

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 389 363**

51 Int. Cl.:
G06F 17/30 (2006.01)
G06F 17/24 (2006.01)
G06F 17/10 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

- 96 Número de solicitud europea: **05809470 .7**
96 Fecha de presentación: **11.10.2005**
97 Número de publicación de la solicitud: **1828941**
97 Fecha de publicación de la solicitud: **05.09.2007**

54 Título: **Dispositivo de tratamiento de datos con definición formal**

30 Prioridad:
12.10.2004 FR 0410736

45 Fecha de publicación de la mención BOPI:
25.10.2012

45 Fecha de la publicación del folleto de la patente:
25.10.2012

73 Titular/es:
PROGILYS (100.0%)
6, RUE SAINT SAËNS
F-75015 PARIS, FR

72 Inventor/es:
DE BECDELIEVRE, MICHEL y
MORAND, HERVÉ

74 Agente/Representante:
ESPIELL VOLART, Eduardo María

ES 2 389 363 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).

DESCRIPCION

Dispositivo de tratamiento de datos con definición formal

5 La invención tiene relación con los dispositivos de tratamiento de datos y, más particularmente, con un dispositivo de tratamiento de datos, el cual comprende un editor de fórmula, capaz de establecer una fórmula que obedece a unas primeras reglas formales, con ayuda de identificadores de variables, un juego de metadatos, capaz de mantener una correspondencia entre los identificadores de variables y unos datos almacenados en memoria, y un generador de código, capaz de generar un código que ejecuta la fórmula matemática sobre unos datos almacenados.

10 Los soportes lógicos matemáticos utilizan un lenguaje natural matemático, los cuales son capaces de interpretarlo para realizar los cálculos formales matemáticos. Sin embargo, no están adaptados para el tratamiento de un gran volumen de datos. Además, no están adaptados para la gestión de índices como para no definidas, por ejemplo bajo forma de función.

Las hojas de cálculo utilizan un lenguaje propio de las mismas para expresar un tratamiento sobre unos datos. Estas, además, están limitadas por el volumen de los datos que pueden tratar.

15 La patente US 2004/0133567 A1 describe optimizaciones y estrategias de ejecución para unas extensiones de tipo hoja de cálculo en lenguaje SQL (del inglés "Structured Query Language", o "lenguaje de consulta estructurado" en español). Se pueden definir cálculos que han de efectuarse sobre una base de datos utilizando una sintaxis de tipo hoja de cálculo en una cláusula insertada en el interior de una consulta SQL. Los cálculos son definidos directamente por medio de operaciones sobre las líneas y las columnas de una tabla rellena de datos extraídos de la base. La definición de los cálculos requiere sin embargo el conocimiento de la sintaxis del lenguaje de la hoja de cálculo y la localización de los datos en los cuales va a centrarse el cálculo en cuanto este último quede definido. Además, los cálculos que se pueden definir están limitados por las posibilidades de este tipo de lenguaje.

Los generadores de código se encargan de la transcripción de objetos informáticos a un código ejecutable. Estos no permiten transformar una fórmula matemática en código ejecutable. Además, están destinados a usuarios conocedores de la informática.

25 La invención tiene por objeto, en particular, superar los inconvenientes antedichos al proponer un dispositivo de tratamiento de datos, el cual comprende una unidad de tratamiento capaz de interactuar con una base de datos, comprendiendo la base de datos unas tablas de datos, una interfaz gráfica de usuario dispuesta para permitir a un usuario introducir una expresión matemática. La unidad de tratamiento comprende un editor de expresión matemática capaz de establecer una representación informática de trabajo de una expresión matemática en interacción con la interfaz gráfica de usuario, estando la expresión matemática en lenguaje natural matemático y comprendiendo ésta al menos un símbolo de variable matemática compleja, que presenta un índice y/o un exponente, y un símbolo matemático agregativo que opera con el símbolo de variable matemática compleja, un diccionario de símbolos dispuesto para almacenar unos primeros metadatos referentes al símbolo de variable matemática compleja, y comprendiendo unas informaciones sobre la estructura de las tablas de la base de datos, un editor de condiciones dispuesto para producir una expresión de condición, asociada al símbolo de variable matemática compleja, siendo la expresión de condición utilizable en combinación con los primeros metadatos para seleccionar un subconjunto de datos de la base de datos, un generador de consultas dispuesto para tratar la representación informática de trabajo para producir una secuencia de código que ejecuta la expresión matemática de manera acorde con las reglas del lenguaje natural matemático y, con el reconocimiento del símbolo matemático agregativo y del símbolo de variable matemática compleja con el cual el opera, de utilizar el editor de condiciones para producir una expresión de condiciones de trabajo, de los primeros metadatos para producir un código capaz de utilizar la expresión de condición para rellenar un cuadro de datos de trabajo con tal subconjunto de datos y de combinar el operador correspondiente al símbolo matemático agregativo con el cuadro de datos de trabajo.

45 El dispositivo según la invención permite así la definición de un tratamiento informático, por ejemplo un cálculo, el cual ha de realizarse sobre grandes volúmenes de datos, expresándose tal tratamiento en lenguaje natural matemático.

Otras características y ventajas de la invención se irán poniendo de manifiesto conforme se examine la descripción detallada que sigue, y los dibujos que se acompañan, en los cuales:

50 la figura 1 es un esquema que representa funcionalmente los diferentes elementos de un dispositivo de tratamiento de datos según la invención, de manera global,

las figuras 2 a 4 son esquemas que representan funcionalmente unos detalles de elementos de la figura 1,

las figuras 5 y 6 son unos organigramas que ilustran el funcionamiento de un elemento de la figura 1,

	la figura 7	es un organigrama que detalla la operación 506 de la figura 5,
	la figura 8	es un organigrama que detalla la operación 508 de la figura 8,
	la figura 8A	es un organigrama que detalla la operación 850 de la figura 8,
	la figura 9	es un organigrama que detalla la operación 900 de la figura 7 y de la figura 8A,
5	las figuras 10A, 10B y 10C	son unos organigramas que detallan la operación 1000 de la figura 9 en diferentes casos,
	la figura 11	es un organigrama que ilustra el funcionamiento de un elemento del dispositivo de la figura 1,
10	la figura 12	es un esquema que ilustra una interfaz gráfica de usuario de otro elemento de la figura 1,
	la figura 13	es un organigrama que ilustra el funcionamiento de otro elemento de la figura 1, y
	la figura 14	es un esquema que ilustra un ejemplo de puesta en práctica del dispositivo de tratamiento de datos según la invención.

Además:

- 15
- el anexo 1 es un glosario,
 - los anexos 2 y 3 transcriben fórmulas matemáticas en diferentes formas, utilizadas para la descripción, y
 - el anexo 4 detalla la noción de función de resumen (hash).

20 Los dibujos y los anexos a la descripción comprenden, en lo fundamental, elementos de índole inequívoca. Por lo tanto, podrán no sólo servir para hacer comprender mejor la descripción, sino también contribuir a la definición de la invención, si es el caso.

25 La presente patente puede contener elementos susceptibles de una protección mediante derechos de autor o copyright. El titular de los derechos no tiene objeción a que cualquier persona reproduzca idénticamente este documento de patente, tal y como aparece en los expedientes y/o publicaciones de las oficinas de patentes. Por el contrario, en todo lo demás se reserva la totalidad de sus derechos de autor y/o de copyright.

30 La figura 1 representa un dispositivo de tratamiento de datos según la invención. El dispositivo de tratamiento de datos comprende una base de datos DB 100 almacenada en la memoria de una unidad central de proceso de ordenador, no representada en la figura 1, y un programa de tratamiento de los datos DPP 200 ejecutable en la memoria de una unidad central de proceso de ordenador, no representada. Por ejemplo, la base de datos DB 100 y el programa de tratamiento de los datos DPP 200 pueden estar almacenados en la misma memoria de unidad central de proceso de ordenador. En el anexo A.1.1 se ofrece una definición del término "base de datos".

35 El programa de tratamiento de los datos DPP 200 es capaz de interactuar con la base de datos DB 100 mediante medios conocidos por el experto en la materia para realizar acciones informáticas convencionales. Estas acciones comprenden operaciones sobre los datos contenidos en la base de datos DB 100 como son la lectura, la escritura, la búsqueda y la actualización y acciones sobre la estructura de la base de datos DB 100 como son la creación, la eliminación de tablas, etc. En el anexo A.1.2 se aporta una definición del término "tabla".

La base de datos DB 100 contiene datos que van a tratarse DAT 110. Los datos que van a tratarse DAT 110 se almacenan de manera organizada en una pluralidad de tablas contenedoras cada una de ellas de líneas y de columnas según se indica en el anexo A.1.2.

40 En la base de datos DB 100, las columnas pueden estar caracterizadas como columna de clave o columna de tipo C, columna técnica o columna de tipo T, columna cuantitativa o columna de tipo Q, o aún como columna de estado o columna de tipo E.

45 En el anexo A.1.10 se ofrece una definición de la expresión "columna de clave". Las columnas técnicas, o columnas T, contienen datos útiles para el funcionamiento del dispositivo según la invención. Por ejemplo, una columna técnica particular puede contener un identificador de un procesador que interviene en el tratamiento de los datos que han de tratarse DAT 110. Las columnas cuantitativas, o columnas Q, contienen datos numéricos que representan cantidades útiles para un usuario del dispositivo según la invención. Las columnas de estado, o columnas E, contienen datos alfanuméricos útiles para un usuario del dispositivo.

El programa de tratamiento de los datos DPP 200 comprende un editor de expresiones matemáticas MathEdt

210 que permite a un usuario componer expresiones matemáticas en lenguaje natural. La composición puede llevarse a cabo por intermedio de una interfaz gráfica de usuario, no representada. En el anexo A.1.3 se aporta una definición de la expresión "lenguaje natural matemático". En concreto, el editor de expresiones matemáticas MathEdt 210 permite la composición de fórmulas matemáticas que relacionan variables, operaciones y funciones matemáticas. Además, el editor de expresiones matemáticas MathEdt 210 permite la composición en lenguaje natural de condiciones matemáticas sobre estas variables. En el anexo A.1.4 se ofrece una definición del término "variable". El anexo A.2.1 ilustra, por ejemplo, una fórmula matemática expresada en lenguaje natural matemático.

En el dispositivo según la invención, las variables que participan en las fórmulas matemáticas compuestas en el editor de expresiones matemáticas MathEdt 210 apuntan a datos que han de tratarse DAT 110. Así, una de las funcionalidades del dispositivo según la invención es la de permitir la realización de un tratamiento sobre datos de la base de datos DAB 100, pudiendo dicho tratamiento ser expresado en un lenguaje matemático natural.

En lenguaje natural matemático, una operación está representada mediante un operador, una variable mediante un símbolo de variable y una función, mediante un símbolo de función. En los anexos A.1.5, A.1.6 y A.1.7 se aporta una definición de los términos "operador", "símbolo de variable" y "símbolo de función", respectivamente. Así, una fórmula matemática compuesta por medio del editor de expresiones matemáticas MathEdt 210 no contiene más que símbolos de variables, símbolos de funciones y de operadores. El anexo A.1.6 expone además una convención relativa a los símbolos de variables. La convención del anexo A.1.6 es propia del dispositivo según la invención. El anexo A.2.3 ilustra por ejemplo un símbolo de variable conforme a las convenciones del anexo A.1.6.

Además, el editor de expresiones matemáticas MathEdt210 permite la composición de condiciones de ejecución de una fórmula matemática compuesta con ayuda de operadores de condiciones, referente por ejemplo a símbolos de variables. En el anexo A.1.15 se ofrece una lista de los operadores de condiciones que se pueden asociar a una fórmula matemática.

El editor de expresiones matemáticas MathEdt 210 es un editor de metalenguaje capaz de generar una representación de una expresión matemática que, compuesta en lenguaje natural matemático, obedece a unas reglas formales. Por ejemplo, el editor de expresiones matemáticas MathEdt 210 puede ser un editor MathML 2.0 capaz de generar expresiones matemáticas conformes al estándar MathML 2.0. En el anexo A.1.8 se aporta una definición del lenguaje MathML 2.0. Un editor de este tipo es conocido por el experto en la materia. Es conocido, por ejemplo, el soporte lógico WEB EQ comercializado por la compañía DESIGN SCIENCE. El anexo A.2.2 ilustra la fórmula del anexo A.2.1 expresada en metalenguaje MathML 2.0 y generada mediante el editor de expresiones matemáticas MathEdt 210.

El programa de tratamiento de los datos DPP 200 comprende además un compilador de expresiones matemáticas MathComp 220 capaz de establecer un árbol de compilación representativo de una expresión matemática en metalenguaje, especialmente en MathML 2.0. En el anexo A.1.9 se ofrecen diferentes definiciones de la expresión "árbol de compilación".

El compilador de expresiones matemáticas MathComp 220 es capaz de establecer un árbol de compilación que contiene tan sólo operaciones identificadas, símbolos identificados, es decir, variables y condiciones. Más adelante en esta memoria descriptiva se atenderá con mayor detalle al compilador de expresiones matemáticas MathComp 220.

