

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 605 936**

51 Int. Cl.:

G06F 17/30 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **29.07.2004 PCT/US2004/024565**

87 Fecha y número de publicación internacional: **24.11.2005 WO05111867**

96 Fecha de presentación y número de la solicitud europea: **29.07.2004 E 04779579 (4)**

97 Fecha y número de publicación de la concesión europea: **14.09.2016 EP 1629407**

54 Título: **Sistemas y procedimientos de mantenimiento y reparación automáticos de bases de datos o de sistemas de archivos**

30 Prioridad:

03.05.2004 US 837932

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

17.03.2017

73 Titular/es:

**MICROSOFT TECHNOLOGY LICENSING, LLC
(100.0%)**

**One Microsoft Way
Redmond, WA 98052, US**

72 Inventor/es:

**OKS, ARTEM A.;
KODAVALLA, HANUMANATHA R. y
SLEEMAN, MARTIN J.**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 605 936 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

Sistemas y procedimientos de mantenimiento y reparación automáticos de bases de datos o de sistemas de archivos

Remisión a solicitudes relacionadas

La presente solicitud reivindica la prioridad para la solicitud de EE. UU. con n.º de serie 10/837.932 presentada el 3 de mayo de 2004.

La presente solicitud está relacionada por medio de la materia objeto con las invenciones que se divulgan en las siguientes solicitudes de cesión común: la solicitud de patente de Estados Unidos n.º 10/647.058 (n.º de expediente del mandatario MSFT-1748), presentada el 21 de agosto de 2003, titulada "SYSTEMS AND METHODS FOR REPRESENTING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM BUT INDEPENDENT OF PHYSICAL REPRESENTATION"; la solicitud de patente de Estados Unidos n.º 10/646.941 (n.º de expediente del mandatario MSFT-1749), presentada el 21 de agosto de 2003, titulada "SYSTEMS AND METHODS FOR SEPARATING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM FROM THEIR PHYSICAL ORGANIZATION"; la solicitud de patente de Estados Unidos n.º 10/646.940 (n.º de expediente del mandatario MSFT-1750), presentada el 21 de agosto de 2003, titulada "SYSTEMS AND METHODS FOR THE IMPLEMENTATION OF A BASE SCHEMA FOR ORGANIZING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM"; la solicitud de patente de Estados Unidos n.º 10/646.632 (n.º de expediente del mandatario MSFT-1751), presentada el 21 de agosto de 2003, titulada "SYSTEMS AND METHODS FOR THE IMPLEMENTATION OF A CORE SCHEMA FOR PROVIDING A TOP-LEVEL STRUCTURE FOR ORGANIZING UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM"; la solicitud de patente de Estados Unidos n.º 10/646.645 (n.º de expediente del mandatario MSFT-1752), presentada el 21 de agosto de 2003, titulada "SYSTEMS AND METHOD FOR REPRESENTING RELATIONSHIPS BETWEEN UNITS OF INFORMATION MANAGEABLE BY A HARDWARE/SOFTWARE INTERFACE SYSTEM"; la solicitud de patente de Estados Unidos n.º 10/646.575 (n.º de expediente del mandatario MSFT-2733), presentada el 21 de agosto de 2003, titulada "SYSTEMS AND METHODS FOR INTERFACING APPLICATION PROGRAMS WITH AN ITEM-BASED STORAGE PLATFORM"; la solicitud de patente de Estados Unidos n.º 10/646.646 (n.º de expediente del mandatario MSFT-2734), presentada el 21 de agosto de 2003, titulada "STORAGE PLATFORM FOR ORGANIZING, SEARCHING, AND SHARING DATA"; la solicitud de patente de Estados Unidos n.º 10/646.580 (n.º de expediente del mandatario MSFT-2735), presentada el 21 de agosto de 2003, titulada "SYSTEMS AND METHODS FOR DATA MODELING IN AN ITEM-BASED STORAGE PLATFORM".

Campo técnico

La presente invención se refiere, en general, a la gestión de bases de datos y de sistemas de archivos y, más en particular, al mantenimiento y la reparación automáticos de bases de datos y de sistemas de archivos para garantizar la fiabilidad de los datos. Diversos aspectos de la presente invención se refieren a responder a, y corregir, corrupciones de datos a un nivel de página de datos para todos los tipos de página de datos, así como a la recuperación (incluyendo operaciones de reconstrucción o de restauración) para diversos escenarios incluyendo, sin limitación, corrupciones de página de índice (agrupado y no agrupado), corrupciones de página de datos y corrupciones de página en el archivo de registro.

Antecedentes

A pesar de que las plataformas de base de datos de cliente (es decir, ordenadores de sobremesa domésticos y de trabajo) usan un soporte físico de una calidad que es mucho más baja que la de las plataformas de servidor, incluso el soporte físico de clase servidor (controladores, unidades de control, discos, y así sucesivamente) pueden dar lugar a una corrupción de datos de tal modo que una operación de lectura no devuelva lo que la aplicación escribió en el almacén de datos. Es evidente que este es claramente un problema que se da con mayor frecuencia con las plataformas de base de datos de cliente (en contraposición a las plataformas de base de datos de servidor) por diversas razones, incluyendo, sin limitación, a la probabilidad aumentada de que una máquina de cliente se apague de forma arbitraria en medio de una operación de escritura debido a una pérdida inesperada de alimentación eléctrica, lo que, a su vez, conduce a páginas rasgadas y a potenciales corrupciones de base de datos (es más habitual que los sistemas de base de datos de servidor usen fuentes de alimentación ininterrumpida para mitigar los problemas derivados de las pérdidas de alimentación eléctrica). El decaimiento de medios es otra fuente de corrupciones de base de datos, en la que los medios de almacenamiento físico se desgastan con el tiempo de una forma bastante literal. Y otra fuente más de interés en lo que respecta a la fiabilidad es la detección y la recuperación frente a corrupciones causadas por los errores de soporte lógico tanto involuntarios (por ejemplo, defectos) así como perniciosos (por ejemplo, virus).

Tradicionalmente, el mantenimiento y la reparación de una base de datos ha recaído en los gestores de bases de datos y similares que tengan un conjunto de competencias bien desarrollado y un conocimiento profundo de los sistemas de base de datos, o al menos en individuos que están familiarizados con y usan con regularidad los sistemas de base de datos - en términos generales, las personas con relativamente expertas con respecto a las

tecnologías de bases de datos. Por otro lado, los usuarios finales típicos de tipo profesional y consumidor de sistemas operativos y programas de aplicación rara vez trabajan con bases de datos y están, en gran medida, mal equipados para hacer frente a cuestiones de mantenimiento y de reparación de bases de datos,

5 A pesar de que el dispar nivel de experiencia entre estos dos grupos ha sido, en gran parte, irrelevante en el pasado, un sistema de archivos implementado en base de datos para un sistema operativo - tal como el sistema operativo que se divulga en las solicitudes de patente de EE. UU. relacionadas que se han identificado en lo que antecede en el presente documento en la sección que se titula "Remisiones" - crea un escenario en el que estos usuarios finales de menor experiencia se verán enfrentados a cuestiones de mantenimiento y de reparación de bases de datos que los mismos se verán, en gran medida, incapaces de resolver. Por lo tanto, un sistema de archivos de sistema operativo implementado en base de datos, o "sistema de archivos de base de datos" (DBFS, *database file system*) para abreviar, de tipo profesional/consumidor, ha de ser capaz de detectar corrupciones y de recuperar sus bases de datos a un estado transaccionalmente consistente y, en los casos de una pérdida de datos no recuperable, entonces el DBFS ha de garantizar la consistencia de los datos al nivel al que se mantienen las unidades de cambio atómico para dichos datos (es decir, al nivel de "artículo" para un DBFS basado en artículos). Además, para los DBFS que se ejecutan por defecto en un modo de compromiso relajado, la durabilidad de las transacciones que se han confirmado justo antes de un apagado anómalo no se garantiza y se ha de responder por ella y corregirse.

10 Además, a pesar de que el usuario final de tipo profesional/consumidor se beneficiará en gran medida de la automatización del mantenimiento y la recuperación de DBFS, los gestores de bases de datos y los de mayor experiencia en bases de datos también se beneficiarán de una solución técnica para un mantenimiento y reparación general de bases de datos. En la técnica es corriente que los administradores de bases de datos usen herramientas de bases de datos (por ejemplo, el asesor de ajuste de base de datos que se proporciona con SQL Server 2000), pero estas herramientas no abordan directamente la fiabilidad sino que, en su lugar proporcionan un medio mediante el cual se administran y se gestionan las copias de seguridad de la base de datos - y no de una forma mayormente automatizada, sino que en su lugar requiere una implicación sustancial por parte de los administradores de bases de datos, en particular cuando no se encuentran disponibles copias de seguridad de base de datos o surgen otras cuestiones de reparación. Por lo tanto, una solución automatizada para abordar la fiabilidad de las bases de datos también sería beneficiosa para los administradores de bases de datos y otros usuarios de bases de datos con experiencia. La presente invención proporciona simplemente una solución de este tipo.

15 *Bullock et al.*: "*Recovery data pages after partial page writes*", Boletín de divulgación técnica de IBM, vol. 34, no 3, agosto de 1991, páginas 69 a 83, describe que un área de gran interés para los sistemas de gestión de bases de datos relacionales es la de la provisión de la recuperación con conexión de datos, tal como después de fallos de E/S o de alimentación eléctrica. Se han adoptado diversos enfoques generales para este problema. Un enfoque de este tipo prevé copias redundantes de los datos en el caso de que sea necesario que se reconstruya la base de datos, al igual que en la técnica de doble paginación (*shadow paging*). Otro enfoque general comporta mantener un registro de recuperación de base de datos que contiene cambios en registros de base de datos. La información registrada en el registro de recuperación de base de datos aseguró que los cambios de las transacciones confirmadas se incorporaron en el estado de la base de datos durante el reinicio de sistema a continuación de un fallo de sistema. Las escrituras incompletas en archivos de base de datos de usuario se detectan y la página de datos se repara en la medida de lo posible, permitiendo en la mayor parte de los casos que continúe el reinicio de sistema. Se pueden aplicar entradas de registro a estas páginas de datos, facilitando la recuperación de datos de usuario que aún se encuentran en el registro de recuperación de base de datos. Una vez que el registro de recuperación de base de datos se ha recuperado completamente, se inicia una segunda fase de recuperación, en la que se usan archivos de índice para reparar adicionalmente estas páginas de datos, facilitando la recuperación de los datos de usuario para los cuales un índice contiene datos redundantes.

20 El documento US A 4 945 474 se refiere a la restauración de una base de datos después de un error de E/S empleando protocolos de registro de escritura anticipada. Las entradas de registro se escriben durante el procesamiento normal. El registro de recuperación se recorre durante REDO (rehacer), asegurando la compleción de las operaciones registradas. La recuperación frente a escrituras de registro incompletas o fallidas detectadas se efectúa y las transacciones no confirmadas se deshacen. Los archivos con errores de E/S se detectan y se etiquetan, evitando que subsiguientes operaciones de REINICIO accedan a los archivos. El sistema prevé adicionalmente una reconstrucción automática de los archivos de índice de errores como parte del procedimiento de REINICIO sin requerir una acción explícita por parte del usuario para invalidar los planes de acceso relacionados con el índice fallido.

