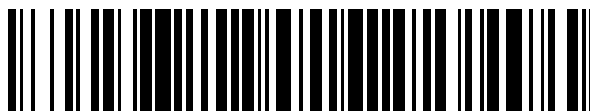


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 380 175**

51 Int. Cl.:
G06F 9/45

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **02709187 .5**

96 Fecha de presentación: **25.01.2002**

97 Número de publicación de la solicitud: **1374046**

97 Fecha de publicación de la solicitud: **02.01.2004**

54 Título: **Sistema y procedimiento de transformación de código objeto**

30 Prioridad:
23.02.2001 US 792551
02.05.2001 US 848384

45 Fecha de publicación de la mención BOPI:
09.05.2012

45 Fecha de la publicación del folleto de la patente:
09.05.2012

73 Titular/es:
Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052, US

72 Inventor/es:
SWETLAND, Brian

74 Agente/Representante:
Carpintero López, Mario

ES 2 380 175 T3

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

DESCRIPCIÓN

Sistema y procedimiento de transformación de código objeto

Antecedentes de la invención**Campo de la invención**

- 5 La presente invención se refiere, en general, al campo de la programación de ordenadores. Más específicamente, la invención se refiere a un sistema y procedimiento para transformar código de programas orientados a objetos.

Descripción de la técnica relacionada

- 10 Java es un lenguaje de programación orientado a objetos que se usa para diseñar programas de ordenador que son independientes de las plataformas. Es decir, el mismo código objeto de Java puede usarse en numerosos sistemas operativos distintos, incluso, por ejemplo, Windows 95, Unix, Solaris y el Sistema Operativo Macintosh. Esta interoperabilidad hace de Java una elección ideal para programar aplicaciones de Internet.

- 15 Una vez que un programa está escrito en código fuente de Java, el compilador de Java genera un código objeto compacto, neutral para las arquitecturas (usualmente mencionado y conocido como código de octetos de Java) que puede ser ejecutado por un intérprete de tiempo de ejecución residente en el ordenador cliente. Este intérprete de tiempo de ejecución se denomina usualmente una "máquina virtual" de Java. La máquina virtual de Java interpreta el código objeto de modo tal que las instrucciones puedan ser ejecutadas por el microprocesador nativo del cliente. Las máquinas virtuales se incluyen en aplicaciones exploradoras de Internet usualmente disponibles, tal como el Navegador™ Netscape o el Explorador™ de Internet de Microsoft.

- 20 Java se generó a partir del popular lenguaje C++ de programación y retuvo muchas características significativas de ese lenguaje. Por ejemplo, Java, como C++, es orientado a objetos. En consecuencia, los programas de Java se desarrollan alrededor de "clases" y "objetos". Estos dos términos no son intercambiables, pero están directamente relacionados entre sí. Una clase puede concebirse como una plantilla o patrón a partir de la cual se hace un objeto. Un objeto puede contener algún número de "campos" en los cuales pueden almacenarse datos (tales como enteros, cadenas de caracteres o referencias a otros objetos). Una clase define un cierto número de acciones ("métodos") que pueden operar sobre el objeto y que son implementadas bien en código de octetos, bien enlazadas a código nativo, o bien son abstractas. Cada objeto es una "instancia" de alguna clase específica, que contiene los campos descritos y que puede ser objeto de operaciones por parte de los métodos descritos.

- 30 Un "fichero de clase" es un objeto binario que describe una única clase: todos los campos, métodos, código objeto de esos métodos, etc. El formato de los ficheros de clase es descrito en detalle por el capítulo cuatro de la obra "La especificación de la máquina virtual de Java" (ISBN 0-201-43294-3). El diseño básico de un fichero de clase se describirá más adelante.

- 35 El documento US 5.966.542 describe un procedimiento y sistema en los cuales un módulo ejecutable, tal como las clases de Java, se almacena en una memoria de sólo lectura. Un cargador estático de clases identifica métodos que contienen referencias simbólicas no resueltas y datos que varían durante la ejecución del módulo. Estos métodos y datos se identifican a fin de situarlos en el espacio de direcciones que reside en una memoria de acceso aleatorio. Al utilizar una memoria de sólo lectura para almacenar clases cargables estáticamente, la memoria de acceso aleatorio se deja disponible para otros usos. A fin de reducir la cantidad de memoria de sólo lectura usada, las constantes de las distintas clases se combinan en un fondo común de constantes, en dos segmentos. En un primer segmento del fondo común de constantes, que contiene a lo sumo 256 constantes, las constantes son objeto de referencia por operandos de 8 bits y, en el segundo segmento, las constantes son objeto de referencia por dos operandos de 8 bits (col. 6, líneas 40 10 a 33)

- 45 Los programadores combinan habitualmente grupos de ficheros de clase ya confeccionados, denominados "bibliotecas de clases", para escribir programas. Por ejemplo, una biblioteca de clases está habitualmente disponible para proporcionar funciones de interfaz gráfica de usuario ("GUI"), tales como rutinas de ventanas, botones, barras deslizables y otras características gráficas.

- Según se ilustra en la **Figura 1**, un fichero 100 de clases ejemplar está compuesto por una cabecera 101, una pluralidad de entradas 102 del fondo común de constantes y uno o más métodos 103 a 105 y / o campos 106, 107. La cabecera 101 contiene datos para identificar un fichero de clase como un fichero de clase (p. ej., la información de revisión). Cada una de las entradas 102 del fondo común de constantes está compuesta, respectivamente, por una parte 111, 121, 131, 141 de cabecera (una "etiqueta" de un único octeto) y una parte 113, 123, 133, 134 de datos de longitud variable. La parte 111, 121, 131, 141 de cabecera indica el tipo de entrada del fondo común de constantes. Según la especificación, las entradas pueden ser del tipo Clase, RefCampo, RefMétodo, RefMétodoInterfaz, Cadena, Entero, Flotante, Largo, Doble, NombreYTipo o Utf8. La parte 113, 123, 133, 143 de datos contiene la información

efectiva del fondo común de constantes asociada a la entrada. Los tipos Utf8, Entero, Flotante, Largo y Doble encapsulan datos constantes sencillos, según lo indican sus nombres. Los otros tipos contienen referencias (por un índice con base) a otras entradas. Por ejemplo, una RefMétodo se refiere a una entrada de Clase y a una entrada de NombreYTipo, que identifican un método específico de una clase específica.

5 Por ejemplo, según se ilustra en la **Figura 2**, cada uno de los ficheros 200, 210 y 220 de clase contiene entradas del fondo común de constantes que se refieren a métodos y campos proporcionados por el fichero 230 de clase. Más específicamente, los ficheros 200, 210 y 220 de clase incluyen, respectivamente, las entradas 203, 213 y 223 del "RefMétodo Pepe" en sus fondos comunes 202, 212 y 222 de constantes, que se refieren al método Pepe 236 de la clase 230. Además, el mismo fichero 230 de clase incluye una entrada "RefMétodo Pepe" en su fondo común 232 de constantes que se refiere al método Pepe 236. De manera similar, cada uno de los fondos comunes de constantes de los ficheros 200, 220 y 230 de clase incluye una entrada "RefCampo Barra" que se refiere al campo Barra 238 en el fichero 230 de clase.

15 En consecuencia, cuando cada uno de los precedentes ficheros de clase se usa en un programa (p. ej., una miniaplicación o aplicación de Java), tres entradas redundantes del fondo común de constantes, referidas al campo Barra 238, se cargan en la memoria (es decir, las entradas RefCampo 240, 241 y 242) y cuatro entradas redundantes del fondo común de constantes, referidas al método Pepe 236, se cargan en la memoria (es decir, las entradas RefMétodo 203, 213, 223, 233). Considerando el hecho de que un programa puede utilizar veintenas de ficheros de clase y que cada fichero de clase puede contener cientos, o incluso miles, de entradas del fondo común de constantes, una significativa cantidad de memoria puede ser consumida por información redundante.

20 En consecuencia, lo que se necesita es un sistema y procedimiento para reducir los requisitos de memoria para programas orientados a objetos.

Resumen

25 Un procedimiento implementado en un ordenador para combinar una pluralidad de ficheros de clase en un lote unificado de ficheros de clase que incluye los métodos y campos de los ficheros de clase originales. El procedimiento incluye:

copiar los métodos de los ficheros de clase originales en el lote unificado de ficheros de clase, generando por ello una entrada de método global dentro del lote unificado de ficheros de clase para cada método dentro de la pluralidad de ficheros de clase;

30 copiar los campos de los ficheros de clase originales en el lote unificado de ficheros de clase, generando por ello una entrada de campo global dentro del lote unificado de ficheros de clase para cada campo dentro de la pluralidad de ficheros de clase; y

35 convertir las entradas locales del fondo común de constantes, de los fondos comunes locales de constantes de cada uno entre la pluralidad de ficheros de clase que hacen referencia a métodos y campos de cada fichero de clase, en entradas globales del fondo común de constantes dentro del lote unificado de ficheros de clase, en donde la conversión comprende eliminar entradas redundantes locales del fondo común de constantes y reemplazar las referencias, que apuntan desde las entradas locales del fondo común de constantes a métodos y campos, por valores de desplazamiento proporcionados en ranuras, a las que se asignan números de ranura basados en desplazamientos desde el extremo superior de las entradas globales del fondo común de constantes, apuntando los valores de desplazamiento desde las ranuras a las entradas globales de métodos y a las entradas de campos de métodos, respectivamente, dentro del lote unificado de ficheros de clase.

Breve descripción de los dibujos

Puede lograrse una mejor comprensión de la presente invención a partir de la siguiente descripción detallada, conjuntamente con los siguientes dibujos, en los cuales:

La FIG. 1 ilustra entradas del fondo común de constantes y otros elementos dentro de un fichero de clase.