El programa de tratamiento de los datos DPP 200 comprende además un generador de instrucciones ComGen 230 capaz de establecer, a partir de un árbol de compilación que contiene operaciones identificadas, variables y condiciones, una lista de tablas afectadas y uniones entre dichas tablas afectadas y datos que han de tratarse DAT 110. Además, el generador de instrucciones ComGen 230 es capaz de generar una serie de instrucciones las cuales, correspondiendo al árbol de compilación, el motor (no representado) de la base de datos DB 100 es capaz de ejecutar. El generador de instrucciones ComGen 230 es, por ejemplo, un generador de instrucciones en lenguaje SQL.

La base de datos DB 100 comprende un diccionario de variables VarDic 120 capaz de mantener una relación entre un identificador de variable propio de cada variable y un nexo hacia columnas de clave de los datos que han de tratarse DAT 110, cuya combinación permite definir de manera única esta variable. En el anexo A.1.10 se ofrece una definición del término columna de clave. En este párrafo, el término variable se entiende en el sentido definido en el anexo A.1.4.2. Opcionalmente, el diccionario de variables VarDic 120 es capaz de precisar, para cada variable, si se trata o no de una variable clave.

La base de datos DB 100 comprende además un diccionario de símbolos de variables VarSymbDic 130 capaz de mantener, al menos para algunas de las variables, una relación entre estas variables del diccionario de variables VarDic120 y símbolos de variables. El diccionario de símbolos de variables VarSymbDic 130 memoriza, para cada variable de interés, una representación en metalenguaje, por ejemplo en MathML 2.0, de un símbolo de variable descompuesto según se describe a continuación.

De acuerdo con la convención expuesta en el anexo A.1.6, un símbolo de variable comprende visualmente una primera serie de caracteres determinantes de nombre de variable, una segunda serie de caracteres dispuesta como exponente y una tercera serie de caracteres dispuesta como índice. Las series de caracteres segunda y tercera son

opcionales. La tercera serie de caracteres puede comprender varias series de caracteres separadas entre sí por el carácter coma ",", determinando entonces cada serie de caracteres un índice. El anexo A.2.3 muestra un símbolo de variable conforme a la convención del anexo A.1.6.

5 Además, la base de datos DB 110 comprende un diccionario de índices IndDic 140, capaz de mantener, para cada variable de interés, una relación entre el identificador de variable y una lista de índices asociados. El diccionario de índices IndDic 140 mantiene, para cada uno de los índices asociados a una variable particular, un nexo entre el índice y una de las columnas de clave de los datos que han de tratarse DAT 110 que definen esta variable.

10 La figura 2 ilustra de manera lógica el almacenamiento de los símbolos de variables en el diccionario de símbolos de variables VarSymbDic 130. Una línea memoriza un símbolo asociado a una variable almacenando en una primera columna la primera cadena de caracteres (casilla *nm*), la segunda serie de caracteres en una segunda columna (casilla *exp*) y cada una de las cadenas de caracteres de la tercera cadena de caracteres en columnas distintas (casillas *ind1*, *ind2* y siguientes). El diccionario de variables VarSymbDic 130 mantiene asimismo un nexo entre un identificador de variable (casilla *ldVar*) asociado a un símbolo de variable y la zona del diccionario de variables VarDic 120 relativa a esta variable.

15 El anexo A.2.4 ilustra la expresión del símbolo de variable del anexo A.2.3 en MathML 2.0. En el anexo A.2.4.1, se aprecia la etiqueta <msubsup> que introduce un símbolo con índice y exponente. El nombre de la variable, el índice y el exponente se introducen cada uno de ellos mediante etiquetas <mrow> de igual nivel jerárquico, siendo este nivel inmediatamente inferior al de la etiqueta <msubsup> como muestran respectivamente los anexos A.2.4.2, A.2.4.3 y A.2.4.4. Se hace notar que cada serie de caracteres está encuadrada por el par de etiquetas <mi> y </mi>.

20 Como indica la figura 1, la base de datos DB 100 comprende un diccionario de los símbolos de función FctDic 150 que mantiene una lista de funciones matemáticas que es posible poner en práctica, en particular sobre las variables. Cada función está memorizada con relación a un símbolo de función expresado en metalenguaje, por ejemplo en MathML 2.0. En lenguaje MathML 2.0, los símbolos de función están marcados mediante unas etiquetas específicas. Por ejemplo, la etiqueta <msqrt> representa la función raíz cuadrada.

25 La base de datos DB 100 comprende además un diccionario de las operaciones OpDic 160 que mantiene una relación entre operadores y operaciones. En lenguaje MathML 2.0, un operador se introduce mediante un par de etiquetas <mo></mo>. Por ejemplo, <mo> + </mo> introduce el operador de la suma.

La base de datos DB 100 comprende finalmente un diccionario de los símbolos reservados ResSymb 170 por el dispositivo. Los símbolos reservados no aparecen en ningún otro diccionario de símbolos.

30 La figura 3 especifica la estructura del diccionario de símbolos de variables VarSymbDic 130. El diccionario de símbolos de variables VarSymDic 130 comprende una tabla de los símbolos simples SVar 132. Un símbolo simple es un símbolo sin índices ni exponentes. En tal caso, sólo se memoriza la primera cadena de caracteres (casilla *nm*). El diccionario de símbolos de variables VarSymbDic 130 comprende asimismo una tabla de los símbolos complejos CVar 134. Un símbolo complejo es un símbolo que presenta uno o varios índices y/o un exponente.

35 La figura 4 especifica la estructura del diccionario de símbolos de funciones FctDic 150. El diccionario de símbolos de funciones FctDic 150 comprende una tabla de las funciones de cálculo CalcFct 152 y una tabla de las funciones agregativas AgrFct 154. En los anexos A.1.11 y A.1.12 se ofrece una definición de las expresiones "función de cálculo" y "función agregativa", respectivamente.

40 El programa de tratamiento de los datos DPP 200 comprende además un editor de condiciones CondEdt 240 capaz de interaccionar con el diccionario de variables VarDic 120, el diccionario de índices IndDic 140 y el generador de código ComGen 230. El editor de condiciones es capaz de asociar con una variable contenida en el diccionario de variables VarDic 120, condiciones sobre el valor de sus columnas de clave. De la misma manera, el editor de condiciones CondEdt 240 es capaz de asociar condiciones con un índice contenido en el diccionario de índices IndDic 140. Las condiciones son memorizadas en un diccionario de condiciones sobre variables VarConDic 180 de la base de datos DB 100. Preferentemente, el editor de condiciones CondEdt 240 es un editor de tipo gráfico capaz de presentar, para una variable seleccionada, la lista de las columnas de clave que definen esa variable. Ventajosamente, el editor de condiciones CondEdt 240 es capaz de interaccionar, en tal caso, con el editor de expresiones matemáticas MathEdt 210 para permitir la selección de una variable directamente a partir de este editor de expresiones matemáticas.

50 El organigrama de la figura 5 detalla el funcionamiento del compilador de expresiones matemáticas MathComp 220 para una expresión matemática del tipo de fórmula compuesta en el editor de expresiones matemáticas MathEdt 210. Una fórmula puede ser expresada en lenguaje natural matemático y en forma genérica mediante la ecuación $Y = F(X)$. El anexo A.3.1 aporta un ejemplo de fórmula matemática expresada en lenguaje natural matemático. A la variable simbolizada con Y se le llama variable de salida, en tanto que a la variable simbolizada con X se le llama variable de entrada. Una fórmula matemática puede incorporar varias variables de entrada.

55 En la operación 500, el compilador de expresiones matemáticas MathComp 220 carga en la memoria los diferentes diccionarios de símbolos, es decir, el diccionario de símbolos de variables VarSymbDic 130, el diccionario de

símbolos de funciones FctDic 150, el diccionario de operadores OpDic 160 y el diccionario de símbolos reservados ResSymbDic 170.

5 En la operación 502, el compilador de expresiones matemáticas MathComp 220 carga la expresión en lenguaje MathML 2.0 de la fórmula $Y = F(X)$. El anexo A.3.2 ilustra la expresión en MathML 2.0 de la fórmula del anexo A.3.1 tal y como es cargada por el compilador de expresiones matemáticas MathComp 220.

La expresión es cargada por mediación de un lector de metalenguaje XMLRdr. El lector de metalenguaje XMLRdr es capaz de establecer una representación simplificada en forma de árbol de una expresión acorde con el estándar XML. En una tal representación, los nodos del árbol son las etiquetas XML que componen la expresión. Cada etiqueta presenta como argumento una cadena de caracteres.

10 Así, el lector de metalenguaje XMLRdr es capaz de establecer una representación en árbol XML de una expresión matemática en MathML 2.0. En el anexo A.3.3 se ilustra un ejemplo de tal representación para el caso de la fórmula del anexo A.3.1. Cada nodo corresponde a una etiqueta MathML 2.0. Se hace notar que cada nodo presenta un nivel jerárquico.

15 En la operación 504, el compilador de expresiones matemáticas MathComp 220 llama a una función de división DivFct. La función de división es capaz de localizar el operador "=" de la expresión $Y = F(X)$ en el árbol XML mediante la búsqueda de la expresión mo ("="). Opcionalmente, la función división DivFct comprueba la presencia de un único operador de este tipo en el árbol XML. La función división DivFct escinde el árbol en una parte salida A y una parte entrada B respectivamente correspondientes a las partes del árbol MathML situadas por encima y por debajo de la expresión mo ("="). Los anexos A.3.3.1, A.3.3.2 y A.3.3.3 ilustran esta división. El anexo A.3.3.1 ilustra la parte A, en tanto que el anexo A.3.3.3 ilustra la parte B. Se hace notar que una fórmula matemática podría estar definida asimismo bajo la forma $F(X) = Y$, sin que ello cambie básicamente el funcionamiento del compilador de expresiones matemáticas MathComp 220.

25 El compilador de expresiones matemáticas MathComp 220 empieza el tratamiento llamando para la parte A una función de identificación IndFct a la operación 506. La función de identificación IndFct, cuyo funcionamiento se describirá a continuación con mayor detalle, es capaz de identificar el símbolo de variable representado por la parte A.

El compilador de expresiones matemáticas MathComp 220 prosigue el tratamiento del árbol mediante la llamada, de la función de identificación IndFct para la parte B a la operación 508. La función IndFct es capaz de identificar los símbolos de variables contenidos en la parte B así como los símbolos de funciones y los operadores.

30 En la operación 510, el compilador de expresiones matemáticas MathComp 220 genera el árbol de compilación a partir de los símbolos de variables y de funciones y de los operadores. El anexo A.3.4 muestra el árbol de compilación correspondiente a la fórmula del anexo A.3.1.

La figura 6 ilustra en detalle la operación 506 de la figura 5. En la operación 600, la función de identificación IndFct llama a una subfunción de lectura de árbol TRdr la cual devuelve la primera etiqueta N1 de nivel 1 al árbol. En la operación 602, se prueba si esta primera etiqueta N1 es de tipo mi.

35 - Si es sí, entonces la parte A representa un símbolo simple en el sentido anteriormente expuesto. Se llama a una función de localización LocFct con parámetros:

- el argumento *arg_mi* de la etiqueta mi, y
- un identificador de la tabla en la cual localizar *arg_mi*, es decir, de la tabla de los símbolos simples SVar 132.

40 La función de localización LocFct devuelve la dirección dentro de los datos que han de tratarse DAT 110 asociada a la línea que contiene *arg_mi* en la *mn* (operación 604) y dada por el diccionario de variables VarDic120.

- Si es no, la primera etiqueta encontrada N1 es del tipo msub, msup o msupsub, es decir, la parte A representa un símbolo complejo. La función de lectura TRdr es llamada entonces para devolver la etiqueta de nivel jerárquico inferior N2 inmediatamente siguiente al nodo N1 (operación 606).

45 En la operación 608, se prueba si el nodo N2 es de tipo mi.

- Si es no, hay un error (operación 610).
- Si es sí, (operación 612), se llama a una función de búsqueda de ocurrencia OccVar teniendo como parámetros el argumento *arg_mi* de la etiqueta mi y un identificador de la tabla en la cual localizar *arg_mi*, es decir, de la tabla de los símbolos complejos CVar 134. La función de búsqueda de ocurrencia OccVar devuelve una lista L1 constituida a partir de la parte de la tabla de los símbolos complejos CVar 134 cuya casilla *nm* contiene *arg_mi*.

50

En la operación 614, se llama a una función comparación CompFct que compara cada elemento de la lista L1 con la continuación del árbol con el fin de identificar la variable designada.

5 La figura 7 es un organigrama que ilustra el tratamiento mediante el compilador de expresiones matemáticas MathComp 220 de la parte A de una ecuación obtenida a la salida del editor de fórmulas matemáticas MathEdt 210 y después de la lectura por parte del lector de metalenguaje XMLRdr. En otras palabras, la figura 7 detalla la operación 506 del organigrama de la figura 5.