25 "*Improved recovery from partial page writes of data pages*" Boletín de divulgación técnica de IBM, IBM Corp. Nueva York, EE. UU., vol. 36, n.º 5, 1 de mayo de 1993, páginas 141 a 143, describe un sistema de gestión de base de datos que usa un protocolo de registro de escritura anticipada para asegurar la capacidad de recuperación después de un bloqueo de sistema. Todos los cambios en la base de datos se escriben en el registro de recuperación antes de que los cambios se escriban en la base de datos. Si se produce un bloqueo de sistema, todos los cambios confirmados se pueden volver a reproducir a partir del registro de recuperación. Por lo general, los cambios se registran al nivel de registro lógico. Cuando una página queda fragmentada con espacio libre, se puede volver necesario realizar una cierta compactación de registros para recopilar el espacio libre. Se hace referencia a la operación como reorganización de páginas. Siempre que una página se introduce por lectura en el grupo de

memorias de almacenamiento intermedio durante la recuperación tras bloqueo, se realiza una validación para determinar si la página solo se escribió parcialmente en disco. Siempre que tiene lugar esto, se aborta una recuperación tras bloqueo y la base de datos se ha de restaurar a partir de la copia de seguridad. El gestor de bases de datos propuesto escribirá una nueva entrada de registro cuando se realiza una reorganización de páginas. Esta entrada de registro contendrá el ID de registro para el registro que se está actualizando y la nueva longitud del registro actualizado o insertado. Esta información se requiere de tal modo que la página se pueda reorganizar con exactamente el mismo resultado que durante el procesamiento normal, de tal modo que los punteros de registro en subsiguientes entradas de registro serán correctos. La entrada de registro de reorganización se ha de escribir antes que las otras entradas de registro para la operación que se está realizando.

El documento WO 01/11486 A se refiere a un sistema de archivos de Internet. De acuerdo con un ejemplo, una aplicación realiza una o más llamadas a un sistema operativo para acceder a un archivo. El sistema operativo incluye rutinas que implementan un sistema de archivos de sistema operativo. Las una o más llamadas se hacen a las rutinas que implementan el sistema de archivos del sistema operativo. En respuesta a las una o más llamadas, se envían una o más instrucciones de base de datos a un servidor de base de datos que gestiona la base de datos. El servidor de base de datos ejecuta las instrucciones de base de datos para recuperar los datos a partir de la base de datos. El archivo se genera a partir de los datos y se proporciona a la aplicación.

Murphy N. *et al.*: "*The design and implementation of the database file system*" obtenido a partir de de <http://citeseer.ist.psu.edu/557151.html> describe el diseño y la implementación de un sistema de archivos de base de datos, un sistema de archivos de nivel de usuario y con estructura basada en bloques que usa la BD Berkeley como su almacén de respaldo. El diseño del sistema de archivos de base de datos se simplifica mediante el uso de potentes formas primitivas de bases de datos tales como transacciones, bloqueo de grano fino, y registro de escritura anticipada. La consistencia de los metadatos y de los datos se garantiza a través del uso de transacciones, mientras que el almacenamiento en memoria caché del sistema de archivos es manejado por la base de datos subyacente. La interfaz de sistema de archivos de base de datos es, en gran medida, consistente con la interfaz de sistema de archivos de POSIX normal, permitiendo que los programas usen el sistema de archivos con un mínimo cambio de código. Además, se construye un servidor de sistema de archivos de red encima del sistema de archivos de base de datos, permitiendo un acceso transparente al sistema de archivos por medio del sistema de archivos de red.

Sumario

El objeto de la presente invención es proporcionar un procedimiento para realizar la recuperación ante desastres de bases de datos mediante el uso de un registro.

Este objeto se soluciona por medio de la materia objeto de las reivindicaciones independientes.

Las reivindicaciones dependientes definen realizaciones preferidas.

Diversas realizaciones de la presente invención están orientadas a un sistema de fiabilidad de datos (DRS, *data reliability system*) para un DBFS, en el que el DRS comprende un marco y un conjunto de directivas para realizar tareas de administración de base de datos (DBA, *database administration*) de forma automática y con poca o ninguna implicación directa por parte de un usuario final (y, por lo tanto, es esencialmente transparente a dicho usuario final). Para varias realizaciones, el marco de DRS implementa mecanismos para notificaciones de errores y eventos de conexión, directivas y algoritmos de manejo de errores/eventos en el DRS. Más en particular, para estas realizaciones, el DRS es un subproceso en segundo plano que está a cargo de mantener y de reparar el DBFS en el segundo plano y, por lo tanto, al nivel más alto, el DRS guarda y mantiene el estado de mantenimiento global del DBFS.

Para diversas realizaciones de la presente invención, el DRS comprende las siguientes características: (1) responder a y corregir corrupciones de datos a un nivel de página para todos los tipos de página; y (2) intentar un segundo nivel de recuperación (reconstrucción o restauración) para: (a) corrupciones de página de índice (agrupado y no agrupado); (b) corrupciones de página de datos; y (c) corrupciones de página en el archivo de registro.

Determinadas realizaciones de la presente invención comprenden adicionalmente una funcionalidad específica para el DRS, incluyendo pero sin limitarse a: (i) manejar casos de reparación/restauración de corrupción de datos; y (ii) (iii) mejorar la fiabilidad y la disponibilidad del sistema; y (iv) mantener una tabla de historial de eventos/errores de DRS para un tercero con experiencia para solucionar problemas de motor de almacenamiento o de base de datos si es necesario.

Breve descripción de los dibujos

El sumario anterior, así como la siguiente descripción detallada de realizaciones preferidas, se entiende mejor cuando se lee junto con los dibujos adjuntos. Para el fin de ilustrar la invención, en los dibujos se muestran unas construcciones a modo de ejemplo de la invención; no obstante, la invención no se limita a los procedimientos y medios específicos que se divulgan. En los dibujos:

la figura 1 es un diagrama de bloques que representa un sistema informático en el que se pueden incorporar aspectos de la presente invención;

la figura 2 es un diagrama de bloques que ilustra la estructura del sistema de fiabilidad de datos (DRS) en un sistema de archivos de base de datos (DBFS) representativo de varias realizaciones de la presente invención;

la figura 3A es un diagrama de bloques que ilustra el acoplamiento de almacenes de datos por medio de un FPM de base de datos a un DBFS;

la figura 3B es un diagrama de flujo que ilustra el proceso mediante el cual se detectan y se corrigen páginas corrompidas durante una operación de acoplamiento de base de datos para varias realizaciones de la presente invención;

la figura 4A es un diagrama de flujo que ilustra la utilización de un DRS en un contexto de inicio/recuperación tras bloqueo de acuerdo con diversas realizaciones de la presente invención;

la figura 4B es una tabla que ilustra los errores que son manejados por el DRS durante el acoplamiento de base de datos;

la figura 5A es un diagrama de flujo que ilustra la utilización de un DRS en unas operaciones con conexión en las que el motor de consulta encuentra un error para diversas realizaciones de la presente invención;

la figura 5B es una tabla que ilustra los errores que son manejados por el DRS durante operaciones con conexión;

la figura 6 es un diagrama de flujo que ilustra el DRS realizando una restauración cuando una página existe de hecho en la instantánea más reciente y hay un registro de transacciones válido;

la figura 7 es un diagrama de flujo que ilustra el procedimiento para poner de nuevo con conexión una base de datos para determinadas realizaciones de DRS de la presente invención en las que la base de datos no se puede recuperar y no hay copia de seguridad utilizable alguna; y

la figura 8 es un diagrama de flujo que ilustra el procedimiento para que el DRS intente recuperar una página de datos de acuerdo con un aspecto de la presente invención.

Descripción detallada

La materia objeto se describe con especificidad para satisfacer requisitos legales. No obstante, la propia descripción no tiene por objeto limitar el ámbito de la presente patente. En su lugar, los inventores de la presente invención han contemplado que la materia objeto que se reivindica también se podría incorporar de otras formas, para incluir diferentes etapas o combinaciones de etapas similares a las que se describen en el presente documento, junto con otras tecnologías presentes o futuras. Además, a pesar de que la expresión "etapa" se puede usar en el presente documento para indicar diferentes elementos de los procedimientos empleados, la expresión no se debería interpretar como que implique orden particular alguno de entre, o entre, diversas etapas que se divulgan en el presente documento a menos que, y excepto cuando, el orden de las etapas individuales se describa de forma explícita.

El sumario anterior proporciona una visión de conjunto de las características de la invención. Se da a continuación una descripción detallada de una realización de la invención. Para diversas realizaciones que se describen en lo sucesivo, las características de la presente invención se describen tal como se implementan en el sistema de base de datos SQL SERVER de MICROSOFT (al que se hace referencia a veces en el presente documento simplemente como "SQL") en solitario o incorporado en el sistema de archivos de WinFS de MICROSOFT para el sistema operativo de ordenadores personales de próxima generación (al que se hace referencia por lo general como "Windows Longhorn" o "Longhorn" para abreviar), siendo esto último la materia objeto principal de muchas de las solicitudes de patente a las que se ha remitido en lo que antecede en el presente documento. Tal como se ha mencionado en lo que antecede, SQL SERVER incorpora el Tiempo de Ejecución de Lenguaje Común (CLR, *Common Language Runtime*) .NET de MICROSOFT para posibilitar que el código gestionado se escriba y se ejecute para operar en el almacén de datos de una base de datos de SQL SERVER. A pesar de que la realización que se describe en lo sucesivo opera en este contexto, se entiende que la presente invención no se limita, en modo alguno, a la implementación en el producto SQL SERVER. En su lugar, la presente invención se puede implementar en cualquier sistema de base de datos que soporte la ejecución de código de programación orientado a objetos para operar en un almacén de base de datos, tal como sistemas de base de datos orientados a objetos y sistemas de bases de datos relacionales con extensiones de tipo objeto-relacional. En consecuencia, se entiende que la presente invención no se limita a la realización particular que se describe en lo sucesivo, sino que tiene por objeto cubrir todas las modificaciones que se encuentran dentro del ámbito de la invención según se define mediante las reivindicaciones adjuntas.

Entorno informático

Numerosas realizaciones de la presente invención se pueden ejecutar en un ordenador. La figura 1 y el siguiente análisis tienen por objeto proporcionar una breve descripción general de un entorno informático adecuado en el que se puede implementar la invención. A pesar de que no se requiere, la invención se describirá en el contexto general de las instrucciones ejecutables por ordenador, tales como módulos de programa, que están siendo ejecutados por un ordenador, tal como una estación de trabajo de cliente o un servidor. En general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos y similares que realizan tareas particulares o implementan tipos de datos abstractos particulares. Además, los expertos en la materia apreciarán que la invención se puede poner en práctica con otras configuraciones de sistema informático, incluyendo dispositivos de

mano, sistemas de múltiples procesadores, electrónica de consumo basada en microprocesadores o programable, PC de red, miniordenadores, ordenadores de gran sistema y similares. La invención también se puede poner en práctica en entornos informáticos distribuidos en los que se realizan tareas por medio de dispositivos de procesamiento remoto que están vinculados a través de una red de comunicaciones. En un entorno informático distribuido, los módulos de programa pueden estar ubicados en dispositivos de almacenamiento en memoria tanto locales como remotos.

