sistema se almacena en una cola temporal 15.

El objeto afectado 16 (la nueva descripción de clase en este ejemplo) se graba 17 a un *fichero de respaldo* temporal mediante el programa MXIICARTET.

50 El *fichero de respaldo* temporal a continuación se copia 18 a un fichero de flujo 19 del Sistema de Ficheros Integrado (IFS) de OS/400 temporal que se registra diariamente al registro diario 20 usado mediante la configuración de replicación asociada (por ejemplo el mismo registro diario como se usa para los ficheros de base de datos definidos para replicación). Esto almacena eficazmente la imagen de respaldo en el registro diario.

55 Los datos de registro diario generados a continuación se transmiten 21 al sistema o sistemas remotos para replicación usando el soporte de registro diario remoto de OS/400 convencional.

60 El programa MXIICARTET a continuación devuelve el control a su solicitante 9 (el punto de salida de sistema), que a continuación solicita 22 el Programa de Procesamiento de Comandos (para el comando CRTCLS en la biblioteca MAXSYS). Para cada comando en la biblioteca MAXSYS, el Programa de Procesamiento de Comandos es MXCPPNULL 10.

65 El programa MXCPPNULL 10 recupera 23 los mensajes almacenados en la cola temporal 15 (mediante el programa MXIICARTET), que se generaron mediante la versión de sistema convencional 12 del comando, y los envía 24 a la aplicación de usuario 1. Esto permite a la aplicación de usuario 1 procesar los mensajes exactamente como si hubieran solicitado la versión de sistema del comando directamente.

5 Haciendo referencia a la Figura 2, la invención se describirá en relación con registrar diariamente un objeto de sistema en un sistema principal para replicación a un registro diario remoto donde el cambio al objeto de sistema es la creación de un objeto de Clase de OS/400 mediante una aplicación de usuario y donde un Programa de Procesamiento de Comandos de sustitución se usa para controlar el procesamiento del comando de sistema real.

10 La aplicación de usuario 25 se ejecuta con una lista de búsqueda de biblioteca de sistema 26 que coloca 27 la biblioteca de comandos duplicados 28 (MAXSYS) por encima de la biblioteca de comandos de sistema operativo 29 (QSYS).

15 Esto produce que el comando sin calificar CRTCLS (Crear Descripción de Clase) 30 se vincule 31 a la versión de biblioteca de MAXSYS 28 del comando en tiempo de ejecución.

20 El programa de replicación MXIICARTET 32 se registra como un programa de salida (usando el punto de salida QIBM_QCA_RTV_COMMAND definido por sistema 33) para la versión de MAXSYS del comando CRTCLS. Esto produce que se solicite 34 el programa MXIICARTET 32 ANTES de que solicite el Programa de Procesamiento de Comando 35 del comando. La interfaz de punto de salida pasa el texto de cadena de comando completo, como se especifica mediante la aplicación de usuario, al programa de salida 32.

25 El programa de salida pasa la cadena de comando completa a través de una cola de datos 36 al Programa de Procesamiento de Comandos (para el comando CRTCLS en la biblioteca MAXSYS). Para cada comando en la biblioteca MAXSYS, el Programa de Procesamiento de Comandos es MXCPPNULL 35.

30 El programa MXIICARTET 32 a continuación devuelve el control a su solicitante 33 (el punto de salida de sistema), y se ejecuta el programa MXCPPNULL 35.

35 El programa MXCPPNULL 35 realiza cualquier pre-procesamiento requerido para determinar si el objeto asociado se define para replicación. La cadena de comando completa se extrae desde la cola de datos 36 y la versión de sistema del comando especificado 37 (en este caso, el comando CRTCLS) se ejecuta 38 mediante el programa MXCPPNULL 35 y cualquier mensaje generado 39 mediante el comando de sistema se almacenan en una cola temporal 40.

40 El objeto afectado 41 (la nueva descripción de clase en este ejemplo) se graba 42 a un fichero de respaldo temporal mediante el programa MXCPPNULL 35.

45 El fichero de respaldo temporal a continuación se copia 43 a un fichero de flujo 44 del Sistema de Ficheros Integrado (IFS) OS/400 temporal que se registra diariamente al registro diario 45 usado mediante la configuración de replicación asociada (por ejemplo el mismo registro diario que se usa para los ficheros de datos definidos para replicación). Esto almacena eficazmente la imagen de respaldo en el registro diario.

Los datos de registro diario generados a continuación se transmiten 46 al sistema o sistemas remotos para replicación usando el soporte de registro diario remoto OS/400 convencional.