45 La FIG. 2 ilustra entradas RefMétodo y RefCampo del fondo común de constantes, referidas a métodos y campos dentro de un fichero de clase.

La FIG. 3 ilustra una vista interna de un dispositivo portal según una realización de la invención. La Fig. 4 ilustra una arquitectura de red que incluye un módulo de conversión de contenido según una realización de la invención.

La FIG. 5 ilustra un lote de ficheros de clase según una realización de la invención.

50 La FIG. 6 ilustra punteros / desplazamientos dentro de un lote.

La FIG. 7 ilustra un proceso para generar un lote según una realización de la invención.

Descripción detallada de realizaciones preferidas

5 En la siguiente descripción, con fines de explicación, numerosos detalles específicos se exponen a fin de proporcionar una comprensión exhaustiva de la presente invención. Será evidente, sin embargo, a alguien experto en la técnica, que la presente invención puede ponerse en práctica sin algunos de estos detalles específicos. En otros casos, estructuras y dispositivos bien conocidos se muestran en forma de diagrama en bloques, para evitar oscurecer los principios subyacentes de la presente invención.

Una arquitectura ejemplar de red

10 Elementos de la presente invención pueden incluirse dentro de una arquitectura con base de cliente-servidor. Según una realización, un servidor portal se comunica con clientes y otros servidores de red por una red (p. ej., Internet). La red por la cual los clientes y servidores transmiten y reciben datos puede estar compuesta por cualquier combinación de canales de comunicación privados (p. ej., alquilados) y / o públicos. Estos pueden incluir, por ejemplo, canales de Señal Digital ("DS") (p. ej., DS-3/T-3, DS-1/T1), canales de Red Óptica Sincrónica ("SONET") (p. ej., OC-3/STS-3), canales de Red Digital de Servicios Integrados ("ISDN"), canales de Línea de Abonado Digital ("DSL"), canales de módem por cable y una gran variedad de canales de comunicación inalámbrica que incluyen canales de difusión
15 satelital y celulares.

Además, pueden usarse diversos protocolos de red para dar soporte a la comunicación por la red, incluso, por ejemplo, la Modalidad de Transferencia Asíncrona ("ATM"), Ethernet y el Anillo con Testigo (al nivel del enlace de datos); así como el Protocolo de Control de Transmisión / Protocolo de Internet ("TCP/IP"), el Intercambio de Paquetes Entre Red
20 Redes ("IPX"), AppleTalk y DECnet (al nivel de red / transporte). Debería observarse, sin embargo, que los principios de la invención no se limitan a ningún canal o protocolo de comunicación específico.

El servidor portal en una realización incluye una base de datos de usuario para almacenar diversos tipos de datos de configuración de usuario y de cuentas. Los usuarios pueden registrarse y conectarse con el servidor portal desde un cliente, especificando un Identificador de usuario y / o contraseña. Según una realización, un usuario se conecta a los
25 servidores mediante una aplicación exploradora, tal como el Navegador™ de Netscape o el Explorador™ de Internet de Microsoft., que se comunica mediante el Protocolo de Transferencia de Hipertexto (en adelante en el presente documento, "HTTP").

En una realización, los usuarios pueden configurar el servidor portal para extraer y gestionar tipos específicos de información. Por ejemplo, un usuario puede configurar el servidor portal para extraer cotizaciones bursátiles actualizadas para un conjunto especificado de valores (p. ej., los que reflejan la cartera del usuario), para recoger el pronóstico meteorológico para el pueblo del usuario y / o para extraer artículos recientes referidos a una franquicia deportiva específica. El servidor portal extraerá entonces la información especificada de otros servidores (p. ej., servidor) en nombre del usuario.
30

Además de la extracción y gestión de información, en una realización, el servidor portal también proporciona servicios de aplicación, tales como correo electrónico, planificación en línea (p. ej., citas, listas de tareas, etc.), mensajería instantánea, gestión de contactos, procesamiento de la palabra y una gran variedad de otros servicios en línea. Los usuarios pueden acceder a estos servicios conectándose al servidor portal con un Identificador válido de usuario y una contraseña. En una realización, el servidor portal genera una página de la Red única y personalizada para cada usuario, que contiene enlaces a toda, o a un subconjunto de, la información y / o de los servicios a los que se ha abonado el usuario.
35
40

Realizaciones de la invención

En una realización, un dispositivo portal almacena y procesa información y / o programas especificados por el usuario, así como información / programas no especificados por el usuario (p. ej., anuncios orientados en base al perfil de los usuarios). La información, o los programas, pueden transmitirse al dispositivo portal a través del cliente y / o directamente mediante difusión inalámbrica. Así, el dispositivo portal en esta realización es una extensión eliminable del servidor portal, almacenando un subconjunto de la información y de los servicios mantenidos por el servidor portal en nombre del usuario. Por ejemplo, un usuario puede configurar el servidor portal para descargar periódicamente la lista de tareas del usuario (u otros datos de planificación) al dispositivo portal (p. ej., cada mañana, cada dos horas, cada vez que el usuario conecta el dispositivo portal con el cliente, etc.). Cuando el usuario abandona la oficina, puede
45 llevarse sencillamente el dispositivo portal consigo y ver su planificación a lo largo del día.

La sincronización de la descarga de información, o de programas, puede depender de la realización específica del dispositivo portal. Por ejemplo, si se usa una realización inalámbrica (descrita más adelante), las descargas pueden tener lugar en cualquier momento cuando el dispositivo portal esté dentro del alcance de transmisión inalámbrica, mientras que si se usa una realización no inalámbrica, las descargas pueden estar limitadas a periodos de tiempo en

que el dispositivo portal está conectado con el servidor portal.

En una realización, el usuario puede personalizar las preferencias y contenido del dispositivo portal, que se descargarán al dispositivo portal desde el servidor portal. Esto puede lograrse, por ejemplo, seleccionando ciertas preferencias / contenido en una página de la Red del servidor portal (p. ej., usando una interfaz de programación en línea según se describe más adelante). Por ejemplo, el usuario puede escoger tener la lista de tareas de cada día descargada en su dispositivo portal y también puede programar el dispositivo (p. ej., mediante el servidor portal) para exhibir continuamente el próximo suceso planificado para el día. Varios otros datos de interfaz de usuario, y datos basados en contenidos, pueden transmitirse al dispositivo portal desde el servidor portal, cumplimentando a la vez los principios subyacentes de la invención.

10 Enlace de Cliente

Una realización del dispositivo portal se comunica con el servidor portal mediante un enlace de comunicación con el cliente. El enlace de comunicación puede establecerse mediante una conexión física de Entrada / Salida con el cliente, tal como una interfaz del Bus Universal en Serie ("USB") o una interfaz de comunicación ("en serie"). Alternativamente, el enlace de comunicación puede ser un enlace inalámbrico, tal como un canal de Entrada / Salida Infrarrojo o un canal de Entrada / Salida de frecuencia de radio ("RF").

En una realización específica, el enlace del cliente se forma usando un canal de comunicación acoplado capacitivamente. Como se conoce en la técnica, un condensador es cualquier elemento dieléctrico emparedado entre dos elementos conductores. En esta realización, uno de los dos elementos conductores está situado dentro del dispositivo portal y el segundo de los dos elementos conductores está situado externamente al dispositivo portal, y está acoplado comunicativamente con un puerto de Entrada / Salida del cliente. Por ejemplo, en una realización, el segundo elemento conductor puede haberse dispuesto dentro del panel del ratón del usuario. Según esta realización, el usuario puede colocar sencillamente el dispositivo portal sobre el panel del ratón para establecer el enlace de comunicación capacitiva con el cliente. Debería observarse, sin embargo, que pueden emplearse otros diversos enlaces del cliente, cumplimentando a la vez los principios subyacentes de la invención.

25 Difusión directa de radio

En una realización, los datos y / o los programas se transmiten al dispositivo portal desde el servidor portal por un enlace de RF. En esta realización, la organización que mantiene el servidor portal y / o que implementa otras características del sistema y procedimiento descritos en el presente documento (en adelante en este documento, la "organización del portal" o "PO") puede alquilar una parte del ancho de banda de transmisión de RF de una o más estaciones de radio. Debería observarse, sin embargo, que pueden usarse varias técnicas de transmisión de RF sin apartarse de los principios subyacentes de la invención.

Realizaciones del dispositivo portal

Como se ilustra en la Figura 3, una realización del dispositivo portal está compuesta generalmente por un microcontrolador 505, una memoria externa 550, un controlador 575 de visor y una batería 560. La memoria externa 550 puede usarse para almacenar programas y / o datos 565 de portal transmitidos al dispositivo portal desde el servidor portal (p. ej., mediante el cliente y / o la estación de radio). En una realización, la memoria externa 550 es memoria no volátil (p. ej., una memoria de sólo lectura programable y eléctricamente borrable ("EEPROM"); una memoria de sólo lectura programable ("PROM"), etc.). Alternativamente, la memoria 550 puede ser una memoria volátil (p. ej., memoria de acceso aleatorio o "RAM"), pero los datos almacenados en la misma pueden mantenerse continuamente mediante la batería 560. La batería 560, en una realización, es una batería celular de tipo moneda (p. ej., del mismo tipo usado en dispositivos electrónicos portátiles tales como calculadoras y relojes). En una realización, cuando la energía de la batería disminuye por debajo de un nivel de umbral, el dispositivo portal 150 notificará al usuario y / o al servidor portal. El servidor portal, en una realización, enviará entonces automáticamente al usuario una nueva batería.

45 El microcontrolador 505 de una realización está compuesto por una unidad central de procesamiento ("CPU") 510, una memoria de sólo lectura ("ROM") 570 y una memoria RAM 540 de borrador. La memoria RAM 570 está adicionalmente compuesta por un módulo intérprete 520 y un módulo 530 de caja de herramientas.