Tal como se muestra en la figura 1, un sistema informático de propósito general a modo de ejemplo incluye un ordenador personal 20 convencional o similares, incluyendo una unidad de procesamiento 21, una memoria de sistema 22 y un bus de sistema 23 que acopla diversos componentes de sistema, incluyendo la memoria de sistema, con la unidad de procesamiento 21. El bus de sistema 23 puede ser cualquiera de varios tipos de estructuras de bus, incluyendo un bus de memoria o controlador de memoria, un bus de periféricos y un bus local, usando cualquiera de una diversidad de arquitecturas de bus. La memoria de sistema incluye una memoria de solo lectura (ROM, *read only memory*) 24 y una memoria de acceso aleatorio (RAM, *random access memory*) 25. Un sistema básico de entrada/salida 26 (BIOS, *basic input/output system*), que contiene las rutinas básicas que ayudan a transferir información entre los elementos dentro del ordenador personal 20, tal como durante el inicio, se almacena en la ROM 24. El ordenador personal 20 puede incluir adicionalmente una unidad de disco duro 27 para leer de y escribir en un disco duro, que no se muestra, una unidad de disco magnético 28 para leer de o escribir en un disco magnético extraíble 29 y una unidad de disco óptico 30 para leer de o escribir en un disco óptico extraíble 31 tal como un CD ROM u otros medios ópticos. La unidad de disco duro 27, la unidad de disco magnético 28 y la unidad de disco óptico 30 están conectadas con el bus de sistema 23 por medio de una interfaz de unidad de disco duro 32, una interfaz de unidad de disco magnético 33, y una interfaz de unidad óptica 34, respectivamente. Las unidades y sus medios legibles por ordenador asociados proporcionan un almacenamiento no volátil de instrucciones legibles por ordenador, estructuras de datos, módulos de programa y otros datos para el ordenador personal 20. A pesar de que el entorno a modo de ejemplo que se describe en el presente documento emplea un disco duro, un disco magnético extraíble 29 y un disco óptico extraíble 31, debería ser apreciado por los expertos en la materia que, en el entorno operativo a modo de ejemplo, también se pueden usar otros tipos de medios legibles por ordenador que pueden almacenar datos a los que puede acceder un ordenador, tales como casetes magnéticos, tarjetas de memoria flash, discos de vídeo digital, cartuchos de Bernoulli, memorias de acceso aleatorio (RAM), memorias de solo lectura (ROM) y similares.

Un número de módulos de programa se puede almacenar en el disco duro, el disco magnético 29, el disco óptico 31, la ROM 24 o la RAM 25, incluyendo un sistema operativo 35, uno o más programas de aplicación 36, otros módulos de programa 37 y datos de programa 38. Un usuario puede introducir instrucciones e información en el ordenador personal 20 a través de dispositivos de entrada tales como un teclado 40 y un dispositivo señalador 42. Otros dispositivos de entrada (que no se muestran) pueden incluir un micrófono, una palanca de mando, un controlador de juegos, una antena parabólica, un escáner o similares. Estos y otros dispositivos de entrada están a menudo conectados con la unidad de procesamiento 21 a través de una interfaz de puerto serie 46 que está acoplada con el bus de sistema, pero se puede conectar por medio de otras interfaces, tales como un puerto paralelo, un puerto de juegos o un bus serie universal (USB, *universal serial bus*). Un monitor 47 u otro tipo de dispositivo de visualización también está conectado con el bus de sistema 23 por medio de una interfaz, tal como un adaptador de vídeo 48. Además del monitor 47, los ordenadores personales por lo general incluyen otros dispositivos de salida periféricos (que no se muestran), tales como altavoces e impresoras. El sistema a modo de ejemplo de la figura 1 también incluye un adaptador de equipo central 55, un bus de interfaz de sistema para ordenadores pequeños (SCSI, *Small Computer System Interface*) 56 y un dispositivo de almacenamiento externo 62 que está conectado con el bus de SCSI 56.

El ordenador personal 20 puede operar en un entorno en red usando conexiones lógicas con uno o más ordenadores remotos, tal como un ordenador remoto 49. El ordenador remoto 49 puede ser otro ordenador personal, un servidor, un encaminador, un PC de red, un dispositivo del mismo nivel u otro nodo de red común, y por lo general incluye muchos o la totalidad de los elementos que se han descrito en lo que antecede en relación con el ordenador personal 20, a pesar de que, en la figura 1, solo se ha ilustrado un dispositivo de almacenamiento en memoria 50. Las conexiones lógicas que se muestran en la figura 1 incluyen una red de área local (LAN, *local area network*) 51 y una red de área extensa (WAN, *wide area network*) 52. Tales entornos de red están generalizados en oficinas, redes informáticas a nivel de empresa, intranet e Internet.

Cuando se usa en un entorno de red de LAN, el ordenador personal 20 está conectado con la LAN 51 a través de un adaptador o interfaz de red 53. Cuando se usa en un entorno de red de WAN, el ordenador personal 20 por lo general incluye un módem 54 u otros medios para establecer comunicaciones a través de la red de área extensa 52, tal como Internet. El módem 54, el cual puede ser interno o externo, está conectado con el bus de sistema 23 por medio de la interfaz de puerto serie 46. En un entorno en red, los módulos de programa que se muestran en relación con el ordenador personal 20, o porciones de los mismos, se pueden almacenar en el dispositivo de almacenamiento en memoria remoto. Se apreciará que las conexiones de red que se muestran son a modo de ejemplo y que se pueden usar otros medios de establecimiento de un enlace de comunicaciones entre los ordenadores.

A pesar de que se concibe que numerosas realizaciones de la presente invención son particularmente adecuadas para los sistemas computarizados, nada en el presente documento tiene por objeto limitar la invención a tales realizaciones. Por el contrario, tal como se usa en el presente documento, la expresión "sistema informático" tiene por objeto abarcar cualquiera de, y todos, los dispositivos capaces de almacenar y procesar información y/o capaces de usar la información almacenada para controlar el comportamiento o la ejecución del propio dispositivo, con independencia de si tales dispositivos son de naturaleza electrónica, mecánica, lógica o virtual.

Visión de conjunto del sistema de fiabilidad de datos (DRS)

Para varias realizaciones de la presente invención, el sistema de fiabilidad de datos (DRS) es un subproceso que mantiene y repara la base de datos en el segundo plano y, por lo tanto, guarda el estado de mantenimiento general del sistema de archivos de base de datos (DBFS). La figura 2 es un diagrama de bloques que ilustra la estructura del DRS en el DBFS. En la figura, un sistema operativo 202 que proporciona servicios a nivel de sistema operativo a una pluralidad de aplicaciones 212, 214 y 216, comprende un DBFS 222 que está lógicamente acoplado con un almacén de datos persistente 232. El sistema operativo 202 comprende además un DRS 242 que es invocado 244 por medio del DBFS 222 siempre que se descubre un error de página 240 de entre una pluralidad de páginas 234, 236 y 238 en el almacén de datos persistente 232 y, entonces, el DRS 242 realiza operaciones de reparación en respuesta al error de página 240.

Para diversas realizaciones de la presente invención, el DRS puede comprender las siguientes características: (1) responder a y corregir corrupciones de datos a un nivel de página para todos los tipos de página; y (2) intentar un segundo nivel de recuperación (reconstrucción o restauración) para: (a) corrupciones de página de índice (agrupado y no agrupado); (b) corrupciones de página de datos; y (c) corrupciones de página en el archivo de registro. Determinadas realizaciones de la presente invención comprenden adicionalmente una funcionalidad específica para el DRS, incluyendo pero sin limitarse a: (i) manejar casos de reparación/restauración de corrupción de datos; (ii) mejorar la fiabilidad y la disponibilidad del sistema; y (iii) mantener una tabla de historial de eventos/errores de DRS para un tercero con experiencia para solucionar problemas de motor de almacenamiento o de base de datos si es necesario.

Determinadas realizaciones de la presente invención prevén que el DRS sea extensible de tal modo que las directivas de recuperación y los mecanismos de detección se puedan actualizar después de que se haya liberado un DBFS. Varias realizaciones están orientadas a un DRS que ejecuta reparaciones al tiempo que se mantiene con conexión la base de datos de DBFS. Aún otras realizaciones están orientadas a la ejecución con pleno acceso al almacén de DBFS (es decir, privilegios de administrador de sistema). Aún otras realizaciones tendrán la capacidad de detectar y de reaccionar frente a fallos en tiempo real. Para varias realizaciones, las reparaciones de DRS serán transaccionales al nivel al que se mantienen las unidades de cambio para dichos datos (es decir, al nivel de "artículo" para un DBFS basado en artículos). Por último, para diversas realizaciones, las reparaciones o bien recuperarán completamente un artículo o bien este dará marcha atrás a sus cambios, y el DRS tendrá la capacidad de continuar la labor de recuperación/restauración incluso si tiene lugar un reinicio cuando el proceso aún no ha concluido.

Para varias realizaciones de la presente invención, el DRS se abonará a eventos de SQL de tal modo que, si SQL desencadena un evento general, el DRS puede interceptar el mismo y reaccionar (incluyendo, sin limitación, los eventos 823/824). Además, otro aspecto de la presente invención es para que se modifique el motor de base de datos para enviar eventos específicos de DRS para las condiciones de error que el DRS va a manejar de forma específica.

Para diversas realizaciones de la presente invención, las corrupciones se detectarán siempre que el DBFS lee o escribe páginas a partir del disco, caso en el cual, entonces SQL generará uno de una multitud de errores dependiendo de qué tipo de corrupción es esta y también desencadenará eventos de DRS específicos para notificar al mismo las condiciones de error específicas. El DRS recibirá esos errores y los colocará en una cola entrante para su procesamiento.

Para varias realizaciones de la presente invención, determinar si una página está corrompida se logra por diversos medios incluyendo, sin limitación, (a) examinar la suma de comprobación para una página y, si la suma de comprobación no es válida, la página se considera corrompida o (b) mediante el examen del número de serie de registro (LSN, *log serial number*) para ver si este se encuentra más allá del final del archivo de registro (en el que un LSN es un número entero que se incrementa con cada transacción de tal modo que, si la última transacción en el registro fuera el LSN 432 y se hallara una página con un LSN más grande, entonces ha de haber tenido lugar un error de escritura no en orden. A este respecto, hay cuatro tipos principales de corrupciones de página que pueden efectuar la operación de un DBFS (además de otras fuentes tales como defectos, etc.), y estos cuatro tipos incluyen páginas rasgadas, decaimiento de medios, fallo de soporte físico, y escrituras no en orden. Las páginas rasgadas tienen lugar cuando una página de datos no se escribe correctamente de forma atómica y, por lo tanto, cualquier parte de la página se puede corromper debido a que, durante una escritura, solo algunos de los sectores de una página alcanzan con éxito el disco antes del evento de fallo, por ejemplo, un fallo de alimentación eléctrica o un fallo de escritura de sector. El decaimiento de medios tiene lugar cuando se han corrompido unos bits de páginas de datos por el decaimiento físico de medios. Un fallo de soporte físico podría aparecer por una diversidad de razones

relacionadas con el bus, el controlador, o el dispositivo de disco duro. En lo que respecta a la escritura no en orden, estos errores se originan en el hecho de que las unidades de IDE no pueden garantizar el orden de las escrituras en el disco, en concreto la unidad de IDE tiene el almacenamiento en memoria caché de escritura habilitado (activado) y, por lo tanto, es posible que las escrituras en el almacén de datos puedan tener lugar no en orden. Si tiene lugar una serie parcial de escrituras no en orden pero son interrumpidas por un fallo de alimentación eléctrica, por ejemplo, entonces pueden tener lugar varios errores, tales como que la página de datos se escriba en disco antes de que se escriba la entrada de registro asociada, por ejemplo. A pesar de que los errores no en orden se pueden detectar mediante la comprobación de los números de secuencia de registro (LSN, *log sequence numbers*) en páginas de datos, no hay forma sencilla alguna de hacer esto, con la excepción de leer cada página.