El programa MXCPPNULL 35 recupera 47 los mensajes almacenados en la cola temporal 40, que se generaron mediante la versión de sistema convencional 37 del comando, y los envía 48 a la aplicación de usuario 25. Esto permite a la aplicación de usuario 25 procesar los mensajes exactamente como si hubieran solicitado la versión de sistema del comando directamente.

Definiciones

**Clase (Descripción de Clase)
Comando**

Un objeto que identifica los atributos de ejecución de un trabajo.
Una declaración usada para solicitar una función del sistema. Un comando consiste en la abreviatura del nombre de comando, que identifica la función solicitada y sus parámetros.

Programa de Procesamiento de Comandos (CPP)

Un programa que procesa un comando. Este programa realiza alguna comprobación de validez y procesa el comando de modo que se realiza la función solicitada.

Programa de Salida

1. Un programa escrito por el usuario que proporciona control durante la operación de una función de sistema.
2. Un programa al que se pasa el control desde punto de salida.

Fichero de respaldo

Un fichero asignado en almacenamiento auxiliar que puede usarse para almacenar datos grabados en disco (sin requerir disquetes o cintas).

Registro Diario de Auditoría de Sistema

Un registro diario usado mediante el sistema para mantener un registro de eventos relevantes de seguridad que tienen lugar.

REIVINDICACIONES

1. Un método implementado por ordenador para registro diario de cambios a objetos de sistema (16) en un sistema operativo con un procesador, incluyendo el método las etapas de, y **caracterizado por**, el procesador:
- 5 i) ejecutar una función ficticia (6) en lugar de una función de sistema (12) cuando se solicita la función de sistema (12);
 ii) ejecutar la función de sistema (12) bajo control de la función ficticia (6); y
 iii) generar, mediante la función ficticia (6), copias de objetos de sistema (16) cambiados por la ejecución de la
 10 función de sistema (12), para registro diario.
2. Un método de acuerdo con la reivindicación 1 en el que la función ficticia (6) se ejecuta en lugar de la función de sistema (12) asignando un nombre de solicitud duplicado a la función ficticia (6) y disponiendo el procesador para adelantar la ejecución de la función de sistema (12) cuando se solicita.
- 15 3. Un método de acuerdo con una cualquiera de las reivindicaciones 1 a 2 en el que la función ficticia (6) incluye un punto de salida (9) y se registra un programa de salida (8) para el punto de salida (9).
- 20 4. Un método de acuerdo con la reivindicación 3 en el que durante la operación de la función ficticia (6) el programa de salida (8) se ejecuta tras alcanzar el punto de salida (9).
5. Un método de acuerdo con la reivindicación 4 en el que la ejecución de la función de sistema (12) se maneja mediante el programa de salida (8).
- 25 6. Un método de acuerdo con la reivindicación 5 en el que los objetos de sistema (16) cambiados por la ejecución de la función de sistema (12) se capturan mediante el programa de salida (8).
7. Un método de acuerdo con la reivindicación 6 en el que el programa de salida (8) genera copias de los objetos de sistema (16) capturados mediante el programa de salida (8).
- 30 8. Un método de acuerdo con la reivindicación 4 en el que la ejecución de la función de sistema (12) se maneja mediante la función ficticia (6).
9. Un método de acuerdo con la reivindicación 8 en el que los objetos de sistema (16) cambiados por la ejecución de la función de sistema (12) se capturan mediante la función ficticia (6).
- 35 10. Un método de acuerdo con la reivindicación 9 en el que el programa de salida (8) genera copias de los objetos de sistema (16) capturados mediante la función ficticia (6).
- 40 11. Un método de acuerdo con una cualquiera de las reivindicaciones 1 a 10 en el que las copias de los objetos de sistema (16) se graban en disco.
12. Un método de acuerdo con una cualquiera de las reivindicaciones 1 a 11 en el que las copias de los objetos de sistema (16) se transmiten en modo de flujo continuo a un sistema de base de datos para registro diario.
- 45 13. Un método de acuerdo con la reivindicación 12 en el que el sistema de base de datos se incorpora con un sistema de replicación.
14. Un método de acuerdo con la reivindicación 13 en el que el sistema de replicación replica las copias de los objetos de sistema (16) a una o más bases de datos locales o remotas.
- 50 15. Un método de acuerdo con una cualquiera de las reivindicaciones 1 a 14 en el que los mensajes o excepciones generados mediante la función de sistema (12) se capturan en una cola.
- 55 16. Un método de acuerdo con la reivindicación 15 en el que la función de sistema (12) se solicita originalmente mediante un proceso y los mensajes o excepciones se reenvían de vuelta al proceso mediante la función ficticia (6).
17. Un método de acuerdo con una cualquiera de las reivindicaciones 1 a 16 en el que los objetos de sistema (16) son uno o más del conjunto de objetos de programa, objetos de configuración, colas y objetos mapeados de espacio/memoria.
- 60 18. Un método de acuerdo con una cualquiera de las reivindicaciones 1 a 17 en el que los objetos de sistema (16) cambiados son aquellos objetos de sistema que se han creado, cambiado o borrado.
- 65 19. Un método de acuerdo con una cualquiera de las reivindicaciones 1 a 18 en el que las funciones de sistema (12) son funciones de sistema de OS/400.