El módulo 530 de caja de herramientas de la memoria ROM 570 contiene un conjunto de rutinas de caja de herramientas para procesar datos, texto y gráficos en el dispositivo portal. Estas rutinas incluyen dibujar texto y gráficos sobre el visor 430 del dispositivo portal, descomprimir datos transmitidos desde el servidor portal, reproducir audio en el dispositivo portal y realizar diversas funciones de entrada / salida y de comunicación (p. ej., transmitir / recibir datos por el enlace del cliente y / o el enlace de RF). Puede incluirse una gran variedad de funciones adicionales del dispositivo portal dentro de la caja 530 de herramientas, cumplimentando a la vez los principios subyacentes de la invención.

En una realización, los microprogramas y los datos 560 del portal se transmiten desde el servidor portal a la memoria

externa 550 del dispositivo portal, mediante una interfaz 570 de comunicación bajo el control de la CPU 510. Pueden emplearse diversas interfaces 570 de comunicación sin apartarse de los principios subyacentes de la invención, incluso, por ejemplo, una interfaz del Bus Universal en Serie ("USB") o una interfaz de comunicación en serie ("en serie"). Los microprogramas en una realización están compuestos por instrucciones compactas interpretadas, conocidas como "códigos de octeto", que son convertidas en código nativo por el módulo intérprete 520 antes de ser ejecutadas por la CPU 510. Una de las ventajas de esta configuración es que, cuando la parte del microcontrolador / CPU del dispositivo portal es renovada (p. ej., por un modelo más rápido y / o menos caro), solamente el módulo intérprete 520 y la caja 530 de herramientas de la memoria ROM necesitan ser reescritos para interpretar los códigos de octeto actualmente existentes para el nuevo microcontrolador / CPU. Además, esta configuración permite a los dispositivos portales 150 con distintas CPU coexistir y ejecutar los mismos microprogramas. Además, la programación de rutinas frecuentemente usadas en el módulo 530 de caja de herramientas de la memoria ROM reduce el tamaño de los microprogramas almacenados en la memoria externa 550, conservando por ello memoria y ancho de banda por el enlace del cliente y / o el enlace 220 de RF. En una realización, pueden desarrollarse nuevos módulos intérpretes 520 y / o rutinas 530 de caja de herramientas para ejecutar los mismos microprogramas en teléfonos celulares, administradores de información personal ("Ptm") o cualquier otro dispositivo con una CPU y memoria.

Una realización de la memoria ROM 570 puede estar compuesta por código interpretado así como código nativo escrito específicamente para la CPU 505 del microcontrolador. Más específicamente, algunas rutinas de caja de herramientas pueden escribirse como código interpretado (según lo indicado por la flecha entre la caja 530 de herramientas y el módulo intérprete 520) para conservar memoria y ancho de banda, por las mismas razones descritas anteriormente con respecto a los microprogramas. Además, en una realización, los datos y microprogramas almacenados en la memoria externa 550 pueden configurarse para prevalecer sobre versiones más antiguas de datos / microprogramas almacenados en la memoria ROM 570 (p. ej., en la caja 530 de herramientas de la memoria ROM).

Como se ha mencionado anteriormente, el dispositivo portal puede comunicarse con el servidor portal usando diversas técnicas de comunicación de RF. Por ejemplo, en una realización específica, el dispositivo portal transmite y recibe datos a / desde una red celular, mediante el estándar de datos en paquetes digitales celulares ("CDPD"). Como se conoce en la técnica, el estándar CDPD es un estándar inalámbrico digital que se despliega como una mejora de la red celular analógica existente. Proporciona un recubrimiento por paquetes sobre la red AMPS (Sistema Avanzado de Telefonía Móvil) y mueve datos a 19,2 Kbps en intervalos no usados, que cambian constantemente, en los canales estándar de voz. En consecuencia, esta realización del dispositivo portal es capaz de explotar el ancho de banda usualmente no usado en una red celular analógica de alcance nacional. Las realizaciones del dispositivo portal también pueden configurarse para transmitir / recibir datos usando una gran variedad de otros estándares de comunicación, incluso estándares de paginación bidireccionales y estándares inalámbricos de tercera generación ("3G") (p. ej., UTMS, CDMA 2000, NTT DoCoMo, etc.).

Como se ha descrito anteriormente, debido a que ciertas realizaciones del dispositivo portal están configuradas para procesar código interpretado independiente del hardware (p. ej., mediante un módulo intérprete 520 tal como una máquina virtual de Java), las aplicaciones pueden migrar a las nuevas plataformas de hardware sin cambios significativos. Además, según se ilustra en la Figura 4, en una realización, la funcionalidad de comunicaciones se proporciona mediante una interfaz 916 de red modular, que puede ser fácilmente modificada / reemplazada sin alterar aplicaciones 910 existentes del dispositivo portal o partes significativas del intérprete 912 de códigos de octetos. Por ejemplo, al cambiar de una red CDPD a una red 3G, solamente el componente 916 de interfaz de red del intérprete 912 de la Máquina Virtual necesitará ser actualizado (junto con cualquier hardware 914 de 3G requerido) para dar soporte al nuevo protocolo 3G.

Como se ha descrito anteriormente (y como se indica en la Figura 4), en una realización, el módulo intérprete 912 en el dispositivo portal es una máquina virtual de Java. Así pues, esta realización del dispositivo portal es capaz de ejecutar una vasta biblioteca de aplicaciones 910 existentes de Java, independientes del hardware (p. ej., miniaplicaciones / códigos de octeto). Además, según se indica en la Figura 4, una realización del dispositivo portal emplea un microprocesador de 32 bits basado en el RISC (Código de un Conjunto Reducido de Instrucciones), tal como un procesador de ARM (Modalidad de Respuesta Asíncrona). Como se conoce en la técnica, los procesadores de ARM son extensamente usados en las agendas electrónicas, los teléfonos celulares y en una gran variedad de otros dispositivos inalámbricos. Debería observarse, sin embargo, que pueden usarse varias otras arquitecturas de hardware y software (y / o firmware) para el dispositivo portal, cumplimentando a la vez los principios subyacentes de la invención.

Como se ha descrito anteriormente, una realización del servidor portal convierte aplicaciones y datos estándar a un formato que el dispositivo portal puede interpretar debidamente. En consecuencia, según se ilustra en la Figura 4, esta realización del servidor portal puede incluir un módulo 920 de conversión de contenido para procesar solicitudes de contenido 940 de Internet del dispositivo portal. Más específicamente, en una realización, el servidor portal actúa como un agente para el dispositivo portal, remitiendo solicitudes 940, 941 de Internet a la sede adecuada de Internet, en nombre del dispositivo portal, recibiendo respuestas desde la sede de Internet en un formato estándar de Internet (p.

ej., páginas de la Red con contenido incrustado de audio / video y gráfico) y convirtiendo las respuestas 924 estándar de Internet a un formato que el dispositivo portal pueda procesar (p. ej., códigos de octeto).

Por ejemplo, el módulo 920 de conversión puede incluir un módulo de representación del lenguaje de marcado de hipertexto ("HTML") (no mostrado) para interpretar el código HTML y descargar cualquier contenido incrustado en el código HTML (p. ej., gráficos, video, sonido, etc.) al servidor portal. El módulo 920 de conversión puede luego combinar el código HTML y el contenido incrustado y generar un conjunto de códigos de octeto para reproducir exactamente el contenido solicitado en el dispositivo portal. Como se ha descrito anteriormente, en una realización, los códigos de octeto pueden ser códigos de octeto, o miniaplicaciones, de Java. Sin embargo, pueden generarse varios otros tipos de código interpretado y / o no interpretado, según el tipo específico de dispositivo portal que se esté usando (p. ej., uno con un módulo intérprete o uno sin él).

Debido a que el servidor portal tiene un conocimiento íntimo de las capacidades / configuración de cada dispositivo portal (p. ej., tamaño de pantalla, capacidades de gráficos / audio, memoria disponible, potencia de procesamiento, preferencias del usuario, etc.), puede reconstruir exactamente el contenido de Internet solicitado, minimizando a la vez el ancho de banda requerido para transmitir el contenido al dispositivo. Por ejemplo, el módulo 920 de conversión puede realizar el preajuste a escala y los ajustes de intensidad del color al contenido solicitado, de modo tal que se represente debidamente dentro del visor del dispositivo portal. Al hacer estos cálculos, la conversión puede incluir como factor la memoria y la potencia de procesamiento disponibles en el dispositivo portal. Además, el módulo 920 de conversión puede comprimir el contenido solicitado usando una gran variedad de técnicas de compresión (y preservar por ello el ancho de banda de la red).

En una realización, el módulo 920 de conversión simplemente descartará el contenido de Internet que no pueda reproducirse en el dispositivo portal, o bien aquel que el usuario haya indicado que no desea reproducir en el dispositivo portal. Por ejemplo, un usuario puede indicar que no desea que se generen sonidos en el dispositivo portal o que no quiere anuncios transmitidos al dispositivo portal. El módulo 920 de conversión eliminará entonces todo sonido o anuncio incrustado en la página de la Red solicitada (u otro contenido de Internet solicitado). Debido a que la representación de HTML y otro procesamiento avanzado de contenido / datos de Internet se desvía al servidor portal según lo descrito anteriormente, el dispositivo portal puede fabricarse usando un microprocesador o microcontrolador de baja potencia, reduciendo por ello el coste de fabricación y / o la energía consumida por el dispositivo.