10 **Clases de página**

Para los fines de la presente invención, todas las páginas se clasifican de acuerdo con una de las siguientes clases de páginas:

- 5 • Páginas de datos: se considera que una página de datos es cualquier página que tenga datos de usuario en la misma, lo que incluye páginas de tipo hoja de índice agrupado.
- 15 • Páginas de índice: estas páginas contienen solo información de índice, y estas incluyen tanto páginas de índice no agrupadas así como páginas no de tipo hoja de un índice agrupado.
- 20 • Páginas de sistema: estas páginas incluyen las páginas de GAM, de SGAM y de Arranque, y el DRS puede intentar una restauración a nivel de página sobre estas páginas incluso a pesar de que el DRS puede no tener soporte de restauración específico alguno para estas páginas; en cualquier evento, si la restauración a nivel de página falla, entonces el DRS intenta una reparación de emergencia (lo que se analiza posteriormente en el presente documento).
- Páginas no recuperables: la página o páginas de PFS (*Page Free Space*, espacio libre de página) a partir de las cinco tablas de sistema (Sysrowsetcolumns, Sysrowsets, Sysallocunits, Syslobtcolumns, Syslobts) comprenden esta clase de páginas, y el DRS restauraría la totalidad de la base de datos en este caso.
- 25 • Páginas de registro: estas son unas páginas que pertenecen al registro de transacciones, y el DRS intentará una reparación de emergencia cuando estas están corrompidas (lo que se analiza posteriormente en el presente documento).

Categorías de corrupción

30 Para varias realizaciones de la presente invención, el DRS está diseñado para resolver tres categorías diferenciadas de corrupciones de página de datos definidas por cuándo se detecta la corrupción, comprendiendo las tres categorías: (1) durante el acoplamiento de base de datos; (2) durante operaciones con conexión normales; y (3) durante la reversión de transacciones.

Detección de corrupción de acoplamiento de base de datos:

35 La figura 3A es un diagrama de bloques que ilustra el acoplamiento de almacenes de datos por medio de un FPM de base de datos a un DBFS. La figura 3B es un diagrama de flujo que ilustra el proceso mediante el cual se detectan y se corrigen páginas corrompidas durante una operación de acoplamiento de base de datos. En general, un gestor de propiedad de archivos de base de datos (DBFPM, *database file property manager*) 302 gestiona el acoplamiento y el desacoplamiento de los almacenes de DBFS 304, 306, y 308. En la etapa 352, cuando un DBFPM 302 acopla un almacén, por ejemplo, el almacén de DBFS 304, SQL ejecuta la recuperación tras bloqueo en la etapa 354 para determinar en la etapa 356 si hay alguna transacción activa 324 en el momento del bloqueo en el registro de transacciones 314 (por ejemplo, cualquier transacción incompleta) y, de ser así, entonces SQL, en la etapa 358, realiza la recuperación tras bloqueo de la base de datos antes de continuar. Durante la recuperación tras bloqueo, normalmente SQL (a) analizará el registro de transacciones después del último punto de comprobación, (b) volverá a realizar cualquier operación en el registro que este no encuentre escrita en disco, y (c) deshará cualquier transacción que no se haya completado. Si no hay error alguno durante la recuperación que se haya descubierto en la etapa 40 360, entonces la base de datos se acopla con éxito en la etapa 362. No obstante, si se descubre un error en la etapa 360 durante la recuperación tras bloqueo, SQL fallará en el acoplamiento de base de datos en la etapa 364.

45 Este escenario de "acoplamiento de base de datos" es importante debido a que este se invoca cada vez que se une un almacén. Se acoplan almacenes cada vez que se inicia un sistema operativo (por ejemplo, el sistema operativo Windows) así como cada vez que unidades externas (por ejemplo, Firewire, USB, etc.) se acoplan o se desacoplan del ordenador. Los escenarios de acoplamiento de base de datos invocan la recuperación tras bloqueo y, por lo tanto, la detección de páginas rasgadas (escrituras incompletas) y, por lo tanto, es deseable que el DRS maneje este caso debido al potencial de que un usuario retire físicamente el soporte físico.

55 No obstante, cuando se usa un DSR de acuerdo con diversas realizaciones de la presente invención, el comportamiento es ligeramente diferente tal como se ilustra por medio del diagrama de flujo de la figura 4A. En

5 primer lugar, en la etapa 402, el DBFPM acopla las bases de datos de DBFS y, en la etapa 404, SQL ejecuta la recuperación tras bloqueo y, en la etapa 408, intenta poner con conexión la base de datos. Si la base de datos se acopla sin error a partir de la recuperación tras bloqueo en la etapa 410, SQL devolverá un éxito y la base de datos se acoplará (con conexión) en la etapa 452. No obstante, si hay errores, el DRS, en la etapa 412, evaluará la base de datos para determinar si la base de datos es transaccionalmente consistente.

10 Una base de datos es inconsistente solo si esta ha experimentado un fallo durante una reversión de transacciones, es decir, un error de deshacer físico o lógico - o si tuvo lugar un error desconocido durante la recuperación tras bloqueo. Si la base de datos es transaccionalmente consistente en la etapa 412, el DRS, en la etapa 414, retardará las reparaciones hasta que la operación de recuperación se ha completado - dicho de otra forma, el DRS declarará el estado de la base de datos como consistente para el acoplamiento de base de datos (un "éxito") y, entonces, espera para comenzar las reparaciones para las corrupciones detectadas después de que la recuperación se haya completado y la base de datos se haya acoplado en la etapa 452 con el fin de evitar todo conflicto con la recuperación tras bloqueo.

15 Una vez que la recuperación se ha completado y el acoplamiento tiene éxito en la etapa 452, en la etapa 478 si hubo reparaciones retardadas, entonces, en la etapa 480, el DRS comenzará las reparaciones y, en la etapa 482, en primer lugar intentará invocar una restauración a nivel de página para la corrupción. Si la restauración a nivel de página falla o no es posible en la etapa 484, entonces, en la etapa 486, el DRS determinará si esto es debido a un fallo de una página de índice y, de ser así, el DRS reconstruirá el índice en la etapa 488. Por otro lado, si el error se encuentra en una página de datos en la etapa 490, entonces el DRS intentará una recuperación de página de datos en la etapa 492. No obstante, si el DRS determina que la base de datos es inconsistente o tiene un error desconocido/no soportado en la etapa 490, inmediatamente el DRS pondrá sin conexión la base de datos (terminando de ese modo la operación de recuperación) y marcará la base de datos como "sospechosa" en la etapa 494 y entonces, en la etapa 496, intentará una reparación de emergencia.

25 La figura 4B es una tabla que ilustra los errores que son manejados por el DRS durante el acoplamiento de base de datos.

Detección de corrupción de operación con conexión:

30 La figura 5A es un diagrama de flujo que ilustra la utilización de un DRS en unas operaciones con conexión en las que el motor de consulta encuentra un error para diversas realizaciones de la presente invención. Este escenario es durante el uso de tiempo de ejecución normal del DBFS, tal como cuando un usuario final está usando el intérprete de comandos del sistema operativo para ejecutar una consulta moderadamente compleja contra el DBFS y, durante la ejecución de esa consulta, el motor de consulta notifica un error. En estas situaciones de operación con conexión en las que tiene lugar un error, el DRS, en la etapa 502, aborta la transacción para intentar arreglar el error. El DRS, en la etapa 504, en primer lugar intentará una restauración a nivel de página. No obstante, si la restauración a nivel de página no se encuentra disponible o falla en la etapa 506, entonces el DRS determinará el tipo de página que está fallando en la etapa 508. En la etapa 510, si este es un fallo de una página de índice, entonces el DRS intentará una recuperación de índice en la etapa 512. Si el fallo es una página de datos en la etapa 514, entonces el DRS intentará una recuperación de página de datos en la etapa 516. Si el fallo es debido a un error de página de registro o de sistema en la etapa 518, el DRS intentará una reparación de emergencia en la etapa 520. Si el fallo es un error no recuperable (5 tablas de sistema, o página de PFS) de vuelta en la etapa 518, entonces el DFS o bien restaurará de forma automática la totalidad de la base de datos en la etapa 522 o bien, como alternativa, solicitará al usuario final que restaure la totalidad de la base de datos.

La figura 5B es una tabla que ilustra los errores que son manejados por el DRS durante operaciones con conexión.

Detección de corrupción de reversión de transacciones:

45 Si un error tiene lugar durante una reversión de transacciones, el DRS pondrá sin conexión la base de datos, la marcará como sospechosa, y reiniciará la base de datos con el fin de invocar la recuperación tras bloqueo. Entonces, el proceso volverá a invocar el procedimiento para la detección de corrupción de recuperación tras bloqueo que se ha divulgado en lo que antecede en el presente documento.

Técnicas de recuperación

50 Tal como es bien conocido y fácilmente apreciado por los expertos en la materia, consistencia transaccional se refiere a visualizar (o facilitar) solo datos confirmados, así como confirmar (escribir en el almacén persistente y registrar la transacción) solo datos correctos. A este respecto, una base de datos es transaccionalmente consistente siempre que se respeten todas las solicitudes de rehacer y de deshacer durante la recuperación tras bloqueo. Por lo tanto, en las páginas de datos y de índice se pueden encontrar presentes muchos tipos de corrupción pero, aún así, pueden ser transaccionalmente consistentes.

55 Diversas realizaciones de la presente invención reconocen que, a menudo, no es deseable realizar con conexión muchas de estas transacciones y, por lo tanto, estas realizaciones intentan realizar todas las reparaciones mientras que la base de datos se encuentra sin conexión.

Cuando tiene lugar un error de SQL, el DRS solo recibirá el id de página y el id de base de datos. A partir de esa información, el DRS interrogará a la página para resolver exactamente qué tipo de página es esta. Estas realizaciones tienen unos mecanismos de recuperación para la corrupción de páginas de datos y de índice en los que el DRS intenta recuperar los artículos de DBFS que se han perdido debido a esa corrupción. No obstante, hay muchos otros tipos de páginas que se pueden perder en la base de datos, incluyendo páginas de GAM, de SGAM, de PFS, de Arranque, y otras. A pesar de que los DBFS no tienen unos mecanismos de copia de seguridad y de recuperación específicos para estos tipos de páginas, el DRS intentará invocar una restauración a nivel de página.

Para una página que se ha corrompido, el DRS intentará una restauración a nivel de página. La restauración a nivel de página será silenciosa, debido a que se puede garantizar al usuario que se perderá todo dato que no se haya confirmado. La misma técnica se aplicará a páginas procedentes de cualquier tabla, y el procedimiento no cambia con independencia del tipo de página o tabla que se está restaurando. Una restauración a nivel de página solo puede tener lugar si la página existe en la instantánea más reciente. Asimismo, se ha de encontrar disponible un registro de transacciones válido. Si la página no se encuentra en la instantánea más reciente, entonces se ha de recuperar la página usando técnicas de restauración de nivel de artículo que son conocidas y apreciadas por los expertos en la materia (tales como una restauración a partir de una operación de copia de seguridad).