El programa de compilación de expresiones matemáticas Mathcomp 220 comprende una lista de etiquetas llamadas "frontera" que es capaz de leer de manera recurrente en el árbol correspondiente a la parte A. La lista de las etiquetas frontera comprende, en particular, las etiquetas MathML 2.0 siguientes:

- 10
- mrow,
 - mi,
 - mo,
 - mn,
 - msub, msup y msubsup.

15 La lectura del árbol por el compilador de expresiones matemáticas MathComp 220 se lleva a cabo de manera recurrente leyendo cada uno de los nodos de un mismo nivel jerárquico desde arriba hacia abajo del árbol y comenzando nuevamente después por los nodos de nivel jerárquico inferior.

20 El compilador de expresiones matemáticas MathComp 220 lee la primera etiqueta frontera del árbol A. En la operación 700, se prueba si esta primera etiqueta frontera leída es una etiqueta de tipo mo. En lenguaje MathML 2.0, una etiqueta de tipo mo define un operador matemático. Si es sí, en la operación 702, se emite una señal de error, pues la parte A corresponde a una variable de salida sola y, por tanto, no admite símbolos de operación en ese nivel jerárquico. Si es no, en la operación 704, se prueba si esta primera etiqueta frontera es de tipo mi. Una etiqueta de tipo mi en lenguaje MathML 2.0 define un identificador, es decir, una cadena de caracteres sin significado matemático particular.

25 Si es sí, se comprueba, que el argumento de esta etiqueta mi pertenece al diccionario de símbolos simples SVar 132 en la operación 706. Más precisamente, se busca este argumento en las casillas *nm* del diccionario de símbolos simples SVar 132.

30 Si es sí, el símbolo simple identificado por la etiqueta mi está definido completamente (es decir, existe un nexo entre este símbolo simple y la variable del diccionario de variables VarDic 130 con la cual se corresponde y, por lo tanto, hacia los datos de los correspondientes datos que han de tratarse DAT 110). El tratamiento prosigue entonces en 708. Si es no, el argumento de esta etiqueta mi puede ser añadido a la tabla de los símbolos simples SVar 132, en la operación 710, y creada una variable correspondiente.

35 Si en la operación 704, la primera etiqueta frontera leída no es de tipo mi, entonces se prueba, en la operación 712, si esta etiqueta es de tipo msub, msup o msubsup. En otras palabras, se prueba si la etiqueta leída define un símbolo con sólo índice, con sólo un exponente, o con un índice y un exponente, respectivamente. Si es no, se emite un error en la operación 714, pues en ese lugar de la ecuación no se ha leído ninguna de las etiquetas permitidas. Si se ha encontrado en la operación 712 una etiqueta de tipo msub, msup o msubsup, entonces el tratamiento prosigue en la operación 900, según se describirá más adelante. Se trata entonces de un símbolo complejo en el sentido anteriormente expuesto.

40 La figura 8 es un organigrama que ilustra el tratamiento de la parte B de una ecuación matemática obtenida a la salida del editor de fórmulas matemáticas MathEdt 210 mediante el compilador de expresiones matemáticas MathComp220 y después de la lectura por parte del lector de metalenguaje XMLRdr. En otras palabras, la figura 8 detalla la operación 508 del organigrama de la figura 5. El compilador de expresiones matemáticas MathComp 220 lee de manera recurrente cada etiqueta frontera, primero según el nivel jerárquico y luego de arriba abajo.

45 En la operación 800, se prueba si la primera etiqueta frontera leída es de tipo mo. Si es sí, esta etiqueta define un símbolo de operador asociado a una función que tiene que ser identificada. Si es no, esta etiqueta es una etiqueta de tipo mi, o bien es una etiqueta de tipo msub, msup o msubsup. En estos dos casos, esto nos sitúa de nuevo en el tratamiento de etiquetas de tipo msub, msup o msubsup tal y como se ha descrito anteriormente y se ha ilustrado mediante la figura 7 (operaciones 704 y siguientes).

50 En el caso en que la primera etiqueta frontera leída es una etiqueta de tipo mo, en la operación 802, se comprueba si el argumento de esa etiqueta está contenido en el diccionario de funciones agregativas AgrFct 154.

Si es sí, se realiza un tratamiento específico, el cual comienza en la operación 850 del organigrama de la figura 8A. Este tratamiento será descrito más adelante.

5 Si es no, se comprueba en la operación 804, que el argumento de la etiqueta *mo* está contenido en el diccionario de funciones de cálculo CalcFct 152. Si es sí, la función es identificada y el tratamiento prosigue en 806. Si es no, en la operación 808, se comprueba si el argumento de esta etiqueta pertenece al diccionario de símbolos reservados ResSymbDic 170. Si es sí, la función es identificada completamente y el tratamiento prosigue en 810. Si es no, se emite un mensaje de error en la operación 812.

10 La figura 8A es un organigrama que ilustra el tratamiento específico puesto en práctica cuando en la operación 802 se ha leído un símbolo de función agregativa. El tratamiento comienza en la operación 850 mediante la extracción del subárbol al que concierne el símbolo de función agregativa. Este subárbol está constituido a partir del conjunto de las etiquetas situadas bajo la etiqueta de tipo *mo* y de nivel jerárquico inferior o igual. En lo sucesivo, tal árbol será denominado parte C.

15 En la operación 852, se lee la primera etiqueta *mi* inmediatamente siguiente a la etiqueta de tipo *mo* considerada en la operación 802 (en el sentido de una lectura de arriba abajo sin distinción de niveles jerárquicos). El argumento de esta etiqueta *mi* introduce una cadena de caracteres conformantes del nombre de un índice de agregación. El nombre de este índice de agregación se almacena en una tabla de índices temporal IndTmp en una casilla *nm*.

20 En la operación 854, se extraen las condiciones sobre este índice. Las condiciones sobre este índice son introducidas mediante etiquetas de tipo *mo*. Las condiciones se almacenan en la tabla de índices temporal IndTmp, no representada, asociadas al nombre del índice *nm*.

25 Se leen las etiquetas de tipo *msub*, *msup*, o *msubsup* inmediatamente siguientes al símbolo de agregación descubierto en la operación 802. Estas etiquetas definen unos símbolos de variables complejas en los que puede centrarse la función agregativa. Estas etiquetas son objeto de un tratamiento específico el cual será descrito más adelante. Este tratamiento comienza con una operación 900 ilustrada en la figura 9. Este tratamiento está destinado a identificar las variables afectadas.

La figura 9 es un organigrama que ilustra el tratamiento de símbolos de variables complejas, es decir, que comprenden índices *y/o* exponentes. En otras palabras, la figura 9 expone el tratamiento de las etiquetas de tipo *msub*, *msup* y *msubsup* mediante el compilador de expresiones matemáticas MathComp 220.

30 En el caso en que es leída una etiqueta de este tipo, por ejemplo en la operación 712 del organigrama de la figura 7 o en la operación 856, éste es el tratamiento que se pone en práctica. En la operación 900, se lee la primera frontera de tipo *mi* o de tipo *mo*. La lectura se lleva a cabo entonces siguiendo la sucesión de las etiquetas, independientemente de los niveles jerárquicos.

35 Si la primera etiqueta leída es de tipo *mo* (operación 902), se comprueba entonces, en la operación 904, si el argumento de esta etiqueta es un paréntesis abierto "(". Si es sí, se trata ya no de una variable sino de la función exponencial, la cual queda entonces identificada (operación 906). Si es no, se emite un mensaje de error en la operación 908.

40 Si la primera etiqueta encontrada no es de tipo *mo* (operación 902), entonces la primera etiqueta encontrada es de tipo *mi* (operación 910). Se comprueba entonces que el argumento de esta etiqueta está contenido en el diccionario de símbolos complejos CVar 134, todavía en la operación 910. Más precisamente, se busca el argumento de esta etiqueta en las casillas *nm* del diccionario de símbolos complejos CVar. Si es no, se prevé, en la operación 912, la eventual inclusión de este argumento en la tabla de los símbolos complejos Cvar 134 bajo una casilla *mn* y la creación de una nueva variable en el diccionario de variables VarDic 120.

45 En el caso contrario, se emite en la operación 914 una lista de las variables contenidas en el diccionario de símbolos complejos CVar 134 cuyo nombre, almacenado bajo la casilla *nm*, equivale al argumento de la etiqueta *mi*. Esta lista emitida contiene, para cada símbolo de variable, el conjunto de las cadenas de caracteres conformantes del exponente *exp y/o* de los índices *ind1*, *ind2*, etc.

En la operación 916, se comprueba, para cada variable emitida, si la sucesión de las etiquetas MathML (tipos y argumentos en identidad) de esta variable es acorde con la sucesión de las etiquetas leídas en la porción de árbol en curso de lectura.

50 Si en el árbol leído no se da con ninguna variable emitida, entonces, en la operación 918, se emite un error. Se puede igualmente prever, por ejemplo, que en la operación 918 el compilador de expresiones matemáticas MathComp 220 almacene por ejemplo la parte de la fórmula no hallada en el diccionario de símbolos complejos CVar en metalenguaje. En este caso, no son emitidos los mensajes de error de la operación 918. Ello permite definir con posterioridad la variable correspondiente al símbolo de que se trate.

5 Si se ha hallado la sucesión de las etiquetas MathML para una variable emitida, entonces, en la operación 920, se comprueba si en el árbol están definidos índices suplementarios (es decir, no almacenados con la variable en el diccionario de variables complejas CVar 134). Si es sí, se pone en práctica un tratamiento particular, el cual comienza con una operación 1000 y será descrito más adelante. Si es no, la variable es identificada completamente y los datos de los datos asociados que han de tratarse DAT 110 son localizados por mediación del diccionario de variables VarDic 130.

10 La figura 10A especifica el tratamiento realizado después de la operación 920 de la figura 9 en el caso en que se trata una parte A. Si se trata la parte A y en el diccionario de símbolos de variables VarSymbDic130 no se presentan índices, entonces el programa propone al usuario añadir el índice al diccionario de índices IndDic en la operación 1000A.

15 La figura 10B especifica el tratamiento realizado después de la operación 920 de la figura 9 en el caso en que se trata una parte B. Si se trata la parte B y en el diccionario de símbolos de variables VarSymbDic 130 no se presentan índices, entonces el programa comprueba, en la operación 1000B, si el índice está contenido en la tabla de los índices temporales IndTmp. Si es sí, el índice está presente en el símbolo de la variable de entrada y, por lo tanto, el índice es identificado (operación 1002B). Si es no, el programa emite un error en la operación 1004B.

20 La figura 10C especifica el tratamiento realizado después de la operación 920 de la figura 9 en el caso en que se trata una parte C. Si se trata la parte C y en el diccionario de símbolos de variables VarSymbDic130 no se presentan índices, entonces el programa comprueba, en la operación 1000C, si el índice está contenido en la tabla de los índices temporales IndTmp. Si es sí, el índice está presente ya sea en el símbolo de la variable de entrada, o bien presente en el operador de agregación. En estos dos casos, el índice es identificado y se añade el índice al diccionario de índices IndDic 140 en la operación 1002C. Si es no, el programa emite un error en la operación 1004C.

25 Las pruebas de las operaciones 1000B y 1000C y los mensajes de error de las operaciones 1004B y 1004C se encargan de que, por ejemplo, una variable A a la cual se define como dependiente de un índice i sea designada A_i . Esto podría generalizarse a variables con varios índices y/o exponentes. Estas operaciones garantizan, entre otras cosas, una claridad en la expresión formal de una fórmula matemática, en particular cuando la misma define una variable.

Por ejemplo, se podría admitir la notación A para una variable que depende de i, es decir, el índice i sería añadido al diccionario de índices IndDic y se podría omitir la emisión de mensajes de error de las operaciones 1004B y 1004C.

30 Por ejemplo todavía, no se añadiría el índice i al diccionario de índices IndDic, pero se omitiría la emisión de mensajes de error de las operaciones 1004B y 1004C. En estos ejemplos, el compilador de expresiones matemáticas MathComp 220 estaría dispuesto para conservar una expresión de la parte de la fórmula afectada por el índice i en metalenguaje bajo forma de árbol. El compilador de expresiones matemáticas, cuando trata una fórmula más general que incluye la fórmula anterior, integra el árbol anteriormente memorizado en el árbol que está construyendo.

35 Se comprende que el diccionario de índices IndDic puede no mantener una asociación entre el índice y valores numéricos de la base de datos DB 100, al menos en un primer tiempo. El diccionario de índices IndDic es entonces un diccionario abierto que puede ser completado con posterioridad, por ejemplo cuando el compilador de expresiones matemáticas MathComp 220 opera sobre una fórmula que precisa los valores numéricos del índice de que se trate. El compilador de expresiones matemáticas MathComp 220 es capaz de reaccionar a la ausencia de valores definidos del índice memorizando la parte de la fórmula afectada en metalenguaje.