En una realización, cuando una página específica de la Red u otro objeto de Internet ha sido convertido a un formato adecuado para su ejecución en el dispositivo portal (p. ej., códigos de octeto y datos de Java), la página, o el objeto, formateados pueden almacenarse localmente en una memoria caché 925 en el servidor portal (u otro servidor mantenido por la PO). Por tanto, la próxima vez que se solicite el contenido, el módulo 920 de conversión puede sencillamente leer el código previamente generado en la memoria caché local 925 (es decir, ya no necesitará extraer el contenido de ubicaciones remotas para reconstruir el código).

Pueden implementarse diversas técnicas y algoritmos de almacenamiento en memoria caché para garantizar que la memoria caché 925 está almacenando eficazmente los datos de Internet (es decir, dando como resultado un porcentaje aceptable de 'aciertos' en memoria caché) y que los datos son actuales. Por ejemplo, el servidor portal puede almacenar en memoria caché los datos de Internet más frecuentemente solicitados (p. ej., la página inicial de Yahoo™) y puede eliminar contenidos de la memoria caché en base a una política de almacenamiento en memoria caché de uso menos reciente. Además, para garantizar que los datos almacenados en la memoria caché son actuales, el servidor portal puede comparar la versión de los datos almacenados en la memoria caché 925 con la versión de los datos almacenados en la sede remota de Internet cuando se solicitan los datos. De manera similar, el servidor portal puede almacenar datos en la memoria caché 925 durante algún periodo de tiempo predeterminado antes de comprobar un servidor remoto en busca de una nueva versión. Pueden emplearse varias otras técnicas de almacenamiento en memoria caché en Internet, cumplimentando a la vez los principios subyacentes de la invención (p. ej., los definidos en el Protocolo de Almacenamiento en Memoria Caché de Internet ("ICP") y / o el Protocolo de Encaminamiento de Formaciones de Memoria Caché ("CARP")).

Procesamiento de ficheros de clase

En una realización de la invención, los ficheros de clase (u otros tipos de código de programa) se comprimen antes de ser transferidos al dispositivo portal (u otro dispositivo de procesamiento de datos). La compresión puede realizarse manualmente (p. ej., por el personal de la organización portal) o automáticamente, usando un algoritmo de compresión / conversión implementado en ordenador. Se describirán ahora varias técnicas de compresión / conversión de ficheros de clase con respecto a las **Figuras 5 a 7**.

Según se ilustra en la **Figura 5**, en una realización, cada uno de los ficheros de clase usados en un programa específico de aplicación (o miniaplicación) se combinan para formar un objeto de programación unificado, mencionado en el presente documento como un "lote" 1200. Con fines de ilustración, el lote específico 1200 mostrado en la **Figura 5** se construye a partir de los ficheros 200, 210, 220 y 230 de clase de la **Figura 2**. Más específicamente, las entradas

redundantes 203, 213, 223 y 233 de RefMétodo Foo se combinan en una única entrada "global" 1203 RefMétodo Foo en un fondo común 1202 de constantes compartidas dentro del lote 1200. De manera similar, las entradas redundantes 240 a 242 de RefCampo Barra se combinan en una única entrada global RefCampo Barra dentro del fondo común de constantes compartidas, reduciendo por ello adicionalmente los requisitos de memoria de la aplicación.

- 5 Los métodos y campos de los ficheros originales 200, 210, 220, 230 de clase se copian asimismo al lote, junto con varios otros objetos de ficheros de clase (no mostrados). Por tanto, el Método Foo 236 se copia al lote como la entrada 1236 de Información de Método y el Campo Barra 238 se copia al lote como la entrada 1238 de Información de Campo. Además, los datos 201, 211, 221 y 231 de cabecera para cada uno de los ficheros de clase se copian al lote, respectivamente, como objetos 1205 a 1208 de Información de Clase.
- 10 En una realización, a diferencia de las entradas del fondo común de constantes en los ficheros de clase estándar, las entradas 1202 del fondo común de constantes compartidas del lote son entradas de longitud fija (p. ej., de 32 bits de longitud). En consecuencia, las entradas del fondo común de constantes y otros datos / códigos dentro del lote se refieren entre sí en base a un valor de desplazamiento desde el extremo superior del fondo común de constantes (es decir, en lugar de en base a una dirección en la memoria). Los desplazamientos se generan durante el proceso de enlace del lote (descrito en mayor detalle más adelante). Por ejemplo, con referencia a la **Figura 6**, a la entrada RefCampo Barra dentro del fondo común de constantes compartidas se asigna un número de ranura basado en su desplazamiento desde el extremo superior del fondo común de constantes: ranura Nº 5 en el ejemplo. La misma entrada del fondo común de constantes incluye un valor de desplazamiento que identifica dónde reside el código referido por la entrada dentro del lote. Por tanto, la entrada transformada 1236 de Información de Método (es decir, la
- 15 que contiene el código del Método Foo) puede ser localizada refiriéndose a la ranura Nº 5 del fondo común de constantes y leyendo el desplazamiento proporcionado por la entrada (identificado como "Desplazamiento X" en el ejemplo). La misma técnica de búsqueda puede usarse para localizar otros códigos / datos dentro del lote. Por ejemplo, la entrada 1238 de Información de Campo puede localizarse en base al desplazamiento (identificado como "Desplazamiento Y") proporcionado por la entrada del fondo común de constantes RefCampo Barra en la ranura Nº 7.
- 20 Por el contrario, una entrada de Información de Clase, una entrada de Información de Método o cualquier otro tipo de entrada dentro del lote 1200 puede incluir un puntero / desplazamiento usado para identificar una entrada específica del fondo común de constantes. Por ejemplo, un código de octeto incluido en la entrada 1236 de Información de Método puede causar la creación de una nueva clase de tipo definida por una entrada específica del fondo común de constantes. Así, según se indica en la **Figura 6**, la entrada 1236 de Información de Método contendrá un puntero a su
- 25 objeto 1208 de Información de Clase, que incluye un puntero a la entrada identificada del fondo común de constantes (es decir, la entrada Nº 6). Así, debido a que todos estos elementos son auto-referentes, un método puede hallar una clase que tiene como dueña y la clase puede identificar el lote al cual está anexada. Además, debido a que todos los elementos se localizan con referencias mutuas dentro del lote, no se requieren técnicas complejas de direccionamiento para identificar los elementos del lote en la memoria (es decir, el lote puede cargarse aleatoriamente en cualquier área de memoria).
- 30 Durante el proceso de generación / enlace del lote, ciertos tipos de datos pueden copiarse directamente desde el fichero de clase al cual pertenecen, mientras que otros tipos de datos / código de fichero de clase deben modificarse y / o crearse. Por ejemplo, durante la generación de objetos de Información de Clase, los siguientes datos pueden copiarse de forma esencialmente no modificada (o levemente modificada) al lote: la referencia de clase a sí misma y a la superclase mediante el fondo común de constantes; el total de campos y métodos; y los indicadores de acceso / atributo (p. ej., público, privado, protegido, estático, abstracto, etc.). Un retropuntero relativo al lote contenedor se crea durante el proceso de generación del lote, de modo tal que la clase pueda identificar el lote al cual pertenece (p. ej., para que puede referirse a entradas del fondo común de constantes dentro del lote). De manera similar, se genera una referencia al objeto de clase durante el proceso de generación.
- 35 Para objetos de Información de Método, los datos que pueden copiarse directamente incluyen una referencia al nombre del método mediante el fondo común de constantes; una referencia a la firma del tipo, mediante el fondo común de constantes; indicadores cualesquiera de acceso / atributo; datos de máximas pilas; datos de máximos elementos locales; el total de argumentos y el total de retorno, por nombrar solamente unos pocos. Un retropuntero relativo al objeto de Información de Clase a la cual pertenece el método también será generado (es decir, de modo tal que el
- 40 objeto de Información de Clase pueda identificar el objeto de Información de Clase dentro del lote). Los códigos de octeto del método, las tablas de excepciones y / o los números de ranura de la tabla virtual ("tablav") asociados al método también pueden modificarse durante el proceso de generación del lote, para admitir las nuevas localizaciones de datos dentro del fondo común de constantes compartidas. Para métodos nativos, los punteros funcionales a las implementaciones de código nativo se rellenarán durante la carga del lote.
- 45 Para la generación de Información de Campo, se genera un puntero relativo que identifica la clase a la cual pertenece (es decir, el objeto Información de Clase) y un desplazamiento del campo dentro de la instancia del objeto. Los datos que pueden copiarse directamente al lote incluyen una referencia al nombre del campo mediante el fondo común de
- 50
- 55

constantes; una referencia a la firma del tipo de campo mediante el fondo común de constantes; e indicadores cualesquiera de acceso / atributo.

Una realización específica de un proceso de generación / enlace de lote para ficheros de clase se describirá ahora con respecto al diagrama de flujo en la **Figura 7**. Debería observarse, sin embargo, que varias etapas estipuladas en la **Figura 7** son con fines de ilustración y no son necesarias para cumplimentar los principios subyacentes de la invención.

En 1400, se carga en memoria y se valida un fichero de clase a incorporar a un lote. Por ejemplo, si el fichero de clase es un fichero de clase de Java, el fichero es validado según la especificación de la máquina virtual de Java. Para el funcionamiento estándar de la máquina virtual de Java, la validación ocurre habitualmente en tiempo de ejecución, cuando una clase se carga por primera vez. Debido a que esta realización de la invención valida cada uno de los ficheros de clase antes de incorporarlos al lote, sin embargo, no se requiere la validación en tiempo de ejecución del fichero de clase, o lote, reduciendo por ello adicionalmente los requisitos de procesamiento del dispositivo sobre el cual se ejecuta el lote (p. ej., el dispositivo portal). El lote puede estar firmado criptográficamente para proporcionar prueba de su autenticidad.