Si la página existe de hecho en la instantánea más reciente y hay un registro de transacciones válido, entonces el DRS realizará las siguientes acciones para restaurar la página, tal como se ilustra por medio de la figura 6: (1) en la etapa 602, determinar el identificador de página de la página que se ha corrompido; (2) en la etapa 604, hallar y copiar la página corrompida a partir de la instantánea más reciente del almacén de DBFS; (3) en la etapa 606, aplicar el registro de transacciones en disco a la página mediante la puesta al día de las transacciones que se aplican a esa página; (4) en la etapa 608, aplicar la página restaurada a la base de datos con conexión; y (5) si hay transacciones aplazadas para la página que se está restaurando en la etapa 610, entonces habrá que reiniciar la base de datos en la etapa 612 (de tal modo que la recuperación tras bloqueo se ejecuta y elimina las transacciones aplazadas), de lo contrario, las operaciones continúan en la etapa 614.

En lo que respecta a la instantánea que se ha mencionado en lo que antecede, VSS (*Volume Shadowcopy Service*, Servicio de Instantáneas de Volumen) proporciona una forma de mantener instantáneas de puntos en el tiempo de determinados volúmenes (por ejemplo, volúmenes de NTFS) tal como será apreciado por los expertos en la materia. Las instantáneas de VSS mantienen la instantánea de un volumen mediante el uso de copia en escritura, es decir, siempre que se modifica una página de disco, la imagen previa de esa página se escribe en un área de almacenamiento de instantáneas de la instantánea más reciente. Cuando TimeWarp está habilitado en una máquina - lo que puede ser la opción por defecto para determinados DBFS y sus correspondientes sistemas operativos - las instantáneas se toman a una tasa por defecto de dos veces al día y se mantiene un máximo de 63 instantáneas. Para hacer uso de las imágenes previas que se almacenan en la instantánea, se mantiene un registro desde el momento en el que se tomó la instantánea hasta el momento actual, y el truncamiento de registro tiene lugar solo en el momento de instantánea de TimeWarp de tal modo que haya en todo momento un registro procedente de la última instantánea disponible. En el caso de que haya una corrupción de página y la página se encuentre disponible en la última instantánea, el registro se puede volver a reproducir desde la instantánea hasta el punto actual en esa página para su restauración.

Durante las instantáneas de TimeWarp, se recupera el almacén de DBFS en la instantánea. Esto hace imposible la restauración a nivel de página para las páginas que han sido tocadas por la recuperación de la base de datos de instantáneas. Para sortear este problema, se adopta un punto de vista de SQL Server sobre la base de datos en el volumen de instantáneas y, entonces, se recupera la base de datos - es decir, el DRS tiene un punto de vista no recuperado sobre la base de datos en el volumen de instantáneas y las páginas ya que el punto de vista se puede usar siempre para una restauración a nivel de página.

Para una reparación de sistema, de registro, o de página desconocida - es decir, si tiene lugar una corrupción de registro o si hay fallos que el DRS no puede corregir (por ejemplo, datos o índice), entonces el DRS le presentará al usuario las siguientes opciones: (a) restaurar la totalidad de la base de datos (store); o (b) recuperar la base de datos en el modo de emergencia.

Para reparar la base de datos en el modo de emergencia, en primer lugar, es importante reconocer la nueva capacidad de DBCC de recuperarse de un registro de transacciones corrompido y de situaciones de base de datos no recuperables. Entonces, si la base de datos no se puede recuperar y no hay copia de seguridad utilizable alguna, el siguiente conjunto de acciones, que se ilustra en la figura 7, podrá de nuevo con conexión la base de datos para determinadas realizaciones de DRS de la presente invención: (a) en la etapa 702, establecer la base de datos en el modo de emergencia; (b) en la etapa 704, ejecutar 'DBCC CHECKDB (base de datos, REPAIR_ALLOW_DATA_LOSS)' lo que tiene un significado especial en el modo de emergencia que (i) obliga a que la recuperación de base de datos avance más allá de los errores (obteniendo tantos datos como sea posible a partir del registro pero dejando una base de datos transaccionalmente inconsistente), (ii) descarta los archivos de registro corrompidos y crea unos nuevos, (iii) ejecuta una reparación de base de datos completa para llevar la base de datos a un estado estructuralmente consistente (una operación "atómica" y unidireccional que no se puede revertir o deshacer, y que es la única forma posible de recuperar la base de datos en una situación de ese tipo sin editar de forma manual los archivos; y (c) ahora que la base de datos es físicamente consistente, el DSR ejecuta el CC en la

totalidad del almacén en la etapa 706. La ejecución con éxito de estas etapas debería garantizar que los datos (los artículos en un DBFS basado en artículos) serán consistentes, pero esto puede querer decir que determinadas aplicaciones serán transaccionalmente inconsistentes.

5 Hay dos tipos de páginas de índice: páginas no agrupadas y páginas no de tipo hoja agrupadas. Las páginas de índice con datos (páginas de tipo hoja de índice agrupado) se consideran páginas de datos. Con esto en mente, para los fallos de índices recuperables, el DRS intenta reparar el índice usando una reconstrucción de índice sin conexión (en la que la base de datos se encontrará con conexión durante esta reparación, mientras que solo el índice se encontrará sin conexión). Si falla la reparación, el DRS descarta el índice e intenta volver a crear el índice. Entonces, si también eso falla, el DRS descartará el índice totalmente o, como alternativa, deshabilitará el índice e intentará reconstruirlo de nuevo en un momento posterior (y tal vez hacer esto de forma indefinida hasta tener éxito).

10 Para la recuperación de página de datos, si el DRS recibe uno de los errores manejados y determina que la página es una página de datos, entonces este intentará una recuperación. Esto puede tener lugar durante cualquier estado de operación de SQL Server (acoplamiento de BD, operación con conexión y reversión). Tal como se ilustra en la figura 8, si el DRS detecta un error de página de datos en la etapa 802, entonces, después de intentar una restauración a nivel de página en la etapa 804 que falla en la etapa 806, el DRS: (a) en la etapa 808, determinará el tipo de página corrompida mediante la recepción de un identificador de página de la página corrompida y la determinación del tipo de página tal como se ha analizado previamente (y dada una corrupción de página de datos, el DRS no tiene que abordar otras transacciones que intentan acceder a la página); (b) en la etapa 810, obtendrá el intervalo de Identificadores de artículo perdidos (iniciar una transmisión, ejecutar DBCC CHECKTable (tabla, REPAIR_ALLOW_DATA_LOSS) que devuelve todas las claves de índice a partir de un índice no agrupado que no coinciden con las claves agrupadas, y las claves de índice no agrupado, contienen los Identificadores de artículo de tal modo que el DRS pueda extraer las mismas y escribirlas en tabla interna; y (c) en la etapa 812, pasará la lista de Identificadores de artículo y nombre de tabla a la WCC (en la que la WCC repara esos Identificadores de artículo (y comprueba el resto del almacén) para garantizar que el DBFS es consistente, y confirma una transmisión para permitir que el DRS aborte y vuelva a intentar estas recuperaciones. Si hay un conjunto de páginas corrompidas, entonces el DRS resuelve la totalidad de los Identificadores de artículo a partir de la totalidad de las páginas corrompidas antes de traspasar las mismas para su procesamiento adicional (tal como mediante la restauración de dichas páginas a partir de una copia de seguridad).

20 Debido a que el DRS puede garantizar que, si la restauración a nivel de página funciona, no se perderá dato de usuario alguno, el DRS solo necesita notificar nominalmente al usuario debido a que lo más probable será que el usuario observe solo una respuesta más lenta de la esperada mientras que está teniendo lugar una restauración a nivel de página. En determinadas realizaciones, el usuario no recibirá notificación alguna (debido a que esto llevará, como mucho, segundos); no obstante, para otras realizaciones alternativas, el DRS hará que una entrada de registro de "evento" capture el hecho de que el DRS restauró una página. En el escenario de caso más desfavorable en el que hay una transacción activa que está pendiente contra la página que se va a restaurar, la base de datos tendrá que desacoplarse y reiniciarse (con el fin de ejecutar la recuperación tras bloqueo) y, por lo tanto, se terminará la totalidad de las conexiones con la base de datos. Las aplicaciones se deberían diseñar ya con esta posibilidad en mente; no obstante, esto podría dar lugar a que las aplicaciones mal diseñadas se "cuelguen". Por lo tanto, determinadas realizaciones de la presente invención proporcionan al usuario una notificación de las circunstancias que pueden conducir a simplemente un evento de este tipo con el fin de permitir que el usuario cierre la totalidad de tales aplicaciones en un esfuerzo por evitar esta consecuencia no deseable.

Experiencias de usuario final de muestra

Para ilustrar la operación del DRS, en el presente caso se da a continuación un puñado de situaciones que caracterizan, en general, la experiencia del usuario así como qué está haciendo el DRS sin que lo perciba el usuario:

45 Corrupción de índice:

Abbey está realizando una consulta de WinFS para hallar la totalidad de los documentos que han sido modificados por ella en la última semana. Durante esta consulta, WinFS observa que la misma lleva un poco más tiempo del habitual. De hecho, algunas de sus subsiguientes consultas también son un poco lentas. Entonces, ella observa un pequeño bocadillo sobre su barra de tareas. Este bocadillo notifica que se ha hallado una cuestión de fiabilidad en su máquina y le ruega que tenga paciencia mientras que Windows repara los errores. Después de un corto tiempo, aparece otro bocadillo e informa a Abbey de que el índice se reconstruyó con éxito. Abbey queda satisfecha cuando observa que su ordenador parece estar funcionando mejor ahora.

En segundo plano el DRS detectó y reparó una corrupción de índice. Este puso sin conexión el índice (de ahí el rendimiento reducido), lo reconstruyó y, entonces, puso de nuevo con conexión el índice.

55 Escritura de página rasgada:

Durante una tormenta eléctrica, Toby está escribiendo un ensayo acerca de Adam Smith para unos deberes escolares para casa y guardando con regularidad. Cuando ha completado aproximadamente un 90 % del ensayo, la totalidad del hogar de Toby queda sin alimentación eléctrica. Toby no está trabajando con un ordenador portátil y no

tiene un respaldo de batería. Afortunadamente para Toby, la alimentación eléctrica se presenta de nuevo en aproximadamente 30 minutos. Toby inicia sesión en el ordenador e intenta abrir su ensayo. Se rasca la cabeza debido a que esto parece llevar un poco más tiempo del habitual.

5 En segundo plano, lo que ha tenido lugar es que el artículo de documento de ensayo de Toby se corrompió debido a una escritura de página rasgada por parte de su unidad de disco duro cuando se perdió la alimentación eléctrica. Cuando el DBFS se reinició, el DRS detectó esta corrupción de página de datos e intentó una recuperación de los datos. El DRS intentó de forma automática restaurar la página de datos a partir de la instantánea automática. Debido a que Toby ha estado guardando su trabajo con regularidad, había una copia del ensayo en la instantánea más reciente. Por lo tanto, el DBFS fue capaz de restaurar las páginas corrompidas de forma automática.

10 Descomposición de bits y corrupción de sectores:

A Susan le gusta su cámara digital hasta tal punto que ha tomado más de 5800 fotos de sus hijos en los últimos 2 años. De estas 5800, ella ha guardado más de 3000 fotografías en WinFS en su ordenador. Desafortunadamente para Susan, su unidad de disco duro que contiene estas queridas fotos ha corrompido un pequeño número de sectores en el disco. Estos sectores corrompidos han corrompido ahora diez de las fotografías de Susan. Cuando Susan intenta ver estas fotografías, se le presenta un error procedente del intérprete de comandos del sistema operativo y se le solicita restaurar estas fotografías a partir de una copia de seguridad. Susan sigue las solicitudes, halla su medio de copias de seguridad (una unidad Zip) y, entonces, restaura los artículos a partir del disco.