40 Como ejemplo, se considera una variable A, definida en función de un índice i (por ejemplo, $A_i = 1 + i$). El índice i será memorizado en el diccionario de índices IndDic y en el diccionario de símbolos de variables VarSymbDic. Aunque el compilador de expresiones matemáticas MathComp 220 ha acabado el tratamiento de la fórmula definitoria de A_i , no existe asociación en el diccionario de índices IndDic entre el índice i y los valores numéricos. Se considera ahora que el compilador de expresiones matemáticas MathComp 220 trata una fórmula que incluye A_i y que precisa i (por ejemplo,

45 $B = \sum_{f=1}^{f=10} A_f$). El compilador de expresiones matemáticas MathComp 220 completa la definición de i en el diccionario de índices IndDic asociando el índice a los valores numéricos presentes en esta fórmula.

50 Ventajosamente, las operaciones anteriormente expuestas de inserción y de búsqueda en tablas se ponen en práctica mediante el procedimiento de obtención de un resumen, el cual permite reducir el tiempo de búsqueda y de inserción de datos en las tablas. El procedimiento de obtención de un resumen es conocido por el experto en la materia y se expone de manera sucinta en el anexo 4. Las variantes de resumen mediante direccionamiento abierto y de resumen mediante concatenación, expuestas respectivamente en los anexos 4.3 y 4.4, pueden ser empleadas en el dispositivo según la invención. Preferentemente, se implanta un resumen mediante concatenación.

55 La figura 11 ilustra en forma de organigrama el funcionamiento del generador de instrucciones ComGen 230. En la operación 1100, el generador de instrucciones ComGen 230 recupera el árbol de compilación del compilador de expresiones matemáticas MathComp 220. En la operación 110, el generador de instrucciones ComGen 230 comienza

5 un tratamiento particular que se realiza sobre cada una de las variables identificadas, denominadas de manera general S. El generador de instrucciones ComGen 230 interacciona con el editor de condiciones CondEdt 240 para conocer las condiciones asociadas a la variable S, en el transcurso de la operación 1120. Si la variable S no lleva asociada ninguna condición, entonces el tratamiento se realiza para la variable siguiente. Si es no, en la operación 1130, las condiciones son transformadas en cláusulas "where".

Paralelamente, el generador de instrucciones ComGen 230 realiza las uniones a partir de las restricciones estructurales y de las restricciones lógicas, derivadas de las relaciones entre las tablas, definidas sobre la estructura de la base, las cuales son equivalentes a los índices implícitos, y los índices explícitos, en el transcurso de la operación 1140.

10 En la operación 1150, el generador de instrucciones establece una serie de instrucciones, o código, aptas para ser interpretadas por el motor de la base de datos DB 100. Una vez identificados los pares tablas-variantes, el generador de instrucciones ComGen 230 puede crear eventualmente una o varias tablas temporales de cálculo CalcTmp 195. En particular, estas tablas temporales de cálculo CalcTmp 195 se crean cuando el generador de instrucciones ComGen 230 encuentra en una expresión (es decir, un conjunto de nodos del árbol de compilación de igual nivel jerárquico), una operación demasiado compleja para poder ser efectuada por medio de una única consulta.

15 A título de ejemplo, en el caso particular de instrucciones expresadas en SQL, no se puede utilizar una expresión de tipo select "a+sum (b)", siendo por ejemplo a y b dos columnas y sum () la función suma de las líneas de una columna, ya que combina unos elementos de niveles de ensamble diferentes.

20 En particular, la presencia en una expresión de una llamada a una función que puede contener consultas, o de una función agregativa, conlleva la utilización de una tabla temporal CalcTmp 195 para permitir la descomposición de la expresión en etapas sucesivas, apuntando las tablas temporales CalcTmp sustituidas, por ejemplo, por cursores, hacia los datos afectados.

25 El programa de tratamiento de los datos DPP 200 comprende además un editor de tratamientos ProEdt 250. El editor de tratamientos ProEdt 250 es un editor gráfico que permite a un usuario definir un encadenamiento estructurado de tratamientos que han de aplicarse a unos datos de los datos que han de tratarse DAT 110. Cada uno de los tratamientos puede quedar definido en forma de una fórmula matemática compuesta por mediación del editor de fórmulas matemáticas MathEdt 210. Como ilustra la figura 12, cada tratamiento Proc1, Proc2, etc., está representado mediante una caja de tratamiento 1200, 1210, etc., respectivamente. Los tratamientos se hallan enlazados entre sí mediante flechas, las cuales definen un encadenamiento de los tratamientos. A título de ejemplo, en la figura 12, la flecha 1205 permite al usuario definir que el tratamiento Proc2 1210 tiene que ser realizado después del tratamiento Proc1 1200. Las flechas 1215 y 1225 establecen que los tratamientos Proc3 y Proc4 respectivamente han de realizarse después del tratamiento Proc2. Al tener las flechas 1215 y 1225 el mismo origen, los tratamientos Proc3 y Proc4 han de realizarse en paralelo.

35 El editor de tratamientos ProEdt 250 interacciona con un diccionario de tratamientos ProcDic 190 de la base de datos DB 100. El diccionario de tratamientos ProcDic 190 es capaz de mantener un nexo entre una representación de una fórmula matemática y un identificador de fórmula, propio de cada fórmula almacenada. De manera ventajosa, el diccionario de tratamientos ProcDic 190 puede contener asimismo fórmulas preestablecidas y reutilizables por el usuario.

40 Una fórmula matemática puede ser almacenada en el diccionario de tratamientos ProcDic 190 en forma de una representación en metalenguaje, por ejemplo MathML 2.0, proveniente por ejemplo del editor de expresiones matemáticas MathEdt 210. Además, una fórmula matemática puede ser almacenada en forma de árbol interpretado, por ejemplo por el compilador de expresiones MathComp 230, en particular con enlaces entre los símbolos de variables y las variables de los datos que han de tratarse DAT 110. Por supuesto, las dos representaciones pueden cohabitar en el diccionario de tratamientos ProcDic 190.

45 En el caso de una ejecución directa de la fórmula almacenada, prima la representación interpretada, mientras que en el caso de una ejecución indirecta (en paralelo con otra fórmula, por ejemplo), prima la representación en forma de metalenguaje.

50 El dispositivo según la invención permite la definición y el almacenamiento de una primera fórmula de forma $Y = f(X)$, donde f representa una función cualquiera, y de una segunda fórmula de forma $Z = g(Y)$, donde g representa igualmente una función cualquiera en metalenguaje. En la generación del código destinado a permitir el cálculo de Z, el generador de instrucciones ComGen 230 reestructura la fórmula en $Z = h(X)$ utilizando una definición de Y por medio de metadatos. En otras palabras, la variable Y no necesita ser almacenada en forma numérica en la base de datos DB 100. Antes bien, la variable Y se almacena en forma de un árbol de instrucciones interpretado parcialmente, estando dicho árbol de instrucciones insertado en el árbol de instrucciones resultante de la interpretación de Z en sustitución de la variable Y. El editor de tratamientos ProEdt 250 es capaz de mantener para cada caja un nexo entre una caja particular y un identificador de fórmula del diccionario de tratamientos ProcDic 190. Además, el editor de tratamientos ProEdt 250 es capaz de mantener una lista de las flechas compuestas por el usuario, estando cada una de ellas definida por un

identificador de fórmula de origen y un identificador de fórmula diana. Una caja particular se caracteriza como tratamiento de referencia a partir de la cual se puede establecer una representación del encadenamiento por medio de todas las flechas. Un sistema de cajas madres desglosables en cajas hijas permite al usuario elaborar con sencillez encadenamientos complejos de expresiones tratadas mediante el programa de tratamiento de datos DDP 200.

5 Se hace notar aquí que mediante el dispositivo según la invención se pueden poner en práctica otras acciones informáticas no definidas mediante fórmulas matemáticas, entre ellas, un tratamiento particular de cálculo de datos faltantes, que está ilustrado en la figura 13.

10 La figura 13 es un organigrama que ilustra el funcionamiento de un calculador de datos faltantes MDATCalc 260 integrado dentro del programa de tratamiento de datos DDP 200. El calculador de datos faltantes MDATCalc 260 es capaz de interaccionar con el diccionario de variables VarDic 120 para seleccionar una variable de clave, designada aquí con V, para la cual faltan valores (operación 1300). En la operación 1302, el calculador pasa una instrucción, por ejemplo una instrucción SQL, al motor de la base de datos DB 100 para realizar una clasificación creciente de la variable. En la operación 1304, se prueba si V es de tipo fecha u hora.

- Si es no, en la operación 1306, se prueba si V es de tipo alfanumérico.

15 - Si es no, se emite un error, ya que el calculador de datos faltantes no puede realizar el tratamiento (operación 1308).

- Si es sí, en la operación 1310, el calculador establece un paso P, el cual separa los datos de la variable V.

20 En la operación 1312, se define el alcance de la variable estableciendo un valor de inicio como el mínimo valor la variable V y un valor de fin como el máximo valor de la variable V. Se define y se pone al valor 0 una variable de incremento i.

En la operación 1314, se inicia un lazo cuya condición de parada es que el valor U_i que ha de calcularse sea superior al valor de fin.

En la operación 1316, se calcula el valor U_i valiendo el mínimo valor de la variable V al que se añade i veces el paso P calculado en la operación 1310.

25 En la operación 1318, se prueba si el valor de la variable V existe en U_i :

- si es sí, se guarda U_i en una lista L en la operación 1320, en 1330 se incrementa i y el lazo se reanuda en 1314,

- si es no, i es incrementado en el valor 1 en 1330 y luego el lazo se reanuda en 1314.

30 Si se determina, en la operación 1304, que la variable V es una fecha o una hora, entonces en la etapa 1332 se calcula un coeficiente medio en orden a determinar una frecuencia F. En la operación 1334, se determina el alcance de la variable V de manera análoga a la operación 1314.

35 En la etapa 1336, se inicia un lazo, análogo al lazo iniciado en 1314, cuya condición de salida es que la fecha D_i calculada sea superior al valor máximo de la variable V $\text{máx}(V)$ determinada en 1334. Mientras la fecha calculada D_i sea inferior a $\text{máx}(V)$, se calcula D_i valiendo el mínimo valor de la variable V determinada en 1334 a la cual se añade i veces la frecuencia F calculada en 1332. Seguidamente se realizan las etapas 1340, 1342 y 1334, las cuales son análogas respectivamente a las etapas 1318, 1320 y 1330.

40 En los dos casos anteriormente expuestos, la salida de los lazos iniciados en 1314 y 1336 prosigue en 1338 mediante la creación de las líneas faltantes a partir de la lista L establecida en las etapas 1320 sucesivas, o 1342. El programa se termina en 1340.

45 La figura 14 ilustra un ejemplo de puesta en práctica del dispositivo de tratamiento de los datos según la invención. El ordenador 1400 comprende una unidad central de proceso 1410, la cual integra en particular una unidad de tratamiento o UCP y la memoria de acceso aleatorio, una pantalla 1420, un teclado 1430, una memoria masiva 1440, por ejemplo de tipo disco duro, un periférico de punteo 1450, por ejemplo de tipo ratón, una impresora 1460 opcional y un periférico de acceso a la red 1470, también opcional.

La memoria masiva 1440, o cualquier otra memoria, del ordenador 1400 alberga un sistema operativo con interfaz gráfica de usuario. El sistema operativo se inicia en general, según es sabido, en el arranque del ordenador.

Además, la memoria masiva 1440 alberga el programa de tratamiento de datos DPP 200 y la base de datos DB 100. El programa de tratamiento de datos DPP 200 puede ser ejecutado por medio de la unidad de tratamiento o UCP.

El dispositivo de tratamiento de datos según la invención puede comprender un soporte lógico destinado a ser ejecutado en la memoria de una unidad central de proceso, por ejemplo la memoria masiva 1440 del ordenador 1400 de la figura 14. En particular, el programa de tratamiento de datos DPP 200 y la base de datos DAB 100 pueden constituir soportes lógicos diferenciados pero capaces de cooperar.

5 El funcionamiento del dispositivo según la invención, tal como se ha descrito anteriormente, puede ser expresado asimismo bajo forma de un procedimiento.

10 Anteriormente se ha descrito el caso de la definición de una variable cuyo índice no está determinado por completo en el momento de esta definición. Se ha explicado que, entonces, el índice quedaba totalmente definido (es decir, apuntando directa o indirectamente hacia unos valores numéricos) en una expresión matemática más general que utiliza dicha variable. Se comprende que se opera una sustitución de la variable por su árbol interpretado por el generador de instrucciones ComGen 230 cuando este último trata la fórmula general. Esto se puede generalizar a los casos de fórmulas definidas de manera incompleta (es decir, el árbol de compilación no es completamente interpretado) pero reutilizadas en expresiones que completan su definición, o incluso cuando un elemento de fórmula está ausente a priori de la base de datos pero es definido con posterioridad.

15 Los datos que han de tratarse DAT 110 pueden hallarse contenidos en una base de datos distinta de la base de datos DB 100 y relacionada con el programa de tratamiento de datos DPP 200 y con el resto de la base de datos DB 100.