En 1405, se crea un objeto de Información de Clase para el fichero de clase, usando datos extraídos de la cabecera del fichero de clase, que incluyen, por ejemplo, el nombre del fichero de clase y otros datos de identificación del fichero de clase y parámetros asociados. En 1410, todas las referencias al fondo común de constantes del fichero de clase (es decir, las entradas "locales" del fondo común de constantes) se asocian a las correspondientes entradas "globales" en el fondo común de constantes compartidas del lote. Si ya existe una entrada global, determinada en 1415 (p. ej., porque otro fichero de clase contenía la entrada), entonces se generan objetos de Información de Método para cada método en el fichero de clase en 1425. Si no existe la entrada global, entonces se crea una nueva entrada global en el fondo común de constantes compartidas en 1420, antes de que se generen los objetos de Información de Método.

Todos los métodos no nativos (p. ej., aquellos métodos que contienen código efectivo de octetos), identificados en 1430, se modifican de la siguiente manera. Los códigos de octeto dentro del método se cargan en un formato de memoria interna en 1435 (p. ej., una lista enlazada de objetos de código operativo de Java, uno por código operativo de Java en el código de octeto). En 1440, las referencias numéricas que identifican las entradas locales se convierten en punteros a entradas globales tales como las entradas del fondo común de constantes compartidas descritas anteriormente. En 1445, ciertos códigos de octeto se convierten / reescriben en formatos más convenientes. Finalmente, en 1450, la tabla de excepciones del método se convierte en referencias a objetos de código operativo de Java, en lugar de referencias numéricas a direcciones de códigos de octeto. Cuando el procesamiento de métodos está completo, en 1455, se crean objetos de Información de Campo para cada campo en el fichero de clase y se incorporan al lote. En 1460, el procesamiento para el fichero de clase está completo y se identifica el próximo fichero de clase a procesar.

Una vez que se ha cargado y enlazado un conjunto completo de ficheros de clase en la representación temporal en la memoria del lote, se realiza una validación adicional para garantizar que todas las referencias estén resueltas. La representación en la memoria se graba en un objeto de lote binario con referencias relativas (mediante un número de entrada del fondo común de constantes globales) en lugar de punteros.

En consecuencia, una realización de la invención convierte temporalmente los códigos de octeto en un gráfico de objetos de códigos operativos de Java, de modo tal que las referencias del fondo común de constantes puedan ajustarse de números locales a globales, y para reescribir ciertos códigos de octeto. Este proceso puede cambiar el tamaño de algunos códigos de octeto, por lo que el gráfico se usa para rastrear a dónde apuntaban las operaciones de bifurcación antes de que se modificaran los códigos de octeto. Los códigos de octeto se modifican y ajustan adecuadamente antes de ser grabadas de nuevo en el lote.

A modo de ejemplo específico, suponiendo que se usen las siguientes líneas de código:

20 op_ifne 5
45 23 op_ldc #3
25 ...

el código de octeto en 20 (op_ifne) salta 5 octetos hacia adelante si una expresión no es igual a cero. El código de octeto op_ldc carga un elemento del fondo común de constantes (entrada #3). Si esa entrada del fondo común de constantes se retransformara, digamos, en #300, los dos octetos de op_ldc ya no funcionarían (solamente pueden referirse a las entradas entre 0 y 255) y, por lo tanto, se reescribirían en una instrucción op_ldcw de tres octetos (por supuesto, ahora la instrucción op_ifne necesita ser ajustada, etc.)

Además, en una realización, las entradas en el fondo común de constantes del lote pueden referirse a entradas en un fondo común de constantes de un lote de sistema compartido (mediante un bit especial fijado en el identificador de

5 tipo). Esto permite dotar a un lote de sistema con la biblioteca nuclear de clases y cada aplicación en su propio lote (de modo que las aplicaciones puedan añadirse / eliminarse fácilmente). Además, en una realización, los códigos de octeto se transforman en una máquina virtual similar que mantiene la semántica de la máquina virtual de Java, pero da soporte a un cierto número de códigos de octeto alternativos que proporcionan un código objeto más compacto y / o un funcionamiento más rápido del sistema.

10 Las realizaciones de la invención pueden incluir varias etapas, como se ha expuesto anteriormente. Las etapas pueden realizarse en instrucciones ejecutables por máquina. Las instrucciones pueden usarse para hacer que un procesador de propósito general o de propósito especial realice ciertas etapas. Alternativamente, estas etapas pueden ser llevadas a cabo por componentes específicos de hardware que contienen lógica cableada para realizar las etapas, o por cualquier combinación de componentes informáticos programados y componentes de hardware a medida.

15 Los elementos de la presente invención también pueden proporcionarse como un medio legible por máquina para almacenar las instrucciones ejecutables por máquina. El medio legible por máquina puede incluir, pero no se limita a, disquetes flexibles, discos ópticos, CD-ROM y discos magneto-ópticos, memorias ROM, RAM, EPROM, EEPROM, tarjetas magnéticas u ópticas, medios de propagación u otro tipo de medios, o medio legible por máquina, adecuados para almacenar instrucciones electrónicas. Por ejemplo, la presente invención puede descargarse como un programa de ordenador que puede transferirse desde un ordenador remoto (p. ej., un servidor) a un ordenador solicitante (p. ej., un cliente) mediante señales realizadas en una onda portadora u otro medio de propagación, mediante un enlace de comunicación (p. ej., un módem o conexión de red).

20 En toda la descripción precedente, con fines de explicación, se expusieron numerosos detalles específicos, a fin de proporcionar una comprensión exhaustiva de la invención. Será evidente, sin embargo, a alguien experto en la técnica que la invención puede ponerse en práctica sin algunos de estos detalles específicos. Por ejemplo, si bien el sistema descrito anteriormente se centra en una implementación de ficheros de clase de Java, los principios subyacentes de la invención pueden llevarse a cabo en varios otros tipos de código objeto (p. ej., C++). En consecuencia, el alcance de la invención debería juzgarse en términos de las reivindicaciones dadas a continuación.

25

REIVINDICACIONES

1. Un procedimiento implementado en un ordenador para combinar una pluralidad de ficheros (200, 230) de clase en un lote unificado (1200) de ficheros de clase que incluye los métodos (204, 205, 206, 234, 235, 236) y los campos (207, 208, 237, 238) de los ficheros (200, 230) de clase originales, comprendiendo el procedimiento:
- 5 copiar los métodos (204, 205, 206, 235, 236) de los ficheros (200, 230) de clase originales al lote unificado (1200) de ficheros de clase, generando por ello una entrada (1208, 1209, 1236) de método global dentro del lote unificado (1200) de ficheros de clase para cada método (204, 205, 206, 234, 235, 236) dentro de la pluralidad de ficheros (200, 230) de clase;
- 10 copiar los campos (207, 208, 237, 238) de los ficheros (200, 230) de clase originales al lote unificado (1200) de ficheros de clase, generando por ello una entrada (1210, 1238) de campo global dentro del lote unificado (1200) de ficheros de clase para cada campo (207, 208, 237, 238) dentro de la pluralidad de ficheros (200, 230) de clase;
- 15 convertir las entradas (203, 242, 233, 241) del fondo común de constantes locales de los fondos comunes (202, 232) de constantes locales de cada uno entre la pluralidad de ficheros (200, 230) de clase que hacen referencia a los métodos (204, 205, 206; 234, 235, 236) y los campos (207, 208, 237, 238) de cada fichero (200, 230) de clase, en las entradas (1203, 1204) del fondo común de constantes globales dentro del lote unificado (1200) de ficheros de clase, en donde la conversión comprende eliminar las entradas (203, 242, 233, 241) redundantes del fondo común de constantes locales,
- caracterizado porque** la conversión comprende adicionalmente reemplazar las referencias que apuntan desde las entradas (263, 242, 233, 241) del fondo común de constantes locales a los métodos (236) y campos (237) por valores de desplazamiento proporcionados en ranuras, a las que se asignan números de ranura en base a desplazamientos desde el extremo superior de las entradas (1203, 1204) del fondo común de constantes globales, apuntando los valores de desplazamiento desde las ranuras a las entradas (1208, 1209, 1236) de métodos globales y a las entradas (1210, 1238) de campos, respectivamente, dentro del lote unificado (1200) de ficheros de clase.
- 20
2. El procedimiento según la reivindicación 1, en el cual la combinación comprende adicionalmente convertir las entradas (203, 242, 233, 241) del fondo común de constantes locales de longitud variable, de cada uno de los ficheros (200, 230) de clase con entradas del fondo común de constantes de longitud fija dentro del lote unificado (1200) de ficheros de clase.
- 25
3. El procedimiento según la reivindicación 1, en el cual la combinación comprende adicionalmente:
- 30 para cada entrada (203, 242, 233, 241) del fondo común de constantes locales, determinar si ya existe una correspondiente entrada (1203, 1204) del fondo común de constantes globales; y
- si ya existe una correspondiente entrada (1203, 1204) del fondo común de constante globales, entonces usar la entrada existente (1203, 1204) del fondo común de constantes globales en lugar de la entrada (203, 242, 233, 241) del fondo común de constantes locales; y
- 35 si no existe la entrada (1203, 1204) del fondo común de constantes globales, entonces generar una nueva entrada (1203, 1204) del fondo común de constantes globales a partir de la entrada (203, 242, 233, 241) del fondo común de constantes locales.
4. El procedimiento según la reivindicación 1, en el cual cada entrada (1208, 1209, 1236) de método global comprende un objeto de Información de Método, y en el cual la generación del objeto de Información de Método comprende copiar directamente datos de un primer tipo de cada método de cada fichero (200, 230) de clase y modificar datos de un segundo tipo antes de copiar los datos de cada método del fichero (200, 230) de clase.
- 40
5. El procedimiento según la reivindicación 1, en el cual los datos del primer tipo incluyen una referencia al nombre del método mediante el fondo común de constantes; una referencia a una firma de tipo mediante el fondo común de constante; indicadores de acceso o atributo; datos de máximas pilas; datos de máximos elementos locales; un total de argumentos y / o un total de retorno.
- 45
6. El procedimiento según la reivindicación 4, en el cual los datos del segundo tipo incluyen códigos de octeto de Método, tablas de excepciones y / o números de ranura de tabla virtual asociados al método.
7. El procedimiento según la reivindicación 1, en el cual cada entrada (1210, 1238) de campo global comprende un objeto de Información de Campo, y en el cual la generación del objeto de Información de Campo comprende copiar directamente datos de un primer tipo de cada campo de cada fichero (200, 230) de clase y modificar datos de un segundo tipo antes de copiar los datos de cada campo del fichero (200, 230) de clase.
- 50
8. El procedimiento según la reivindicación 5, en el cual los datos del primer tipo incluyen una referencia al nombre del

campo mediante un fondo común de constantes; una referencia a la firma del tipo de campo mediante el fondo común de constantes; e / o indicadores cualesquiera de acceso / atributo.