20 En segundo plano, el DBFS está intentando una restauración a nivel de página. No obstante, debido a que estas fotografías no se han modificado (nunca), estas no se encuentran en la instantánea más reciente y, por lo tanto, para esta realización particular es necesario solicitar a Susan que inserte sus copias de seguridad (solo para esos artículos).

Conclusión

25 Los diversos sistemas, procedimientos y técnicas que se describen en el presente documento se pueden implementar con soporte físico o soporte lógico o, cuando sea apropiado, con una combinación de ambos. Por lo tanto, los procedimientos y aparatos de la presente invención, o determinados aspectos o porciones de los mismos, pueden adoptar la forma de código de programa (es decir, instrucciones) incorporados en medios tangibles, tales como disquetes flexibles, CD-ROM, unidades de disco duro, o cualquier otro medio de almacenamiento legible por máquina, en el que, cuando el código de programa se carga en y es ejecutado por una máquina, tal como un ordenador, la máquina se vuelve un aparato para poner en práctica la invención. En el caso de la ejecución de código de programa en ordenadores programables, el ordenador incluirá, en general, un procesador, un medio de almacenamiento legible por el procesador (incluyendo memoria volátil y no volátil y/o elementos de almacenamiento), al menos un dispositivo de entrada, y al menos un dispositivo de salida. Uno o más programas se implementan, preferentemente, en un lenguaje de programación orientado a objetos o de procedimientos de alto nivel para comunicarse con un sistema informático. No obstante, el programa o programas se pueden implementar en lenguaje máquina o ensamblador, si se desea. En cualquier caso, el lenguaje puede ser un lenguaje compilado o interpretado, y combinarse con implementaciones en soporte físico.

40 Los procedimientos y aparatos de la presente invención también se pueden incorporar en forma de código de programa que se transmite a través de algún medio de transmisión, tal como a través de cableado o hilos eléctricos, a través de fibra óptica, o por medio de cualquier otra forma de transmisión, en la que, cuando el código de programa se recibe y se carga en y es ejecutado por una máquina, tal como una EPROM, una matriz de puertas, un dispositivo de lógica programable (PLD, *programmable logic device*), un ordenador de cliente, un aparato grabador de vídeo o similares, la máquina se vuelve un aparato para poner en práctica la invención. Cuando se implementa en un procesador de propósito general, el código de programa se combina con el procesador para proporcionar un aparato único que funciona para realizar la funcionalidad de indexación de la presente invención.

45 A pesar de que la presente invención se ha descrito en conexión con las realizaciones preferidas de las diversas figuras, se ha de entender que se pueden usar otras realizaciones similares o que se pueden hacer modificaciones y adiciones a la realización descrita para realizar la misma función de la presente invención sin apartarse de la misma. Por ejemplo, a pesar de que se describen realizaciones a modo de ejemplo de la invención en el contexto de los dispositivos digitales que emulan la funcionalidad de los ordenadores personales, un experto en la materia reconocerá que la presente invención no se limita a tales dispositivos digitales, debido a que lo se describe en la presente solicitud puede ser de aplicación a cualquier número de entornos o dispositivos informáticos existentes o emergentes, tales como una consola de juegos, un ordenador de mano, un ordenador portátil, etc., ya sea cableado o inalámbrico, y se puede aplicar a cualquier número de tales dispositivos informáticos conectados por medio de una red de comunicaciones, y que interactúan a través de la red. Además, se debería poner de relieve que, en el presente documento, se contempla una diversidad de plataformas informáticas, incluyendo sistemas operativos de dispositivos de mano y otros sistemas de interfaz de soporte físico/soporte lógico específicos de la aplicación, en concreto a medida que sigue proliferando el número de dispositivos inalámbricos en red. Por lo tanto, la presente invención no se debería limitar a realización singular alguna sino que, en su lugar, se debería interpretar en cuanto a su amplitud y a su ámbito, de acuerdo con las reivindicaciones adjuntas.

60

REIVINDICACIONES

1. Un sistema de fiabilidad de datos (242), DRS, automatizado, para un sistema de archivos de sistema operativo implementado en base de datos, DBFS, estando lógicamente acoplado el DBFS con un almacén (232) de DBFS que comprende una pluralidad de páginas (234, 236, 238), comprendiendo dicho DRS unos medios para:

5 responder a un conjunto de corrupciones de datos e intentar un primer nivel de recuperación para reparar una página corrompida a un nivel de página para todos los tipos de página, en el que, si dicha página corrompida existe en la instantánea más reciente de dicho almacén de DBFS y si se encuentra disponible un registro de transacciones válido, la página corrompida se halla y se copia (606) a partir de dicha instantánea de dicho almacén de DBFS y dicho registro de transacciones se aplica (608) a dicha página corrompida mediante la
10 puesta al día de las transacciones que se aplican a dicha página; e
intentar un segundo nivel de recuperación para corrupciones de página si la página corrompida no se pudo reparar en el primer nivel de recuperación, comprendiendo dicho intento de un segundo nivel de recuperación abordar corrupciones de página de índice, en el que, en el caso de fallos de índices recuperables, se intenta reparar el índice usando una reconstrucción de índice sin conexión, con la base de datos estando con conexión y
15 el índice estando sin conexión.

2. El sistema de la reivindicación 1, en el que dicho intento de un segundo nivel de recuperación comprende abordar corrupciones de página de datos.

3. El sistema de la reivindicación 1, en el que dicho intento de un segundo nivel de recuperación comprende abordar corrupciones de página en un archivo de registro.

20 4. El sistema de la reivindicación 1, en el que dicho DRS funciona como un subproceso en segundo plano.

5. Un procedimiento implementado por ordenador para un sistema de fiabilidad de datos (242), DRS, automatizado, para un sistema de archivos de sistema operativo implementado en base de datos, DBFS, estando lógicamente acoplado el DBFS con un almacén (232) de DBFS que comprende una pluralidad de páginas (234, 236, 238), comprendiendo dicho procedimiento:

25 responder a un conjunto de corrupciones de datos e intentar un primer nivel de recuperación para reparar una página corrompida a un nivel de página para todos los tipos de página; e
intentar un segundo nivel de recuperación para corrupciones de página si la página corrompida no se pudo reparar en el primer nivel de recuperación,

en el que intentar dicho primer nivel de recuperación comprende:

30 si dicha página corrompida existe en la instantánea más reciente de dicho almacén de DBFS y si se encuentra disponible un registro de transacciones válido:
hallar y copiar (606) la página corrompida a partir de dicha instantánea de dicho almacén de DBFS; y
aplicar (608) dicho registro de transacciones a dicha página corrompida mediante la puesta al día de las transacciones que se aplican a dicha página; y

35 en el que intentar dicho segundo nivel de recuperación comprende abordar corrupciones de página de índice, en el que, en el caso de fallos de índices recuperables, se intenta reparar el índice usando una reconstrucción de índice sin conexión, con la base de datos estando con conexión y el índice estando sin conexión.

6. El procedimiento de la reivindicación 5, en el que intentar dicho segundo nivel de recuperación comprende abordar corrupciones de página de datos.

40 7. El procedimiento de la reivindicación 5, en el que intentar dicho segundo nivel de recuperación comprende abordar corrupciones de página en un archivo de registro.

8. El procedimiento de la reivindicación 5, en el que dicho DRS funciona como un subproceso en segundo plano.

9. Un medio legible por ordenador que comprende instrucciones legibles por ordenador para un sistema de fiabilidad de datos (242), DRS, automatizado, para un sistema de archivos de sistema operativo implementado en base de
45 datos, DBFS, estando lógicamente acoplado el DBFS con un almacén (232) de DBFS que comprende una pluralidad de páginas (234, 236, 238), comprendiendo dichas instrucciones legibles por ordenador instrucciones para:

responder a un conjunto de corrupciones de datos e intentar un primer nivel de recuperación para reparar una página corrompida a un nivel de página para todos los tipos de página; e
50 intentar un segundo nivel de recuperación para corrupciones de página si la página corrompida no se pudo reparar en el primer nivel de recuperación;

en el que intentar dicho primer nivel de recuperación comprende:

si dicha página corrompida existe en la instantánea más reciente de dicho almacén de DBFS y si se encuentra disponible un registro de transacciones válido:

hallar y copiar (606) la página corrompida a partir de dicha instantánea de dicho almacén de DBFS; y

5 aplicar (608) dicho registro de transacciones a dicha página corrompida mediante la puesta al día de las transacciones que se aplican a dicha página; y

en el que intentar dicho segundo nivel de recuperación comprende abordar corrupciones de página de índice, en el que, en el caso de fallos de índices recuperables, se intenta reparar el índice usando una reconstrucción de índice sin conexión, con la base de datos estando con conexión y el índice estando sin conexión.

10 10. El medio legible por ordenador de la reivindicación 9, que comprende además instrucciones mediante las cuales intentar dicho segundo nivel de recuperación comprende instrucciones para abordar corrupciones de página de datos.

11. El medio legible por ordenador de la reivindicación 9, que comprende además instrucciones mediante las cuales intentar dicho segundo nivel de recuperación comprende instrucciones para abordar corrupciones de página en un archivo de registro.

15 12. El medio legible por ordenador de la reivindicación 9, que comprende además instrucciones mediante las cuales dicho DRS funciona como un subproceso en segundo plano.