20 El programa de tratamiento de datos DPP 200 puede hallarse albergado en la memoria de diferentes ordenadores del tipo ilustrado mediante la figura 14 conectados por intermedio de sus periféricos de acceso a la red 1410 a un ordenador de tipo servidor cuya memoria masiva alberga la base de datos DB 100.

El programa de tratamiento de datos DPP 200 puede hallarse albergado en la memoria de un ordenador de tipo servidor y controlado desde un programa de interfaz ejecutado en un ordenador remoto. Por ejemplo, el programa de tratamiento de datos DPP 200 puede estar dotado de medios que permiten su pilotaje desde un navegador de Internet ejecutado en un ordenador remoto.

25 La base de datos DB 100 y/o el programa de tratamiento de los datos DPP 200 pueden estar provistos de medios que permiten alimentar la base de datos DB 100 con datos que han de tratarse DAT 110, por ejemplo mediante extracción de datos desde bases de datos remotas.

30 La invención cubre igualmente al soporte lógico descrito, puesto a disposición en cualquier soporte legible por ordenador. La expresión "soporte legible por ordenador" comprende los soportes de almacenamiento de datos, magnéticos, ópticos y/o electrónicos, al igual que un soporte o vehículo de transmisión, como es una señal analógica o digital.

Anexo 1 - Glosario

A.1.1 - Base de datos

35 Todo conjunto de ficheros estructurados, susceptibles de ser interconectados mediante nexos entre estos ficheros.

A.1.2 - Tabla

40 Los elementos fundamentales de una base de datos son las "tablas". Cada tabla es un conjunto de líneas; cada línea se organiza en una o varias columnas; y cada columna corresponde a un tipo de datos. Cada línea comprende un solo y único valor (el "dato") para cada columna de la tabla. Físicamente, el almacenamiento de los datos puede corresponderse con esta estructura en tabla, o bien ser diferente.

A.1.3 - Lenguaje natural matemático

Descripción de una fórmula matemática o de condiciones matemáticas con ayuda de símbolos matemáticos convencionales, como por ejemplo:

$$\sum, +, -, \Pi, \sqrt, f(x), x^n$$

45 La convención matemática utilizada puede ser, por ejemplo, la del sistema educativo científico francés.

A.1.4 - Variable

A.1.4.1 - En sentido matemático, cantidad susceptible de cambiar de valor.

A.1.4.2 - En el sentido de una base de datos, nexo identificado hacia unos datos almacenados, es decir, hacia

una parte de uno o varios ficheros.

A.1.5 – Operador

Símbolo que representa una operación o una serie de operaciones que han de efectuarse sobre unos conceptos cualesquiera, por ejemplo lógicos, matemáticos o físicos.

5 A.1.6 - Símbolo de variable

Serie de caracteres asociada a una sola y única variable para representar esta variable en lenguaje matemático natural.

10 Por convención, un símbolo se constituye a partir de un nombre de variable en forma de una primera serie de caracteres, de un exponente en forma de una segunda serie de caracteres y de índices en forma de una tercera serie de caracteres y del carácter coma "," como índice particular. El exponente y los índices son opcionales. La primera serie se puede componer de caracteres alfanuméricos, mayúsculas o minúsculas. Los operadores están prohibidos.

La segunda serie y la tercera serie se pueden componer de todos los caracteres sin restricción. Las segunda y tercera series son opcionales.

15 Los índices van separados por el carácter coma ",". La tercera serie de caracteres se puede descomponer así en varias series de caracteres asociadas cada una de ellas a un índice.

A.1.7 - Símbolo de función

Símbolo asociado a una función matemática en lenguaje natural matemático. Por ejemplo, el símbolo de la función raíz cuadrada es $\sqrt{\quad}$.

A.1.8 - Lenguaje MathML 2.0

20 Especialización para las matemáticas del lenguaje estándar de formateo de los datos estructurados XML. El lenguaje MathML se compone, como el lenguaje XML, de «etiquetas abridoras» y «cerradoras» las cuales permiten identificar los elementos constitutivos de una expresión matemática. Existen diferentes variantes de este metalenguaje como «presentación» y «semántica». La variante «presentación» es la utilizada aquí.

A.1.9 - Árbol de compilación

25 A.1.9.1 - Árbol n-ario que contiene las operaciones que han de efectuarse de una manera jerárquica correspondiente a las reglas de prioridades matemáticas. Sea f una función, entonces el nodo f del árbol contiene los parámetros de la función f , es decir, los parámetros de la función f son subnodos del nodo f .

A.1.9.2 - Representación arborescente de una fórmula matemática constituida a partir de funciones simples, o complejas, pero estructurada para una transcripción en un lenguaje informático cualquiera.

30 A.1.9.3 - Ordenación estructurada y jerárquica de etapas de compilación. El árbol de compilación corresponde a la fase de precompilación que interpreta la secuenciación de una expresión matemática expresada en metalenguaje.

A.1.10 - Clave o columna clave

Columna de una base de datos capaz de definir sola o en combinación con otras columnas claves unívocamente una línea en un fichero de una base de datos.

35 A.1.11 - Función de cálculo

Función para la cual la dimensión del símbolo asociado a la variable de salida es inferior o igual a la dimensión de las variables a las cuales se aplica dicha función. Por ejemplo, la función logaritmo de Neper \ln es una función de cálculo.

A.1.12 - Función agregativa

40 Función para la cual la dimensión del símbolo asociado a la variable de salida es estrictamente inferior a la dimensión de las variables a las que se aplica dicha función. Por ejemplo, las funciones producto sobre el índice i , \prod y suma sobre el índice i , \sum son funciones agregativas.

A.1.13 - Dimensión de una variable

Número de variables independientes cuyo valor debe ser conocido para conocer el valor de dicha variable.

A.1.14 - Dimensión de un símbolo de variable

Número de índices asociados a un símbolo de variable. La dimensión de un símbolo de variable corresponde a la dimensión de la variable a la cual está asociada. Cada índice corresponde a una variable independiente.

A.1.15 - Operadores de condición

- 5 Los operadores *Si*, *Entonces*, *Si es no*, *Mientras*; <, >, =, ≠, O, Y y *No* son operadores de condición.

Anexo 2

A.2.1 - Fórmula expresada en lenguaje natural matemático

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

A.2.2 - Fórmula del anexo A.2.1 expresada en lenguaje MathML 2.0

```

<mrow>
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mrow>
        <mo>-</mo>
        <mi>b</mi>
      </mrow>
      <mo>&PlusMinus;</mo>
      <msqrt>
        <mrow>
          <msup>
            <mi>b</mi>
            <mn>2</mn>
          </msup>
          <mo>-</mo>
          <mrow>
            <mn>4</mn>
            <mo>&InvisibleTimes;</mo>
            <mi>a</mi>
            <mo>&InvisibleTimes;</mo>
            <mi>c</mi>
          </mrow>
        </mrow>
      </msqrt>
    </mrow>
    <mrow>
      <mn>2</mn>
      <mo>&InvisibleTimes;</mo>
      <mi>a</mi>
    </mrow>
  </mfrac>
</mrow>

```

A.2.3 - Ejemplo de un símbolo de variable

$$Dcut_{ind,j,m_k}^{31/12/n-1}$$

A.2.4 - Símbolo de variable A.2.3 en lenguaje MathML

	$\langle \text{math} \rangle$ $\langle \text{mrow} \rangle$
A.2.4.1	$\langle \text{msubsup} \rangle$
A.2.4.2	$\langle \text{mrow} \rangle$ $\langle \text{mi} \rangle Dcut \langle \text{mi} \rangle$ $\langle \text{mrow} \rangle$
A.2.4.3	$\langle \text{mrow} \rangle$ $\langle \text{mi} \rangle ind \langle \text{mi} \rangle$ $\langle \text{mo} \rangle , \langle \text{mo} \rangle$ $\langle \text{mi} \rangle j \langle \text{mi} \rangle$ $\langle \text{mo} \rangle , \langle \text{mo} \rangle$ $\langle \text{msub} \rangle$ $\langle \text{mrow} \rangle$ $\langle \text{mi} \rangle m \langle \text{mi} \rangle$ $\langle \text{mrow} \rangle$ $\langle \text{mrow} \rangle$ $\langle \text{mi} \rangle k \langle \text{mi} \rangle$ $\langle \text{mrow} \rangle$ $\langle \text{msub} \rangle$ $\langle \text{mrow} \rangle$
A.2.4.4	$\langle \text{mrow} \rangle$ $\langle \text{mi} \rangle 31/12/n-1 \langle \text{mi} \rangle$ $\langle \text{mrow} \rangle$
A.2.4.5	$\langle \text{msubsup} \rangle$ $\langle \text{mrow} \rangle$ $\langle \text{math} \rangle$

Anexo 3

A.3.1 - Fórmula en lenguaje natural matemático

$$a_i = b_i + \sum_j c_{i,j}$$

A.3.2 - Fórmula A.3.1 en MathML 2.0

```

<m:math xmlns:m='http://www.w3.org/1998/Math/MathML'>
  <m:mrow>
    <m:msub>
      <m:mi>a</m:mi>
      <m:mi>i</m:mi>
    </m:msub>
    <m:mo>=</m:mo>
    <m:mrow>
      <m:msub>
        <m:mi>b</m:mi>
        <m:mi>i</m:mi>
      </m:msub>
      <m:mo>+</m:mo>
      <m:mrow>
        <m:munder>
          <m:mo>&sum;</m:mo>
          <m:mrow>
            <m:mi>j</m:mi>
          </m:mrow>
        </m:munder>
        <m:msub>
          <m:mi>c</m:mi>
          <m:mrow>
            <m:mi>i</m:mi>
            <m:mo>,</m:mo>
            <m:mi>j</m:mi>
          </m:mrow>
        </m:msub>
      </m:mrow>
    </m:mrow>
  </m:mrow>
</m:math>

```

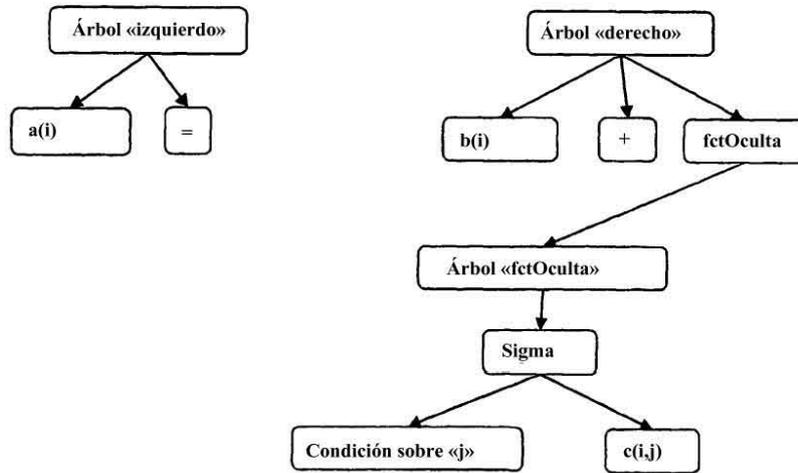
ES 2 389 363 T3

A.3.3 - Fórmula A.3.1 en árbol XML

A.3.3.1	mrow msub mi (« a ») mi (« i »)
A.3.3.2	mo (« = »)

A.3.3.3	mrow msub mi (« b ») mi (« i ») mo (« + ») mrow munder mo (« ∑ ») mrow mi (« j ») msub mi (« c ») mrow mi (« i ») mo (« , ») mi (« j »)
---------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

A.3.4 - Fórmula A.3.1 en forma de árbol de compilación



Anexo 4

A.4.1 La obtención de un resumen

5

La obtención de un resumen permite realizar operaciones de búsqueda y de inserción en una tabla en tiempo constante. Las memorias asociativas presentan esta propiedad: la posición ocupada por un objeto sólo está determinada por este objeto. Es suficiente calcular la posición en la memoria de este objeto, es decir, disponer de una función f que asocie con un objeto x una posición $h(x)$. El tiempo de cálculo de la posición $h(x)$ tiene que ser independiente de x y tan reducido como sea posible.

10

La organización de las tablas de resumen se opone radicalmente a la de las tablas ordenadas: en una tabla ordenada, la posición de un objeto depende del número de objetos más pequeños que ya han sido insertados en la tabla.

Para insertar en una tabla los elementos de un conjunto $[0, N-1]$, una solución corriente consiste en utilizar un cuadro t de dimensión N y en ordenar cada objeto i en $t[i]$ con $h(i) = i$.

15

Para tratar un conjunto cualquiera E de N enteros, es necesario construir efectivamente una función h inyectiva del conjunto E en el conjunto de los índices del cuadro $[0, N-1]$.

Para insertar en una tabla los objetos de un conjunto E de cardinal mayor, e incluso mucho mayor, que N , es necesario construir una función que, aun sin ser inyectiva, posea buenas propiedades de dispersión.

A.4.2 Función de resumen

20

La función h definida para la fórmula siguiente es un ejemplo simple de función de resumen sobre unas cadenas de longitud l con valores dentro del intervalo $[0, N-1]$.

$$h(x) = \sum_{i=0}^{l-1} x[i] B^{l-1-i} \text{ mod } N$$

donde

- B es una potencia de 2 para facilitar el cálculo, por ejemplo B vale 256, y
- N es un número primo para evitar colisiones aritméticas.