20. Un método de acuerdo con una cualquiera de las reivindicaciones 1 a 19 que incluye las etapas de:
- i) ejecutar la función de sistema (12) durante la que tienen lugar los cambios a objetos de sistema; y
 - ii) registrar diariamente cambios a objetos de sistema (16) durante la ejecución de la función de sistema (12).
21. Un método de acuerdo con la reivindicación 20 en el que los cambios a objetos de sistema (16) se registran diariamente integrando comandos de registro diario en el código de las funciones de sistema (12).
22. Un método de acuerdo con la reivindicación 20 en el que los cambios a objetos de sistema (16) se registran diariamente asociando puntos de salida (9) a la función de sistema (12) y solicitando un programa de salida (8) durante la ejecución de la función de sistema (12).
23. Un sistema para registrar diariamente cambios a objetos de sistema (16) **caracterizado por que** el sistema incluye:
- i) un procesador adaptado para ejecutar una función ficticia (6) en lugar de una función de sistema (12) cuando se solicita la función de sistema (12), en donde la función ficticia (6) ejecuta la función de sistema (12) y genera copias de objetos de sistema (16) cambiados por la ejecución de la función de sistema (12) para registro diario; y
 - ii) memoria para uso mediante el procesador durante la ejecución.
24. Un sistema de acuerdo con la reivindicación 23 en el que la función ficticia (6) se ejecuta en lugar de la función de sistema (12) asignando un nombre de solicitud a la función ficticia (6) que es un duplicado del nombre de solicitud de la función de sistema (12) para adelantar la ejecución de la función de sistema (12).
25. Un sistema de acuerdo con una cualquiera de las reivindicaciones 23 a 24 en el que la función ficticia (6) incluye un punto de salida (9) y se registra un programa de salida (8) para el punto de salida (9).
26. Un sistema de acuerdo con la reivindicación 25 en el que durante la ejecución de la función ficticia (6) el programa de salida (8) se ejecuta tras alcanzar el punto de salida (9).
27. Un sistema de acuerdo con la reivindicación 26 en el que la ejecución de la función de sistema (12) se maneja mediante el programa de salida (8).
28. Un sistema de acuerdo con la reivindicación 27 en el que los objetos de sistema (16) cambiados por la ejecución de la función de sistema (12) se capturan mediante el programa de salida (8).
29. Un sistema de acuerdo con la reivindicación 28 en el que el programa de salida (8) genera copias de los objetos de sistema (16) capturados mediante el programa de salida (8).
30. Un sistema de acuerdo con la reivindicación 26 en el que la ejecución de la función de sistema (12) se maneja mediante el programa ficticio (6).
31. Un sistema de acuerdo con la reivindicación 30 en el que los objetos de sistema (16) cambiados por la ejecución de la función de sistema (12) se capturan mediante el programa ficticio (6).
32. Un sistema de acuerdo con la reivindicación 31 en el que el programa de salida (8) genera copias de los objetos de sistema (16) capturados mediante el programa ficticio (6).
33. Un sistema de acuerdo con una cualquiera de las reivindicaciones 23 a 32 en el que las copias de los objetos de sistema (16) se graban en disco.
34. Un sistema de acuerdo con una cualquiera de las reivindicaciones 23 a 33 en el que las copias de los objetos de sistema (16) se transmiten en modo de flujo continuo a un sistema de base de datos para registro diario.
35. Un sistema de acuerdo con la reivindicación 34 en el que el sistema de base de datos se incorpora con un sistema de replicación.
36. Un sistema de acuerdo con la reivindicación 35 en el que el sistema de replicación replica las copias de los objetos de sistema (16) a una o más bases de datos locales o remotas.
37. Un sistema de acuerdo con una cualquiera de las reivindicaciones 23 a 36 en el que se capturan en una cola mensajes o excepciones generados mediante la función de sistema (12).
38. Un sistema de acuerdo con la reivindicación 37 en el que la función de sistema (12) se solicita originalmente mediante un proceso y los mensajes o excepciones se reenvían de vuelta al proceso mediante la función ficticia (6).