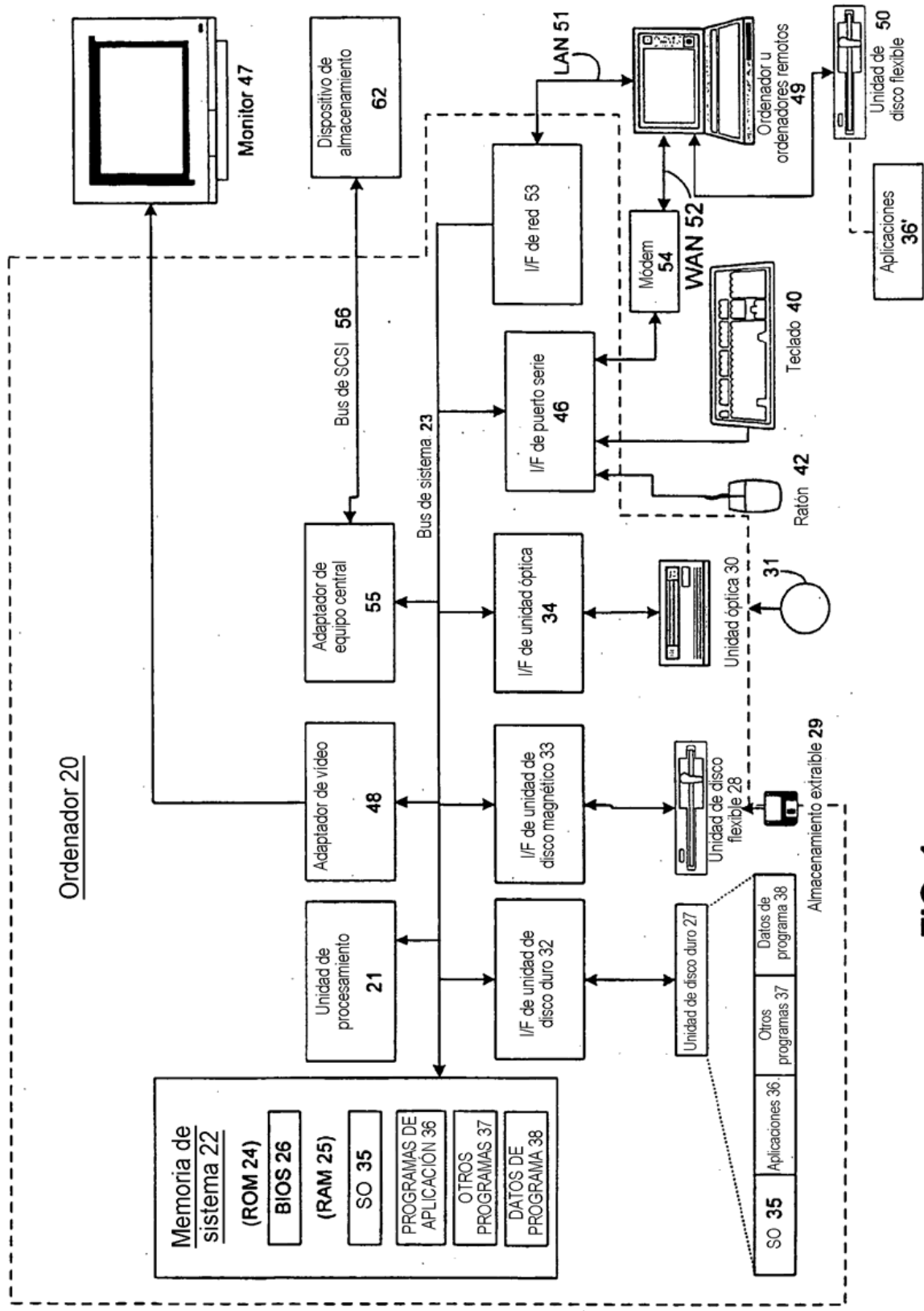


FIG. 1

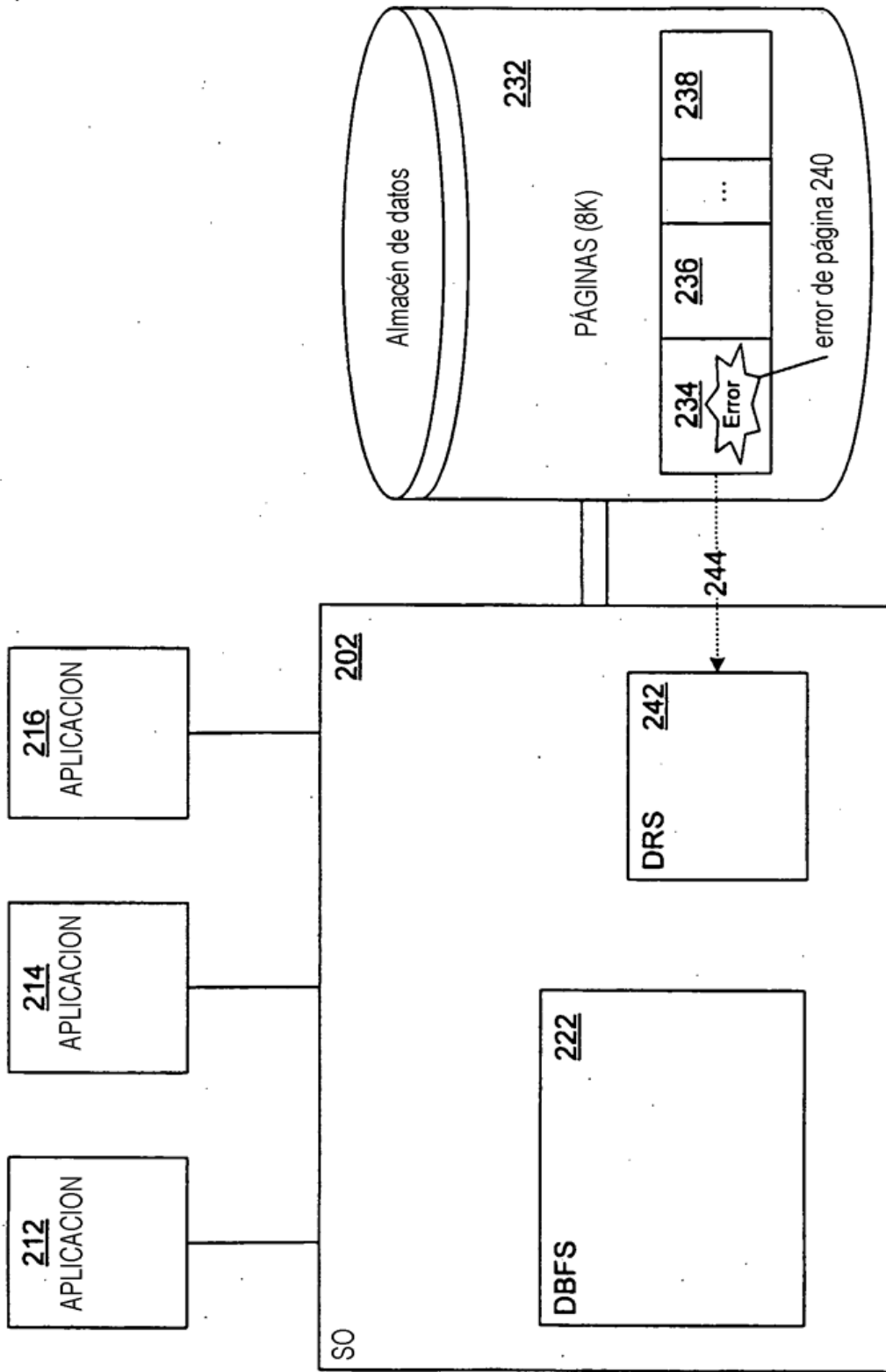


FIG. 2

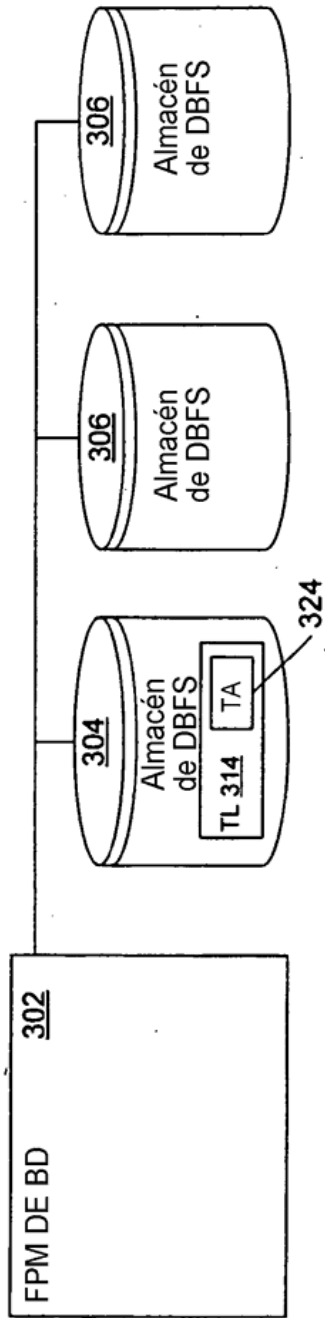


FIG. 3A
TÉCNICA ANTERIOR

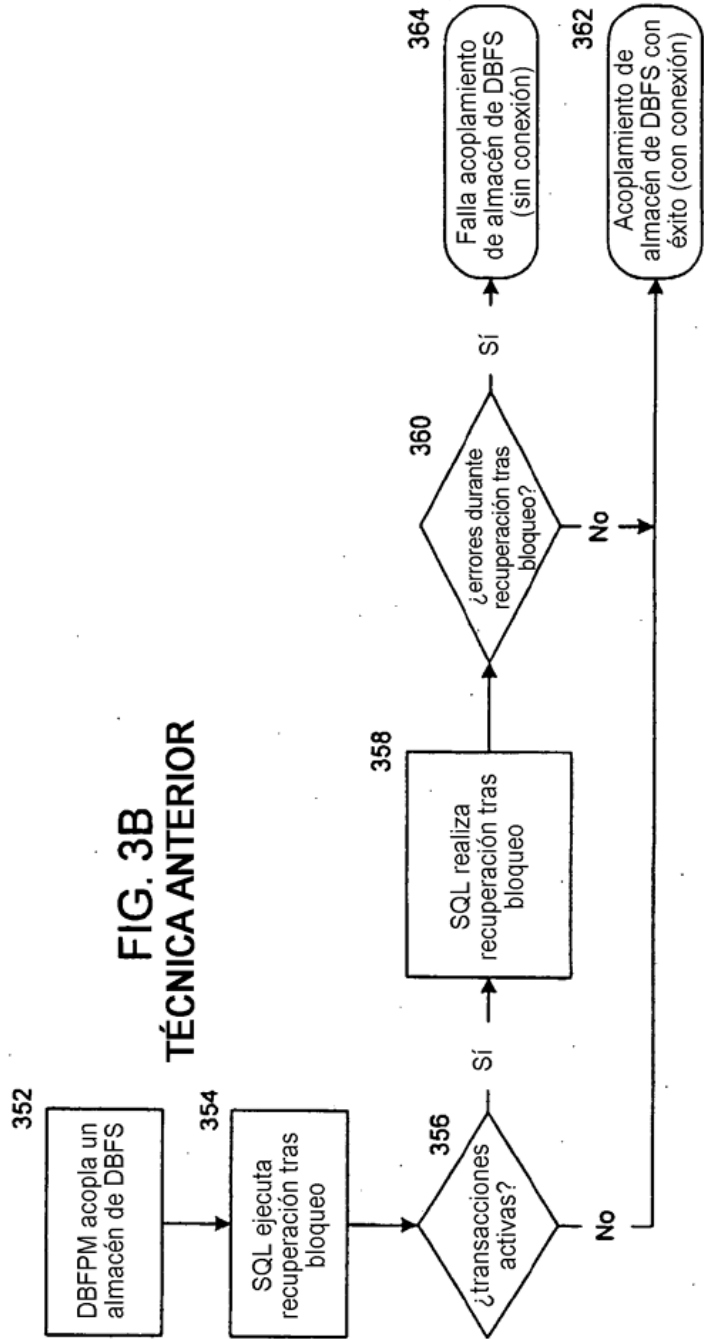


FIG. 3B
TÉCNICA ANTERIOR

FIG. 4A

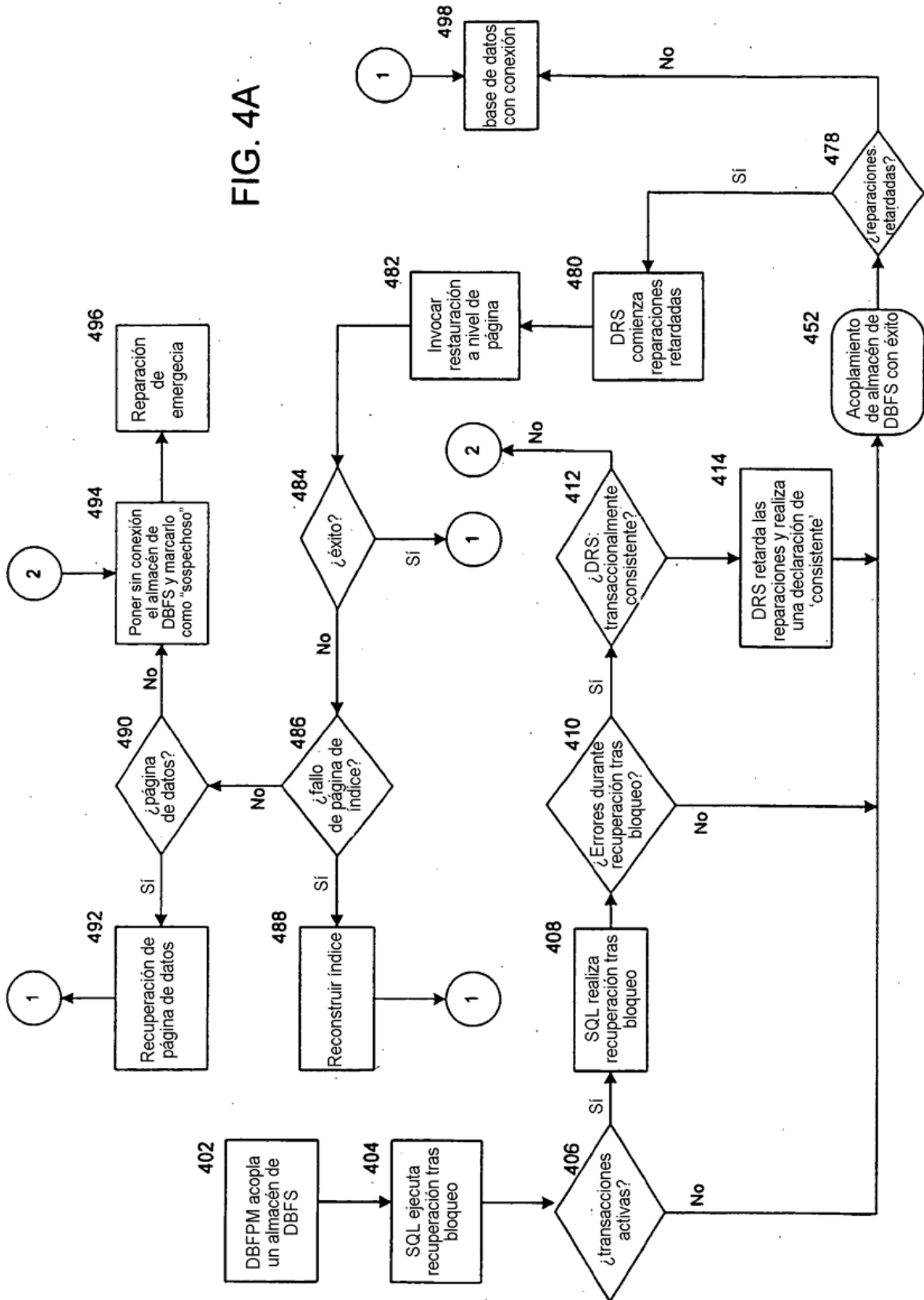


FIG. 4B

MANEJO DE ERRORES DE ACOPLAMIENTO DE BASE DE DATOS

Número de error	Descripción
824	Error de E/S %1!s! detectado por Sistema de Archivos de Windows durante %2!s! de la página %3!s! de dbid %4!s! en el desplazamiento %5!s! en el archivo %6!s!1'.
823	Error de E/S %1!s! detectado por SO durante %2!s! en el desplazamiento %3!s! en el archivo %4!s!1'.
605	Intento de obtención de página lógica %1!s! en base de datos. %2!s! pertenece a la unidad de atribución %3!s!:%4!s!, no a %5!s!:%6!s!