9. El procedimiento según la reivindicación 1, que comprende adicionalmente:

- 5 generar un objeto de Información de Clase dentro del lote unificado (1200) de ficheros de clase para cada cabecera (201, 231) de cada fichero (200, 230) de clase, conteniendo cada objeto de Información de Clase datos de cabecera de cada cabecera (201, 231) de fichero de clase.

10. El procedimiento según la reivindicación 7, que comprende adicionalmente:

- 10 generar un puntero relativo para cada entrada (1208, 1209, 1236) de método global y cada entrada (1210, 1238) de campo global, apuntando el puntero relativo a un objeto de Información de Clase con el que está asociada la entrada (1208, 1209, 1236) de método global o la entrada (1210, 1238) de campo global.

11. El procedimiento según la reivindicación 1, que comprende adicionalmente:

- 15 generar un puntero relativo para cada entrada (1208, 1209, 1236) de método global y cada entrada (1210, 1238) de campo global, apuntando el puntero relativo a una entrada (1203, 1204) del fondo común de constantes globales con la cual está asociada la entrada (1208, 1209, 1236) de método global o la entrada (1210, 1238) de campo global.
- 15 12. El procedimiento según la reivindicación 1, en el cual, para disponer los elementos de los dos o más ficheros (200, 230) de clase, un módulo (920) de conversión de contenido combina entradas redundantes del fondo común de constantes de dichos dos o más ficheros (200, 230) de clase, para formar una entrada (1203, 1204) global del fondo común de constantes en un fondo común (1202) de constantes compartidas dentro de dicho lote unificado (1200) de ficheros de clase.

20 13. El procedimiento según la reivindicación 12, en el cual la combinación comprende adicionalmente:

reescribir dichas entradas (1203, 1204) del fondo común de constantes globales para apuntar a los elementos contenidos dentro de dicho lote unificado (1200) de ficheros de clase, correspondiendo dichos elementos a elementos contenidos en dichos uno o más ficheros (200, 230) de clase, y previamente identificados por dichas una o más entradas redundantes del fondo común de constantes.

25 14. El procedimiento según la reivindicación 13, en el cual una de dichas entradas (1203, 1204) del fondo común de constantes globales es una entrada de RefMétodo y dicho elemento identificado por la entrada del fondo común de constantes es un método copiado a dicho lote unificado (1200) de ficheros de clase desde uno o más ficheros (200, 230) de clase.

30 15. El procedimiento según la reivindicación 14, en el cual, para disponer los elementos de los dos o más ficheros (200, 230) de clase, el módulo (920) de conversión de contenido convierte referencias numéricas en entradas locales dentro de un código de octeto en dicho método, o punteros a entradas (1203, 1204) del fondo común de constantes globales.

35 16. El procedimiento según la reivindicación 15, en el cual, para disponer los elementos de los dos o más ficheros (200, 230) de clase, el módulo (920) de conversión de contenido convierte una tabla de excepciones asociada a dicho método en referencias a objetos de código operativo de Java ("objetos jop"), en lugar de referencias numéricas a direcciones de códigos de octeto.

17. El procedimiento según la reivindicación 13, en el cual una de dichas entradas (1203, 1204) del fondo común de constantes globales es una entrada RefCampo.

40 18. El procedimiento según la reivindicación 1, en el cual, para disponer los elementos de los dos o más ficheros (200, 230) de clase, un módulo (920) de conversión de contenido valida dichos dos o más ficheros (200, 230) de clase antes de asociar dichos elementos para formar dicho lote unificado (1200) de ficheros de clase.

45 19. El procedimiento según la reivindicación 18, en el cual, para disponer los elementos para los dos o más ficheros (200, 230) de clase, el módulo (920) de conversión de contenido convierte los ficheros (200, 230) de clase en un gráfico de objetos de código operativo de Java ("objetos jop") para rastrear a dónde apuntaban las operaciones antes de la modificación de dichos ficheros (200, 230) de clase; ajusta referencias del fondo común de constantes, de números locales a globales, en base a dicho gráfico; y combina los ficheros (200, 230) de clase en el lote unificado (1200) de ficheros de clase.

50 20. El procedimiento según la reivindicación 1, en el cual un servidor portal (110) extrae automáticamente los ficheros (200, 230) de clase de una sede de Internet, a nombre del dispositivo de procesamiento de datos, en respuesta a solicitudes por los ficheros (200, 230) de clase provenientes del dispositivo de procesamiento de datos.

21. El procedimiento según la reivindicación 1, en el cual un servidor portal (110) extrae manualmente ficheros (200, 230) de clase de una sede de Internet, en nombre de un dispositivo de procesamiento de datos, en respuesta a solicitudes por parte de una organización que gestiona el servidor portal (110).

5 22. Un programa de ordenador que contiene una secuencia de código que, cuando es ejecutada por una máquina, hace que dicha máquina lleve a cabo el procedimiento de cualquiera de las reivindicaciones 1 a 21.

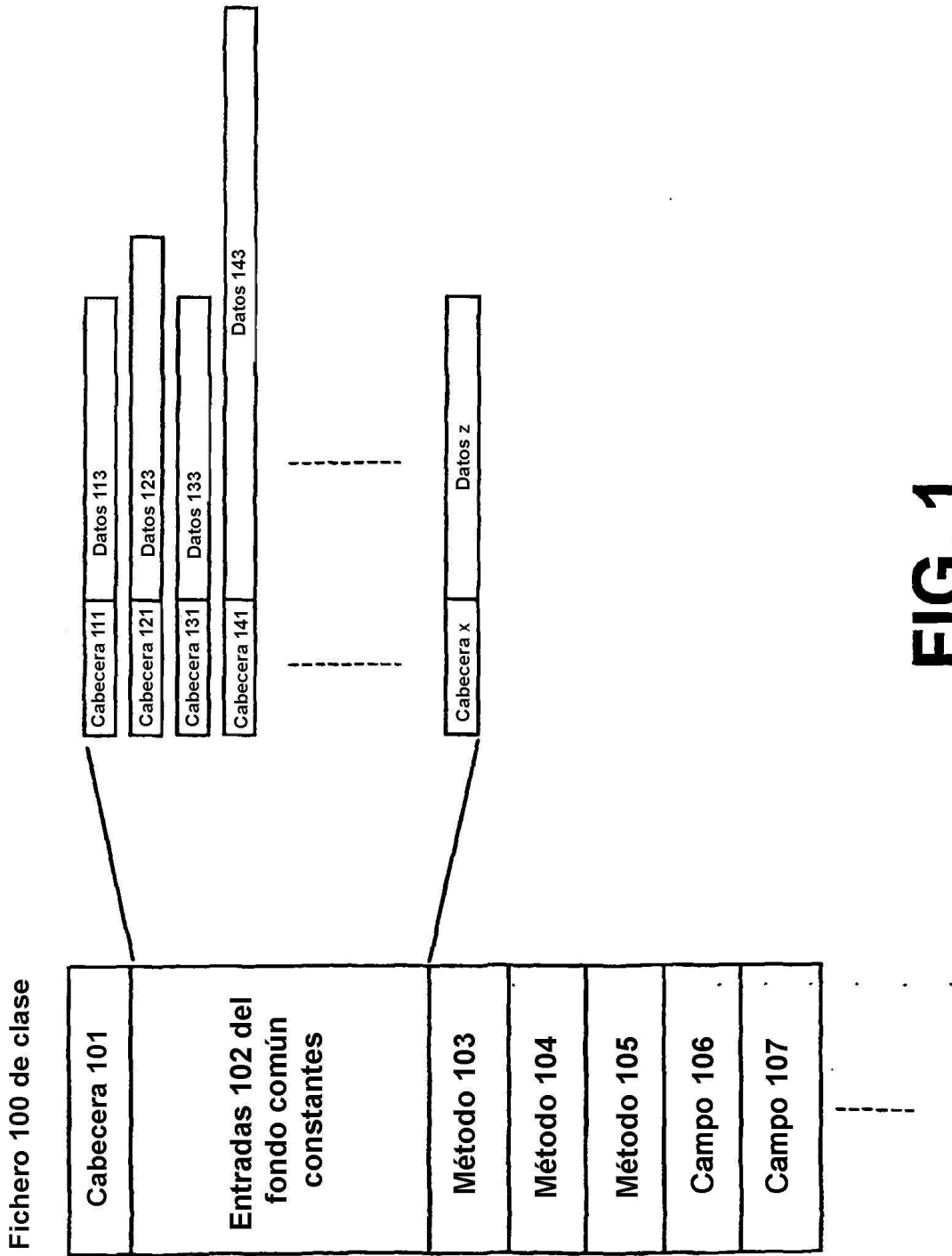


FIG. 1

(Técnica anterior)