25

$h(x)$ se denomina valor de resumen asociado a la clave x . Puesto que la función de resumen h no es inyectiva, es necesario tratar las colisiones. Hay colisión cuando dos claves x e y diferentes, ($x \neq y$), tienen el mismo valor de resumen, ($h(x) = h(y)$). Los anexos A.4.3 y A.4.4 exponen dos técnicas de resolución, escogidas según que el espacio de memoria disponible sea o no ilimitado.

A.4.3 Resumen mediante direccionamiento abierto

30

La tabla de resumen se implementa como un cuadro t cuyas casillas van a contener los objetos. Inicialmente, la tabla está vacía: cada casilla contiene un objeto especial vacío, por ejemplo un objeto cuya clave es la cadena vacía "".

Se realizan operaciones de inserción de la manera siguiente:

- 5
- Si ha de insertarse un objeto de clave x y $t[h(x)]$ está vacío, entonces la inserción se lleva a cabo en este espacio.
 - Si $t[h(x)]$ ya está ocupado y el contenido tiene una clave diferente de x , entonces se calculan los valores de resumen supletorios $h_1(x), h_2(x), \dots$ hasta encontrar $t[h_i(x)]$ vacío o conteniendo un objeto de clave x .

Para generar estos valores de resumen supletorios, un procedimiento simple es el de escoger $h_i(x)$ según se indica mediante la fórmula siguiente:

10

$$h_i(x) = (h(x) + i) \bmod N$$

Este procedimiento, denominado resumen lineal, consiste en probar las casillas que siguen a $t[h(x)]$. Existen otros procedimientos, tales como el resumen cuadrático, el resumen doble como también el resumen aleatorio, que presentan mejores capacidades de dispersión.

15 Para buscar un objeto de clave x , se someten a prueba los objetos en $t[h(x)]$ y, eventualmente, en $t[h_1(x)], t[h_2(x)], \dots$, hasta que la clave del objeto que ahí se encuentre sea igual a x , o bien que el objeto esté vacío.

Cuando la tabla también permite supresiones, es necesario sustituir un objeto suprimido por un objeto especial *suprimido*, distinto del objeto vacío. Al insertar, se utiliza la primera casilla vacía o suprimida, mientras que al buscar, tan sólo se parará en la primera casilla vacía.

20 Como máximo, son necesarias N operaciones, siendo N la dimensión de la tabla. Se define el coeficiente de carga α de una tabla de resumen de dimensión N que contiene n objetos como sigue:

$$\alpha = \frac{n}{N}$$

α siempre está comprendido entre 0 y 1. Para dar una estimación asintótica de la complejidad de las operaciones, se hace que n y N tiendan al infinito, permaneciendo α constante. Se muestra que, en una hipótesis de uniformidad, el número medio de accesos necesarios para una búsqueda negativa es como mucho de:

25

$$\frac{1}{1 - \alpha},$$

y para una búsqueda positiva, de:

$$\frac{1}{\alpha} \log \frac{1}{1 - \alpha} + \frac{1}{\alpha}$$

30 Por ejemplo, para una tabla medio llena, hay que contar con llevar a cabo 2 accesos para la búsqueda de un objeto que no se encuentra en la tabla, y 3.387 accesos si se encuentra en ella. Se trata efectivamente de un algoritmo en $\Theta(1)$.

A.4.4 Resumen por concatenación

35 El procedimiento de resumen por direccionamiento abierto con resolución lineal presenta el inconveniente de efectuar las comparaciones con varias casillas sucesivas de la tabla las cuales, sin embargo, no tienen el mismo valor de resumen que la cadena buscada. Para evitar estas comparaciones inútiles, se realizan cadenas que relacionan los objetos que tienen el mismo valor de resumen.

La tabla de resumen se implementa mediante un cuadro de punteros y no como un cuadro de registros. La búsqueda de un registro se lleva a cabo simplemente recorriendo la lista de los registros que tienen el mismo valor de resumen, con ayuda de la función de búsqueda sobre las listas encadenadas.

La inserción de un registro d en una tabla t se realiza como sigue.

- 40
- Se busca la clave x en $t[h(x)]$.
 - En caso de fracaso, se produce una asignación dinámica de una nueva celda cuyos dos campos están llenos y

que se ubica encabezando la lista encadenada.

- En caso de éxito, se produce una sustitución del antiguo campo por el nuevo.

5 La supresión de un registro en una tabla es más delicada, ya que es necesario determinar la celda precedente a la celda c que contiene el registro que ha de suprimirse. Trato aparte merece el caso de la celda de cabecera, la cual no tiene predecesor. En cualquier caso, la celda suprimida tiene que ser desatribuida mediante una función de borrado.

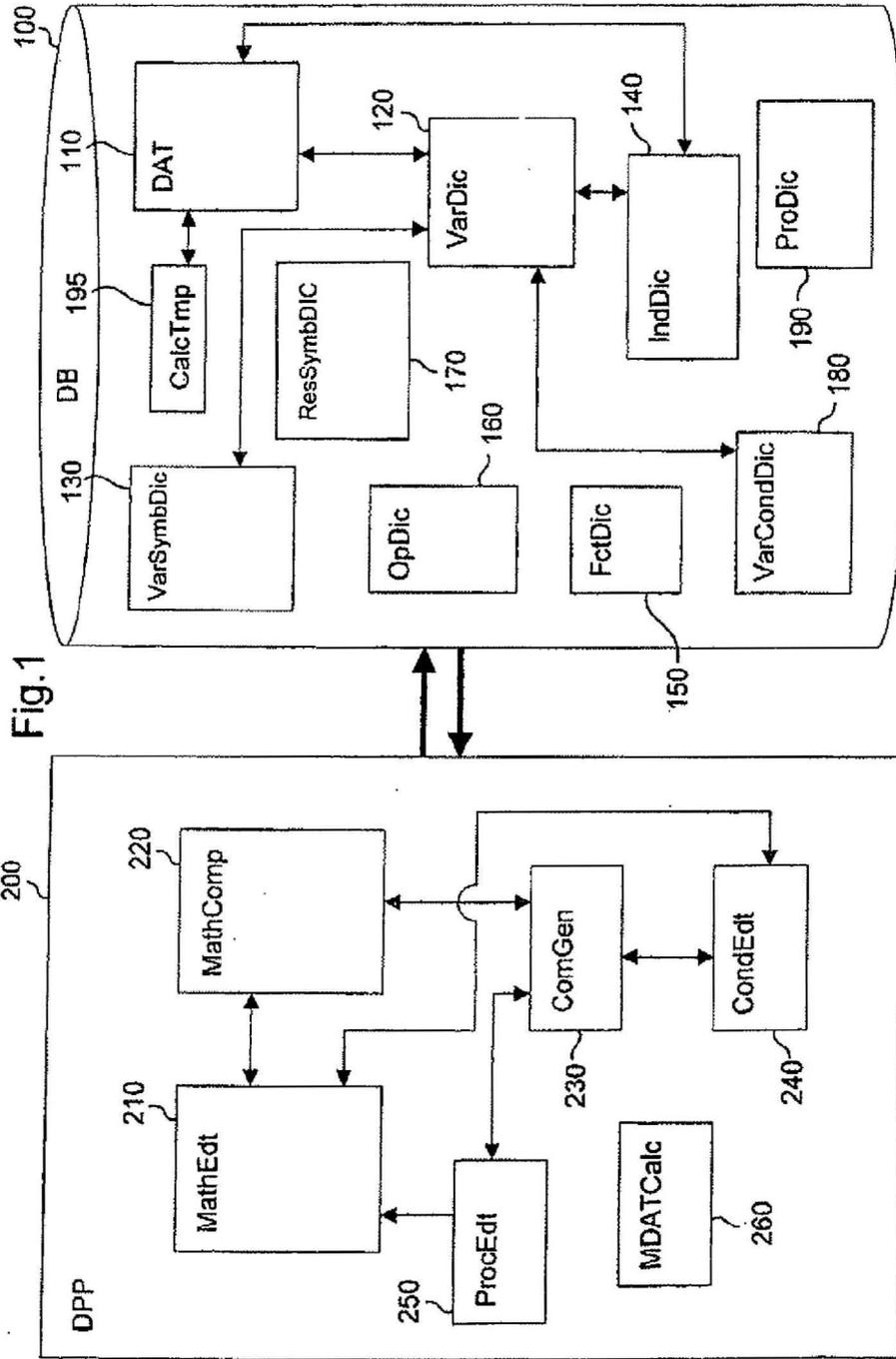
El coeficiente de carga α es la longitud media de las listas encadenadas. En una hipótesis de uniformidad de la función resumen, se muestra que el número medio de accesos necesarios para una búsqueda, negativa o positiva, es como mucho de $1+\alpha$.

REIVINDICACIONES

1. Dispositivo de tratamiento de datos, que comprende
- una unidad de tratamiento (200) capaz de interactuar con una base de datos (100), comprendiendo la base de datos unas tablas de datos (110);
- 5
- una interfaz gráfica de usuario establecida para permitir a un usuario introducir una expresión matemática;
- comprendiendo la unidad de tratamiento:
- un editor de expresión matemática (210) capaz de establecer una representación informática de trabajo de una expresión matemática en interacción con la interfaz gráfica de usuario, hallándose la expresión matemática en lenguaje natural matemático y comprendiendo ésta al menos un símbolo de variable matemática compleja, que presenta un índice y/o un exponente, y un símbolo matemático agregativo que opera con el símbolo de variable matemática compleja;
- 10
- un diccionario de símbolos (130) el cual, establecido para almacenar metadatos referentes al símbolo de variable matemática compleja, comprende información sobre la estructura de tablas (110) de la base de datos (100);
- 15
- un editor de condiciones (180; 240) dispuesto para producir una expresión de condición, asociada al símbolo de variable matemática compleja, siendo la expresión de condición utilizable en combinación con los metadatos para seleccionar un subconjunto de datos de la base de datos (100);
- 20
- un generador de consultas (220; 230) dispuesto para tratar la representación informática de trabajo para producir una secuencia de código que ejecuta la expresión matemática de acuerdo con las reglas del lenguaje natural matemático y con el reconocimiento del símbolo matemático agregativo y del símbolo de variable matemática compleja con el cual opera:
- utilizar el editor de condiciones (180; 240) para producir una expresión de condición de trabajo;
- 25
- utilizar los metadatos para producir un código capaz de utilizar la expresión de condición de trabajo para rellenar un cuadro de datos de trabajo con tal subconjunto de datos y,
 - combinar el operador correspondiente al símbolo matemático agregativo con el cuadro de datos de trabajo.
2. Dispositivo de tratamiento de datos según la reivindicación 1, **caracterizado porque** el símbolo matemático agregativo comprende un operador agregativo, de la clase suma o producto, el cual trabaja sobre un símbolo de variable matemática compleja.
- 30
3. Dispositivo de tratamiento de datos según la reivindicación 2, **caracterizado porque**, habiéndose asociado anteriormente un símbolo de variable matemática compleja a al menos una columna de al menos una tabla (110), la expresión de condición restringe el juego de valores contenido en esta columna.
- 35
4. Dispositivo de tratamiento de datos según una de las reivindicaciones precedentes, **caracterizado porque** el editor de expresión matemática (210) está dispuesto para establecer una representación informática de una secuencia de expresiones matemáticas que comprende los símbolos matemáticos agregativos referentes a los símbolos de variables matemáticas complejas y **porque** el generador de consultas (230) está dispuesto para reaccionar iterativamente al encuentro, en cada expresión, de un símbolo matemático agregativo y de un símbolo asociado de variable matemática compleja produciendo, con ayuda de metadatos particulares y del código ya establecido para ese símbolo de variable matemática compleja, código ejecutable capaz de resolver la condición contenida en dicha expresión, llenando al propio tiempo el cuadro de datos de trabajo (195) con los datos derivados de la resolución de la condición, así como de aplicar a este cuadro de datos de trabajo el operador correspondiente al símbolo matemático agregativo.
- 40
5. Dispositivo de tratamiento de datos según una de las reivindicaciones precedentes, **caracterizado porque** el editor de expresión matemática (210) está dispuesto para tratar una expresión matemática que comprende los símbolos matemáticos agregativos, operante cada uno de ellos sobre un símbolo de variable matemática compleja, **porque**, en presencia de un símbolo matemático agregativo en la representación informática de trabajo procedente del editor de expresión matemática, el editor de condiciones (240) es susceptible de ser accionado para establecer una correspondencia entre un símbolo matemático agregativo y un metadato particular con su símbolo asociado de variable matemática compleja produciendo, con ayuda del metadato particular y de la naturaleza del operador correspondiente al símbolo matemático agregativo del código ejecutable capaz de resolver la condición contenida en dicha expresión, llenando al propio tiempo el cuadro de datos de trabajo (195) con los datos derivados de la resolución de la condición,
- 45
- 50

así como de aplicar este operador particular a este cuadro de datos de trabajo.