39. Un sistema de acuerdo con una cualquiera de las reivindicaciones 23 a 38 en el que los objetos de sistema (16) son uno o más del conjunto de objetos de programa, objetos de configuración, colas y objetos mapeados de espacio/memoria.

5 40. Un sistema de acuerdo con una cualquiera de las reivindicaciones 23 a 39 en el que los objetos de sistema cambiados son aquellos objetos de sistema que se han creado, cambiado o borrado.

41. Un sistema de acuerdo con una cualquiera de las reivindicaciones 23 a 40 en el que el procesador es un procesador de AS/400.

10 42. Un sistema de acuerdo con una cualquiera de las reivindicaciones 23 a 41 en el que el procesador está operando bajo el sistema operativo OS/400.

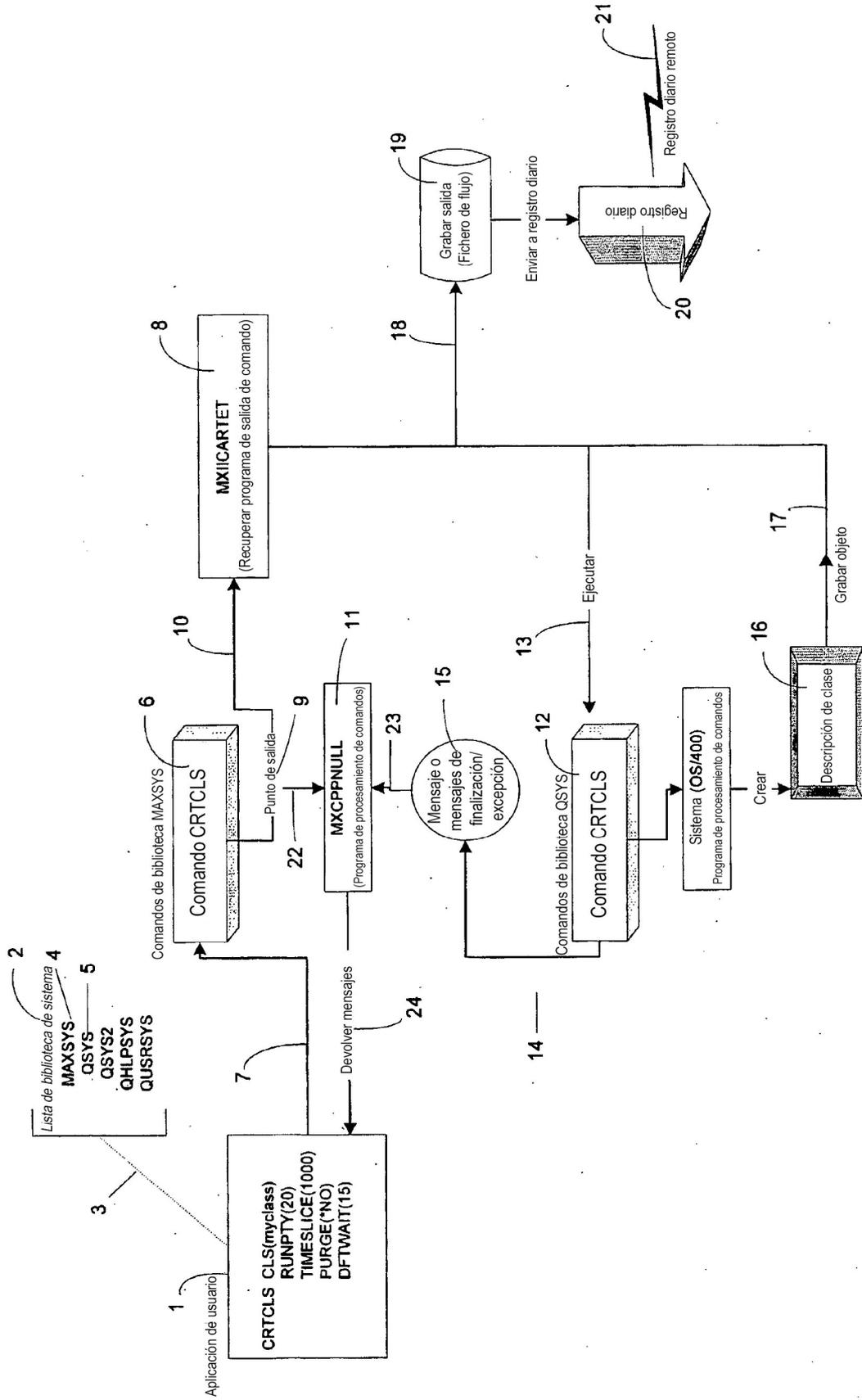


Figura 1

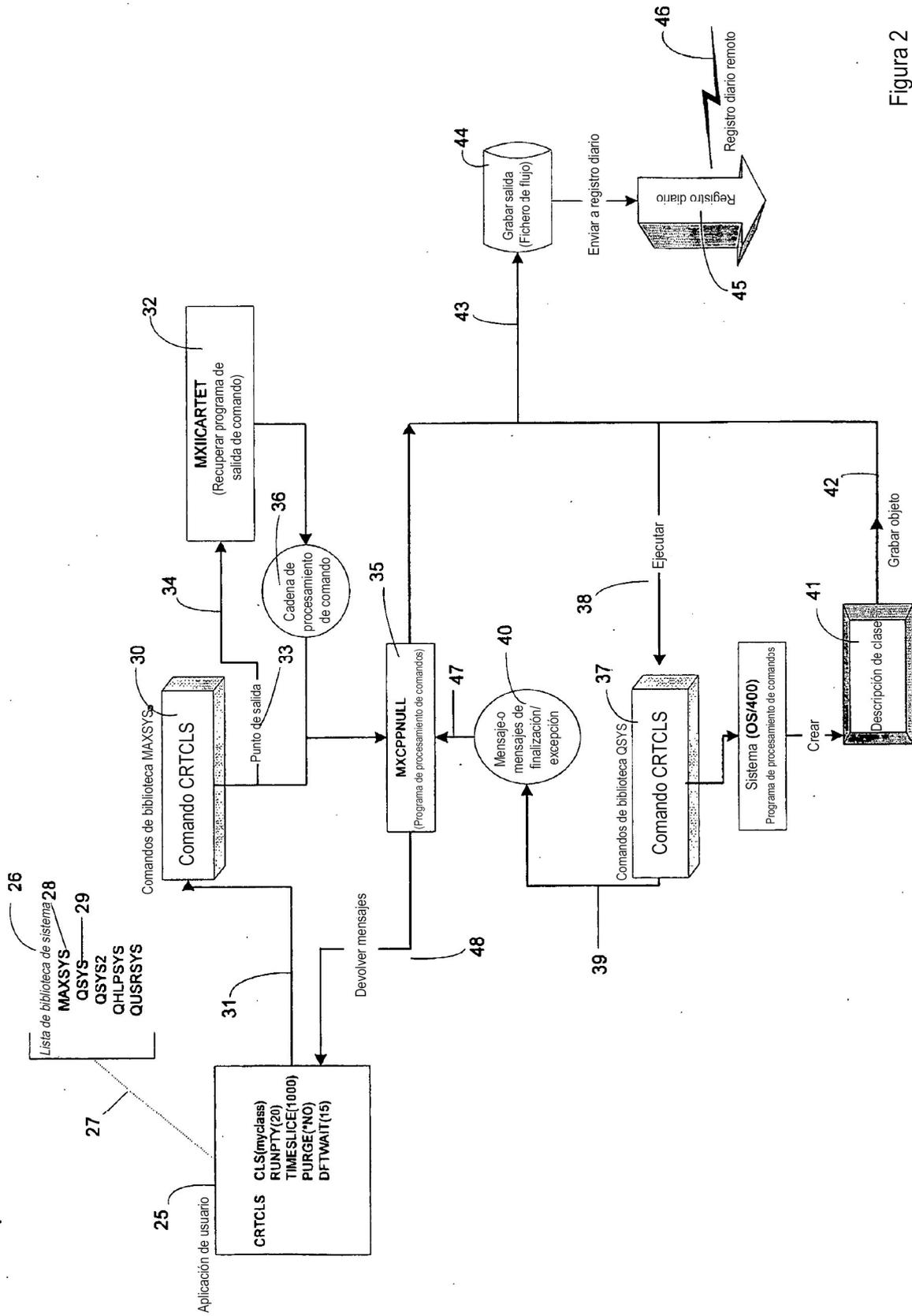


Figura 2