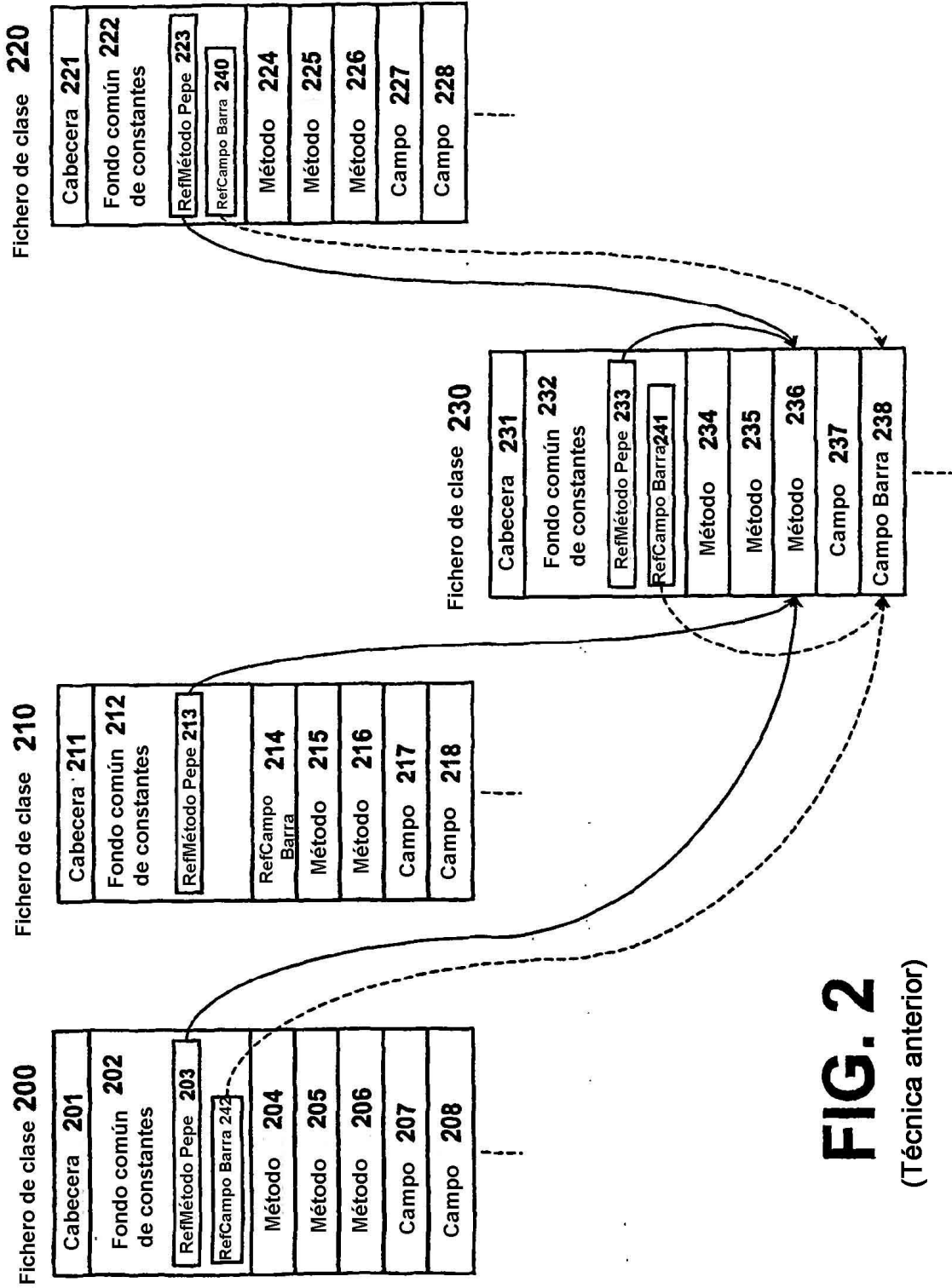


FIG. 2
(Técnica anterior)