- 5 6. Dispositivo de tratamiento de datos según una de las anteriores reivindicaciones precedentes, **caracterizado porque** el generador de consultas (230; 220) comprende un mecanismo apto para efectuar selectivamente tratamientos diferentes según la comparación de una cadena de la representación informática de trabajo con tres conjuntos de diferentes posibilidades, los cuales respectivamente representan un operador de agregación, un operador simple y un símbolo reservado, pudiendo dicho mecanismo ser activado sobre una prueba relativa a un símbolo de operador matemático.
- 10 7. Dispositivo de tratamiento de datos según la reivindicación 6, **caracterizado porque** el generador de consultas (220; 230) comprende un mecanismo apto para efectuar selectivamente los tratamientos diferentes sobre la base de una prueba relativa a un símbolo de variable.
8. Dispositivo de tratamiento de datos según una de las anteriores reivindicaciones precedentes, **caracterizado por** comprender además un editor de tratamiento (250) capaz de establecer una expresión de un encadenamiento ordenado de expresiones matemáticas a partir de unas segundas reglas formales.
- 15 9. Dispositivo de tratamiento de datos según una de las reivindicaciones precedentes, **caracterizado por** comprender además un generador de consultas suplementario (260) capaz de establecer, para un símbolo de variable particular, partiendo de datos almacenados (110) correspondientes a este símbolo de variable particular, una progresión matemática que relaciona estos datos entre sí.
- 20 10. Dispositivo de tratamiento de datos según la reivindicación 9, **caracterizado porque** el generador de consultas suplementario (260) es capaz de generar un código que ejecuta dicha progresión para completar el conjunto de los datos almacenados (110) asociados a dicho símbolo de variable particular.
11. Dispositivo de tratamiento de datos según una de las reivindicaciones precedentes, **caracterizado porque** el generador de consultas (220; 230) está dispuesto para autorizar condicionalmente, en una expresión matemática, un símbolo de variable matemática cuyos metadatos asociados contienen una expresión en espera, sin correspondencia inmediata con los datos almacenados en memoria.
- 25 12. Dispositivo de tratamiento de datos según una de las reivindicaciones precedentes, **caracterizado porque** el cuadro de datos de trabajo (195) es un cuadro abierto, susceptible de ser llenado con posterioridad por el generador de consultas (220; 230) en particular para el tratamiento de expresiones matemáticas sucesivas.
- 30 13. Programa-producto susceptible de cooperar con un ordenador para constituir el dispositivo de tratamiento de los datos según una de las reivindicaciones precedentes.



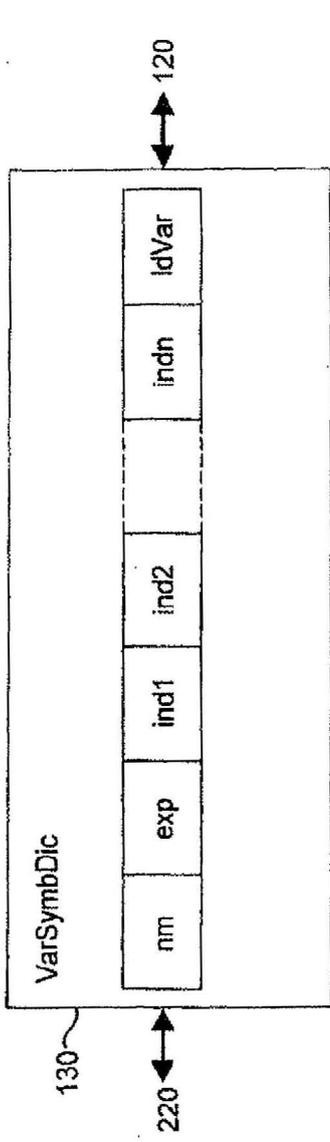


Fig. 2

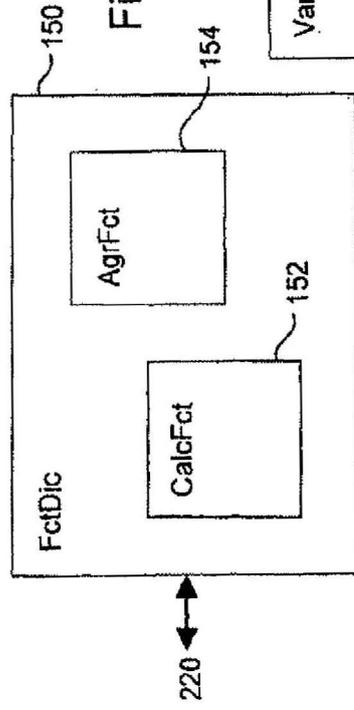


Fig. 4

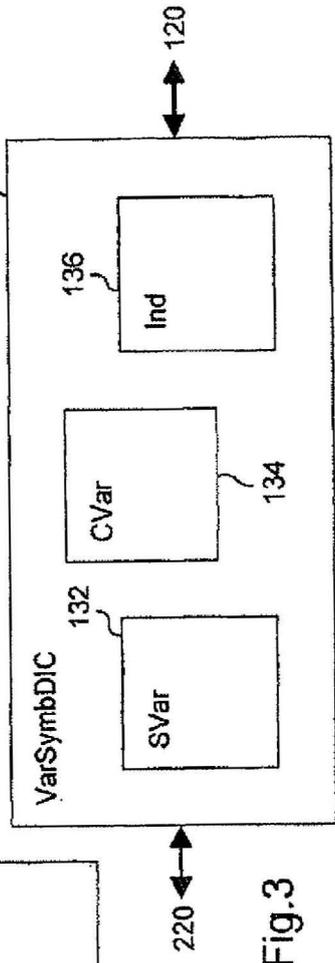


Fig. 3

Fig.5

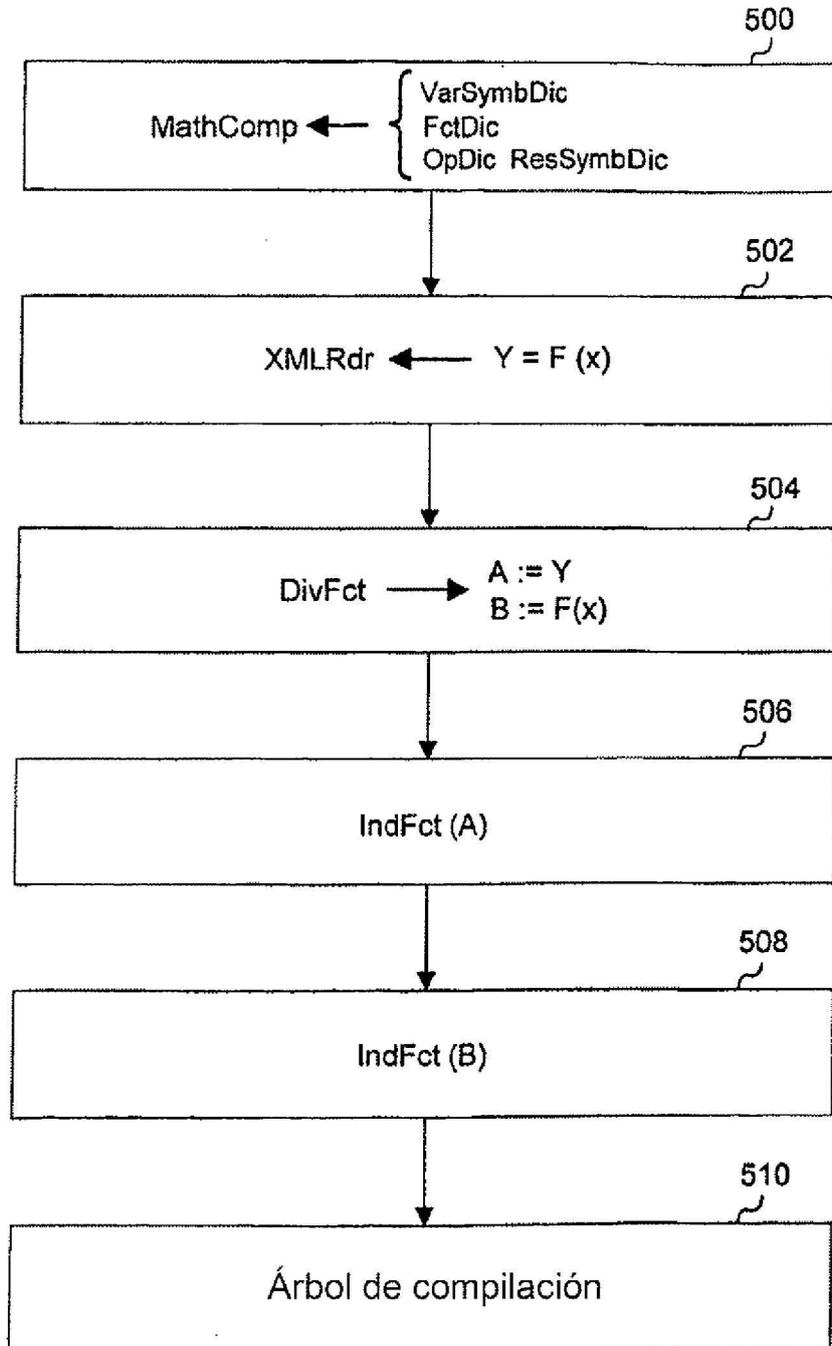
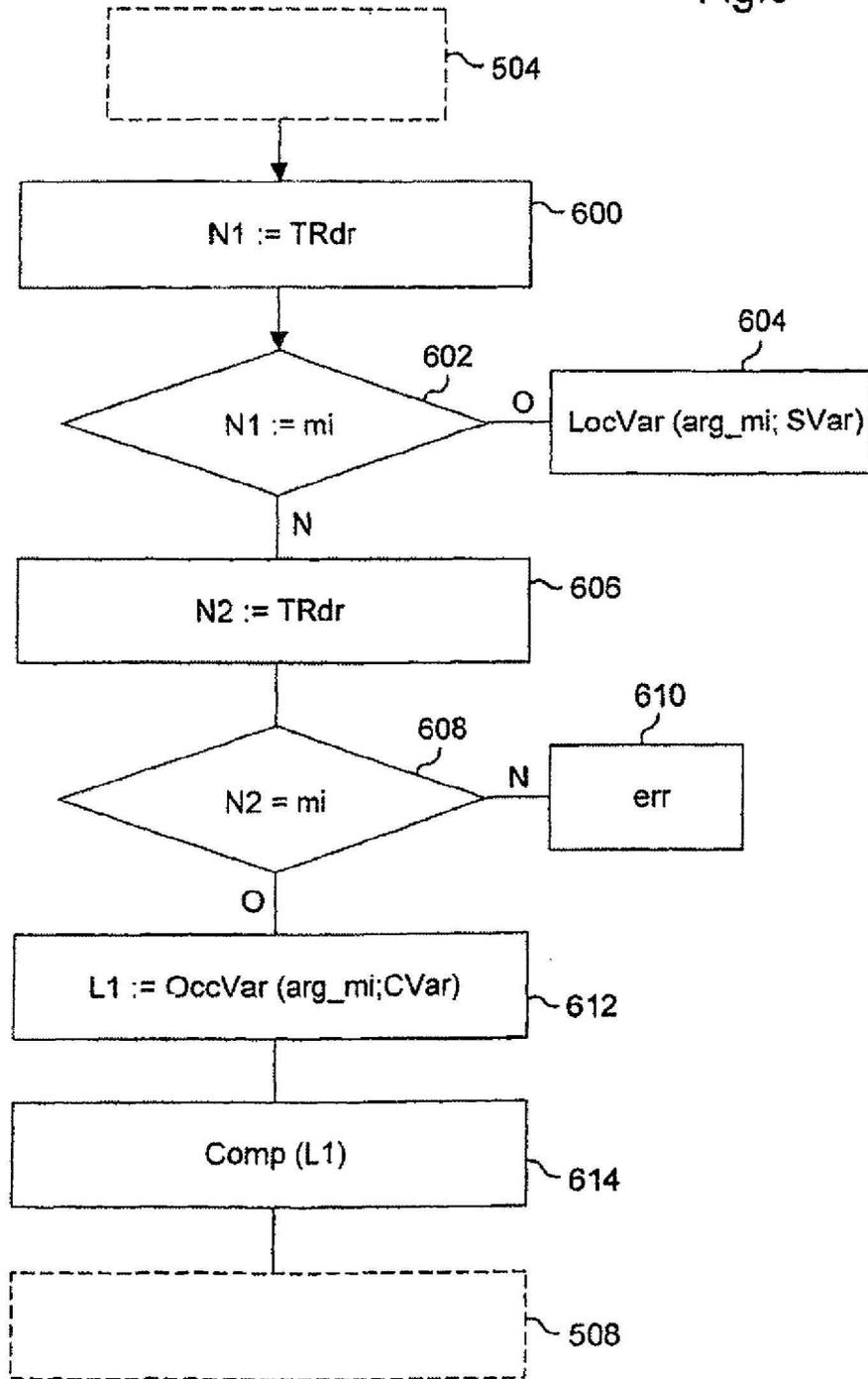