FIG. 5B

MANEJO DE ERRORES DE OPERACIONES CON CONEXIÓN

Número de error	Descripción
824	Error de E/S %1!s! detectado por Sistema de Archivos de Windows durante %2!s! de la página %3!s! de dbid %4!s! en el desplazamiento %5!s! en el archivo %6!s!1'.
823	Error de E/S %1!s! detectado por SO durante %2!s! en el desplazamiento %3!s! en el archivo %4!s!1'.
605	Intento de obtención de página lógica %1!s! en base de datos. %2!s! pertenece a la unidad de atribución %3!s!:%4!s!, no a %5!s!:%6!s!
615	No se pudo hallar ID de tabla de base de datos %1!s!, nombre %2!s!.

FIG. 5A

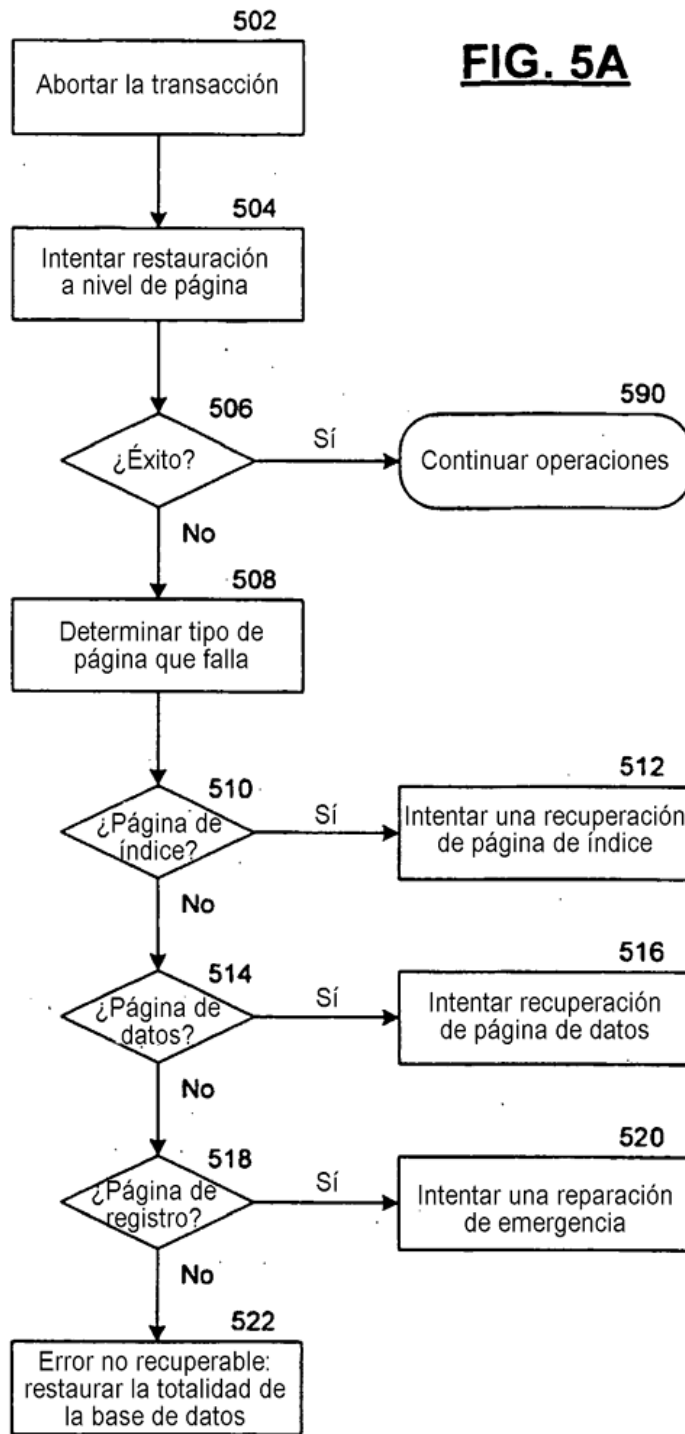


FIG. 6

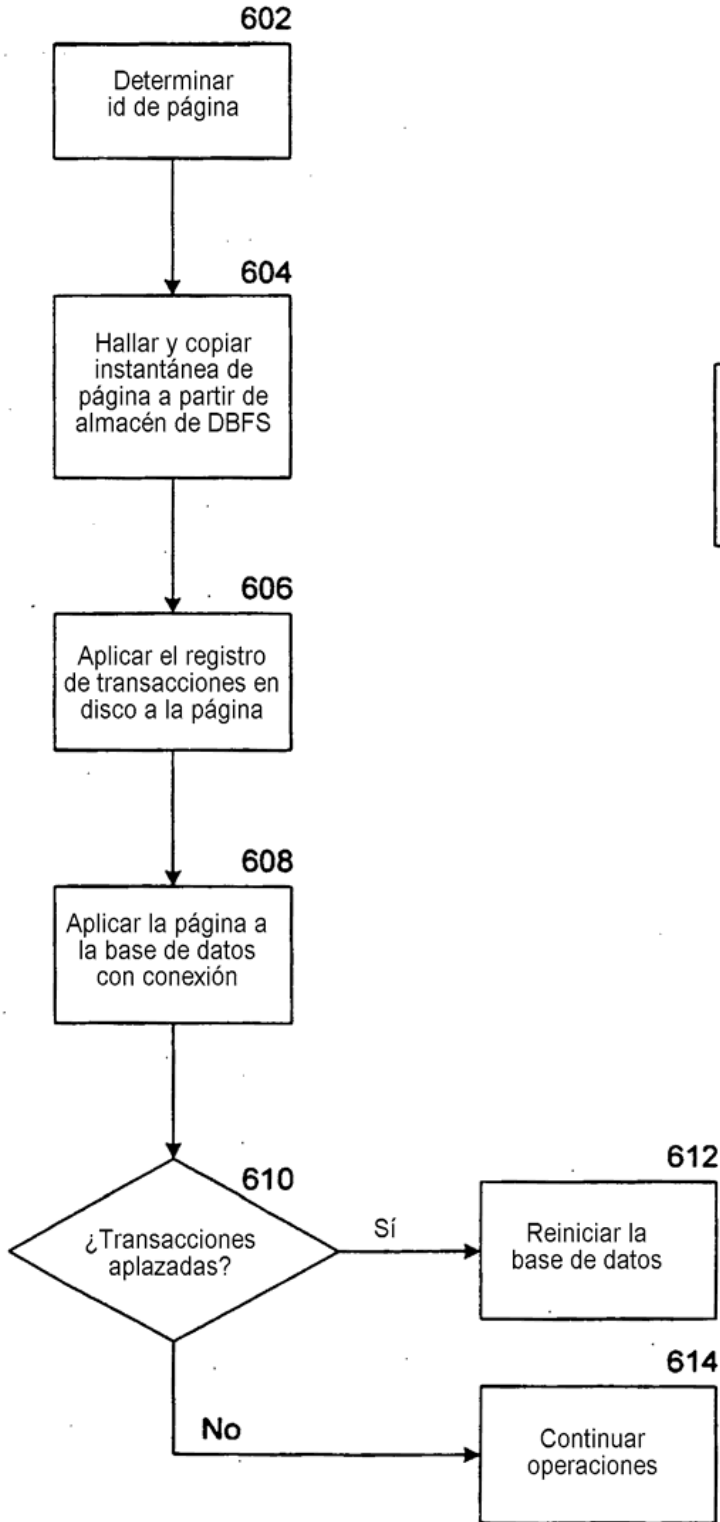


FIG. 7

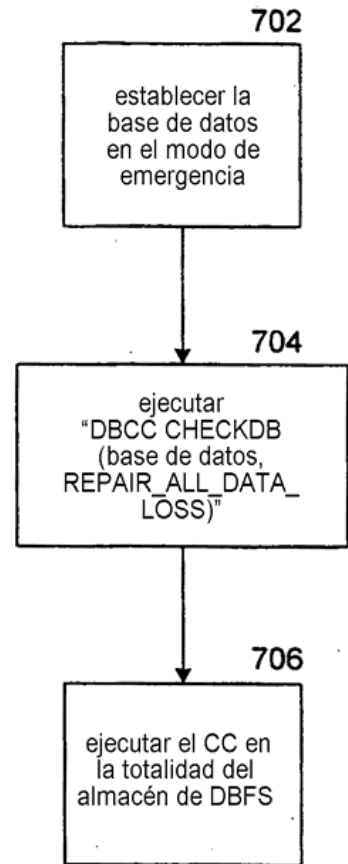


FIG. 8