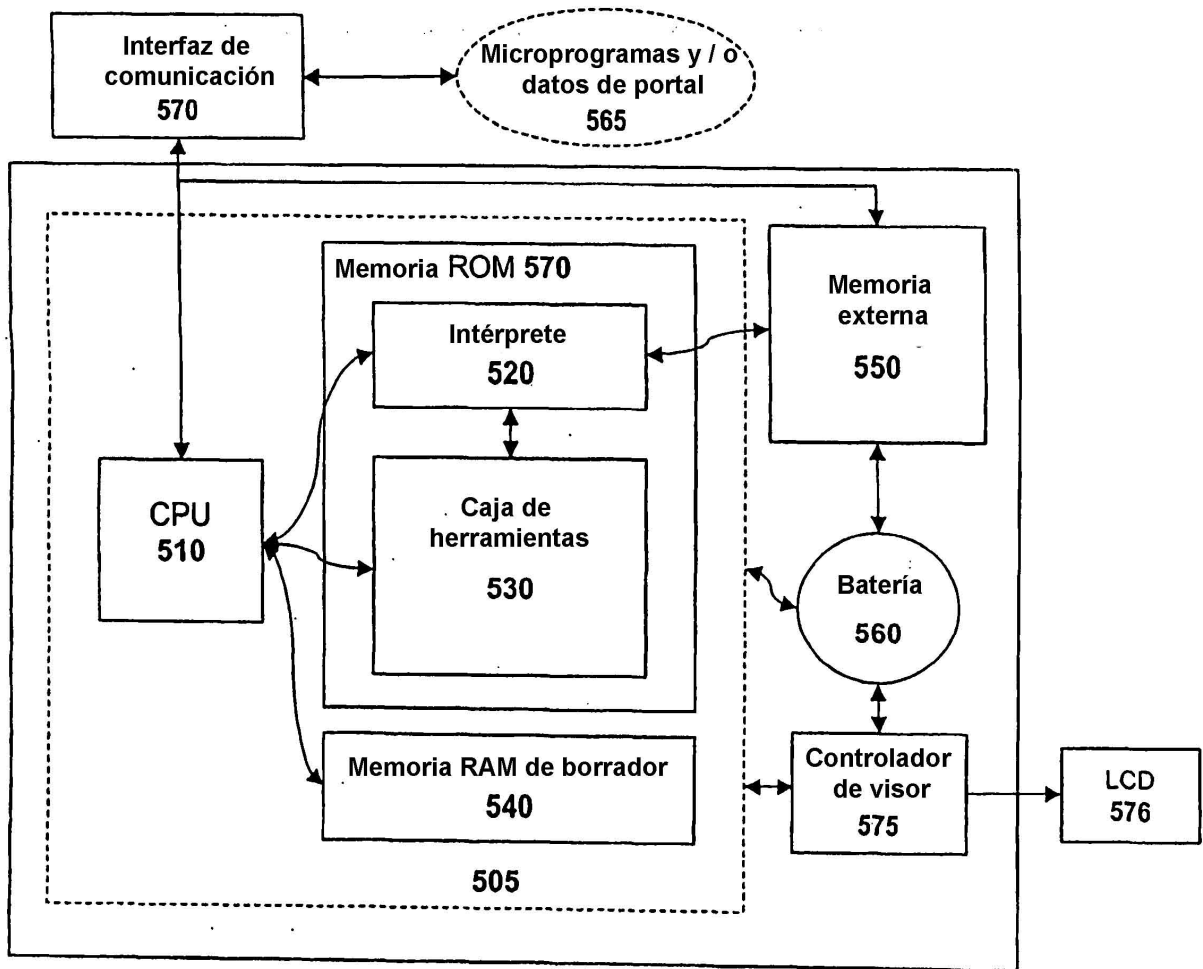


FIG. 3

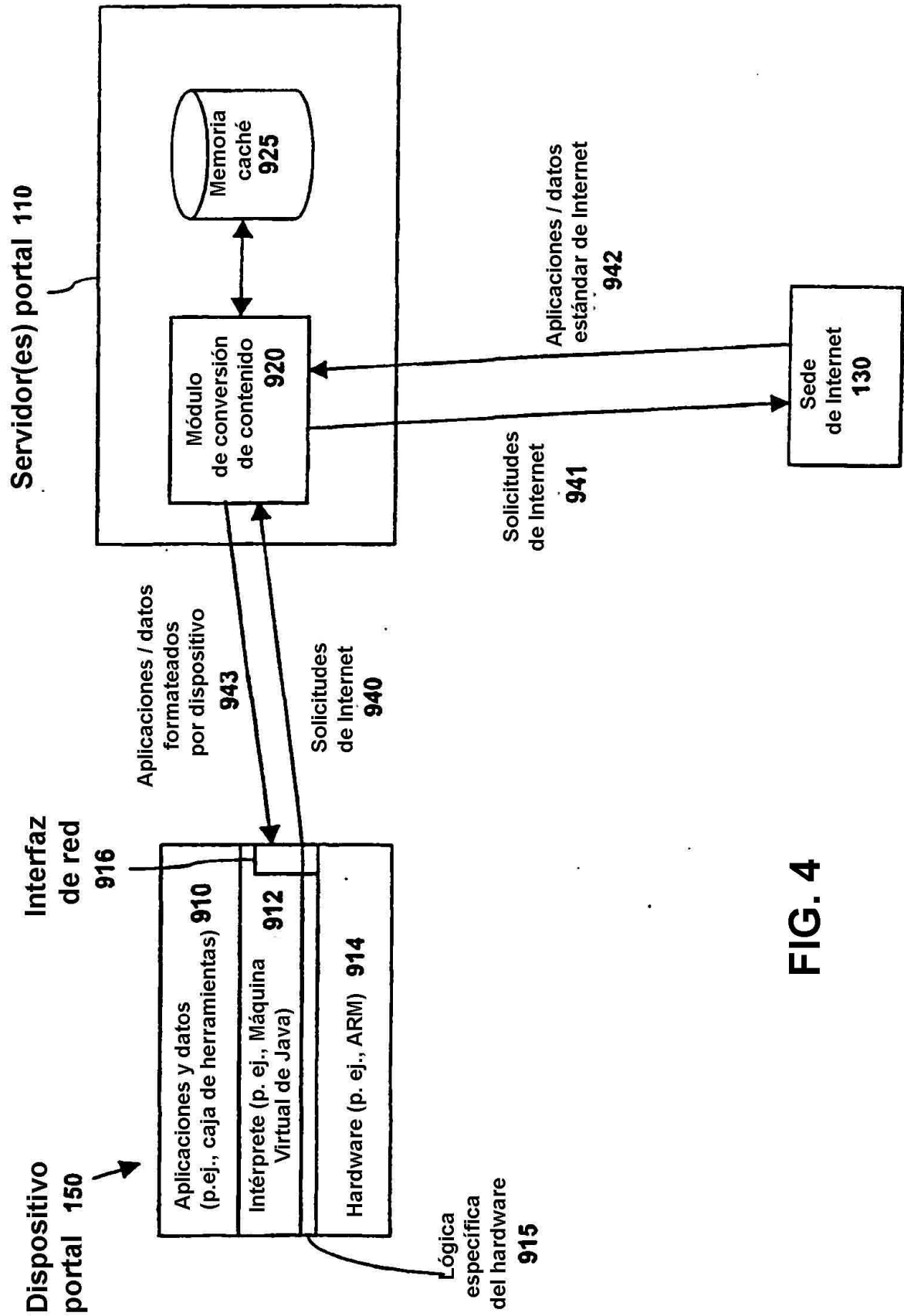


FIG. 4

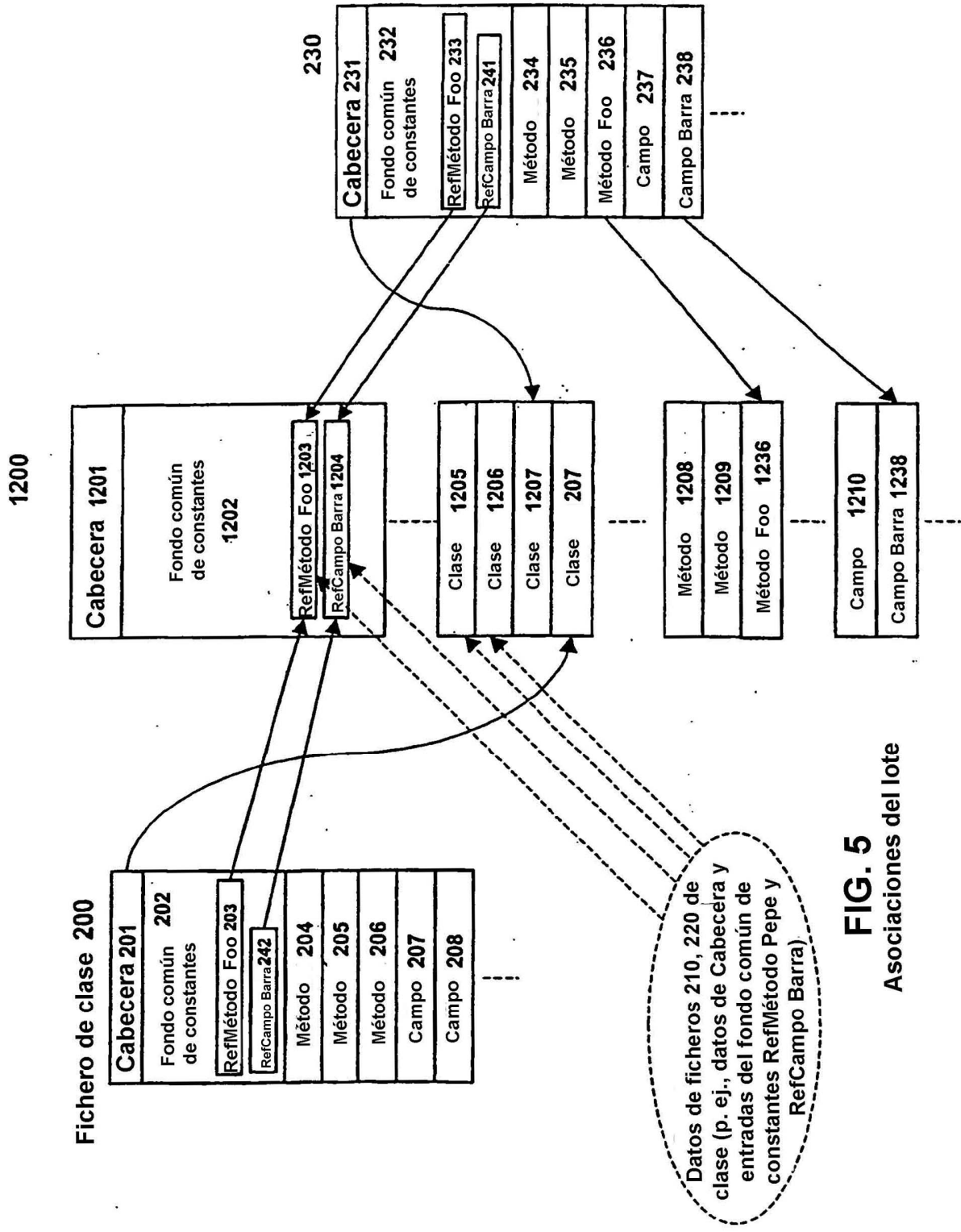


FIG. 5
Asociaciones del lote

Lote 1200

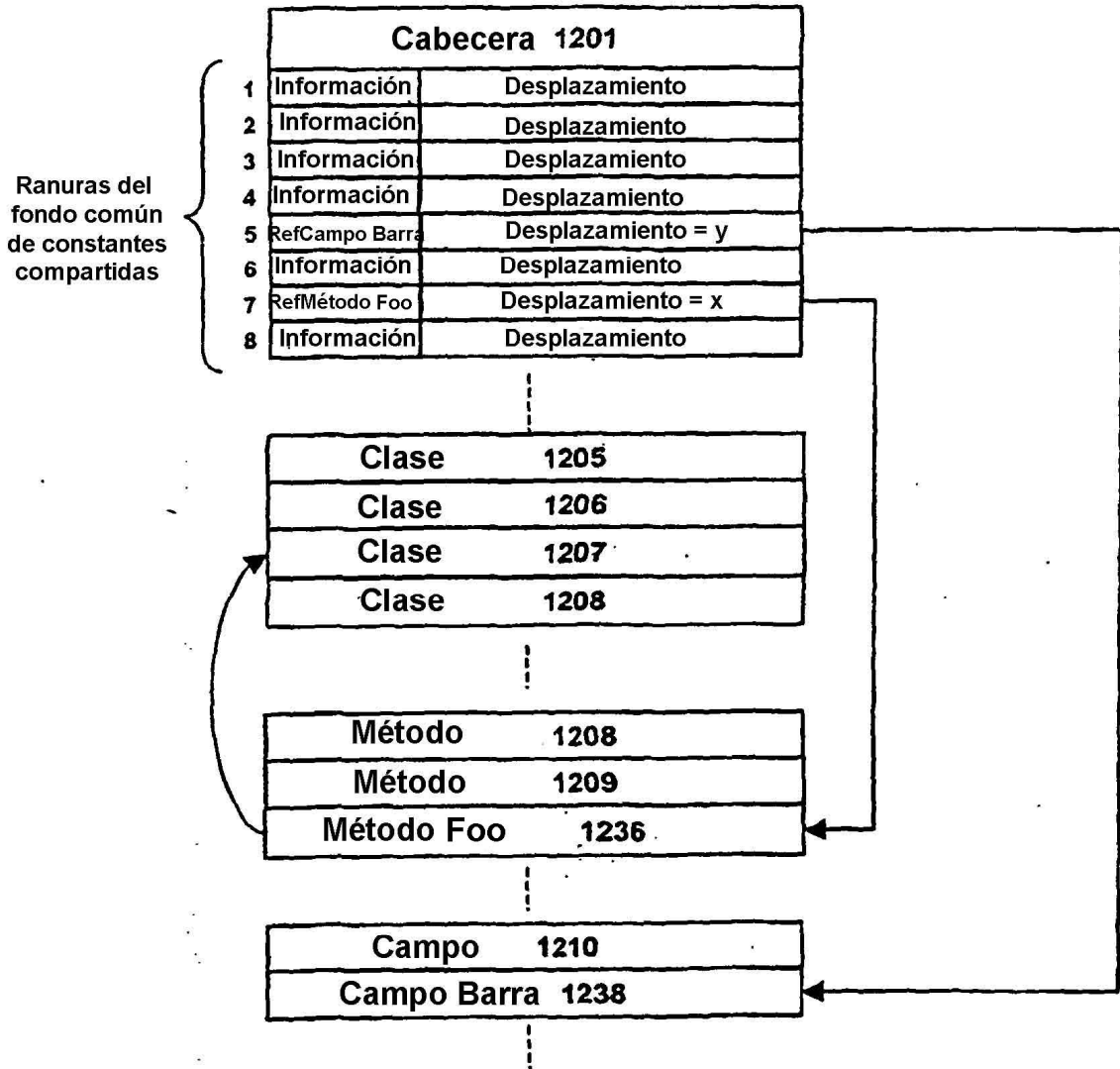


FIG. 6

FIG. 7