Fig.6



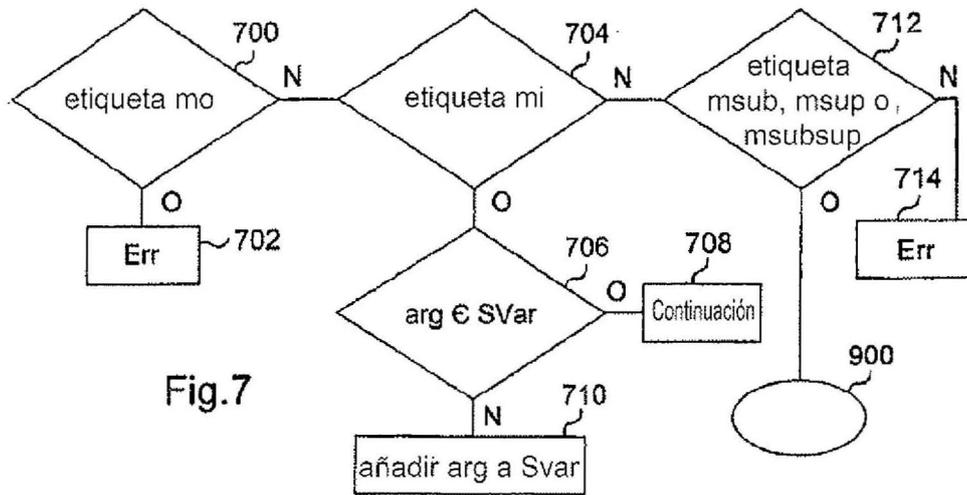


Fig.7

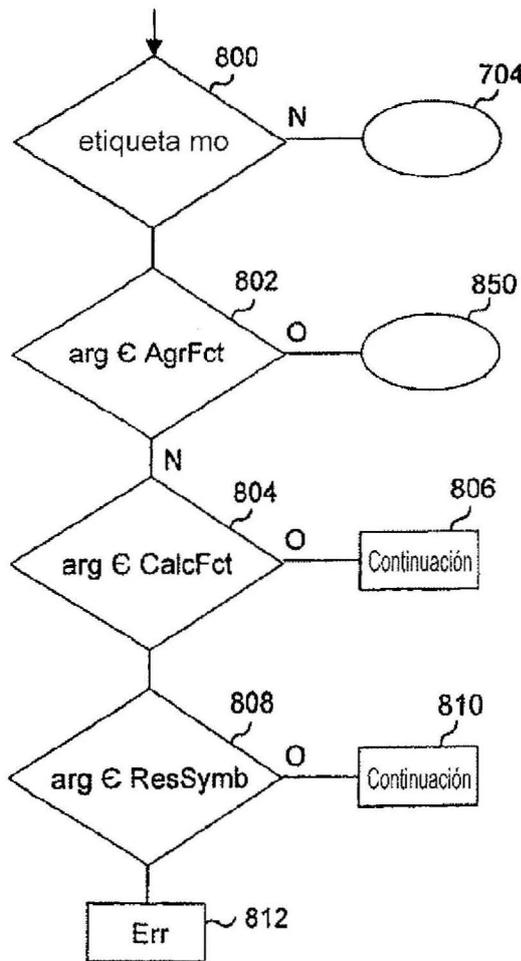


Fig.8

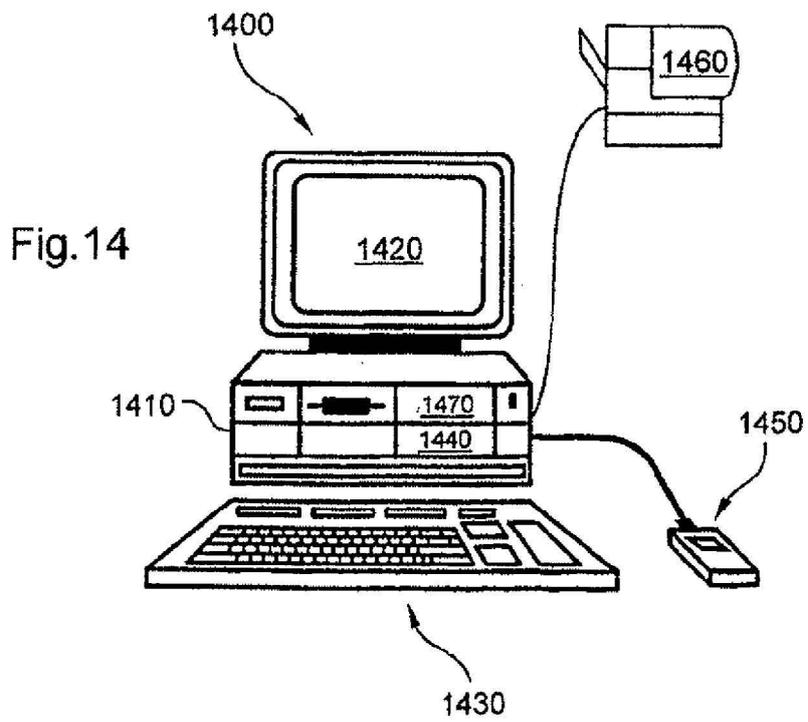
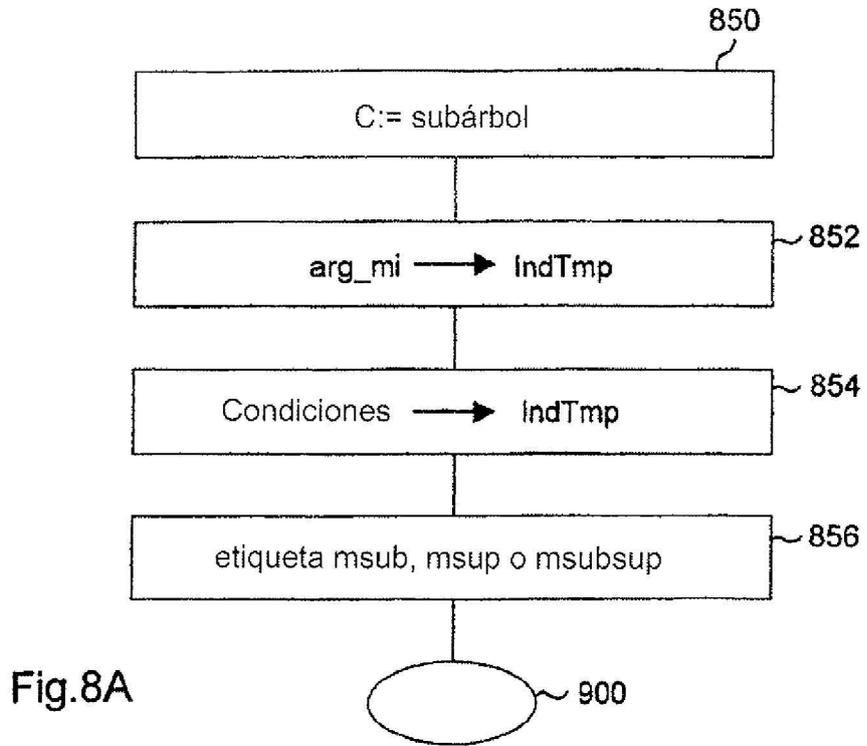
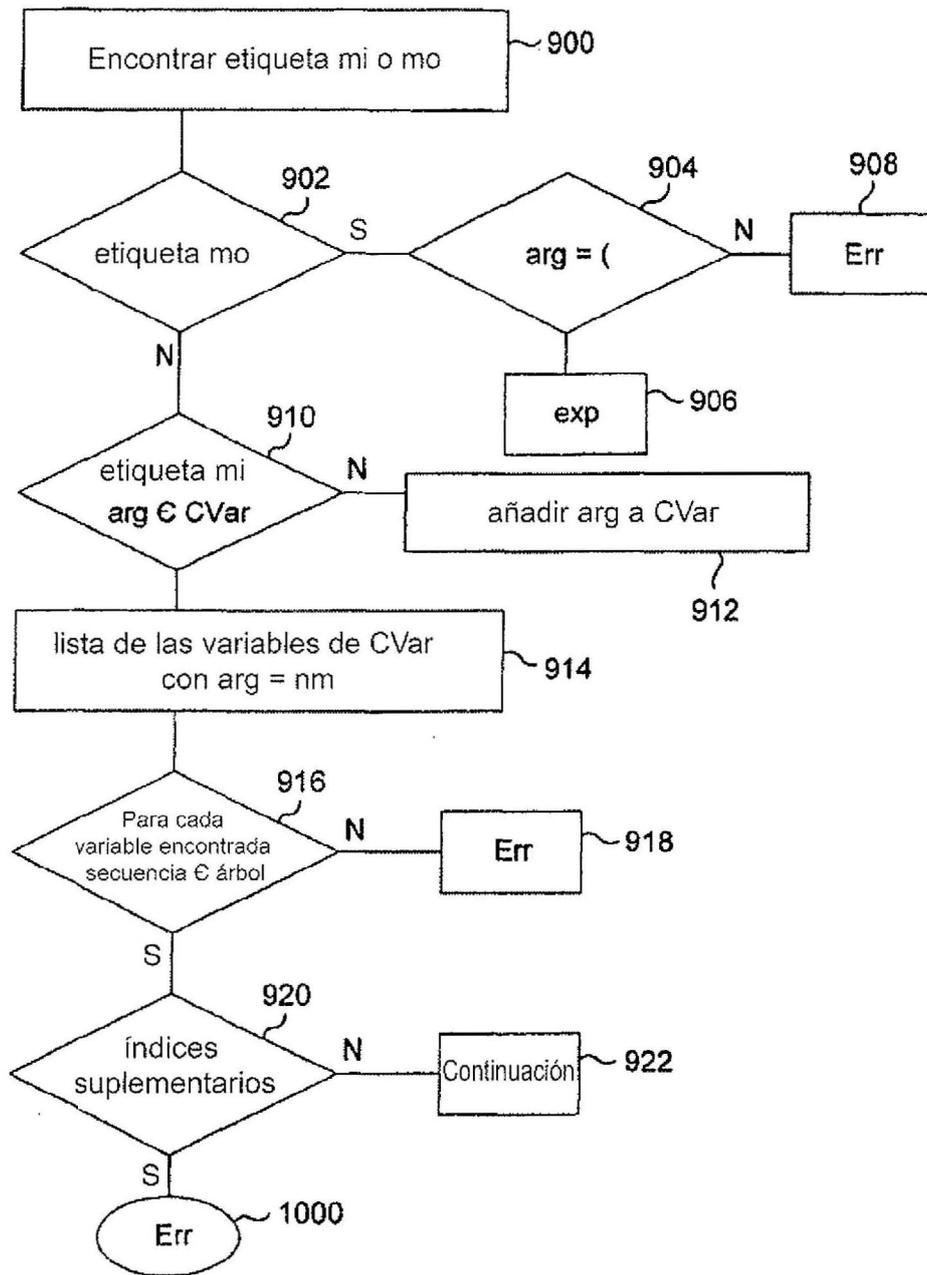


Fig.9



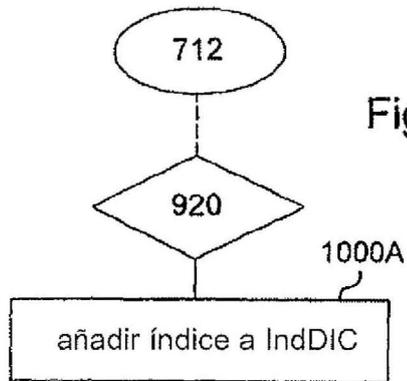


Fig.10A

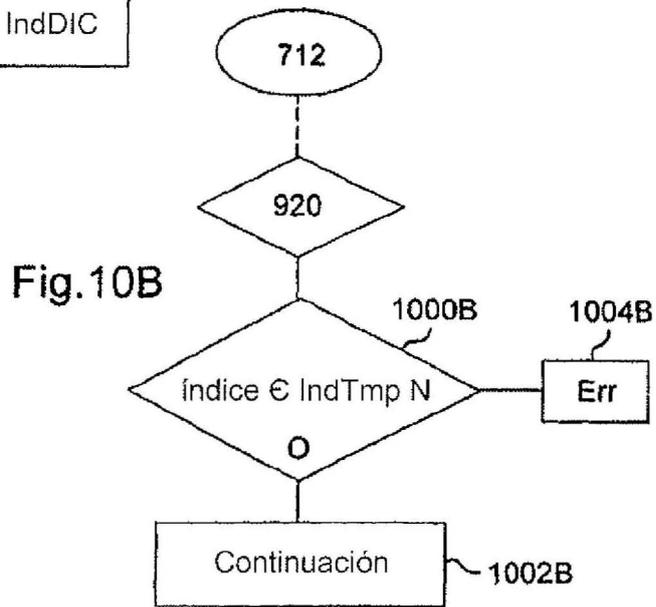


Fig.10B

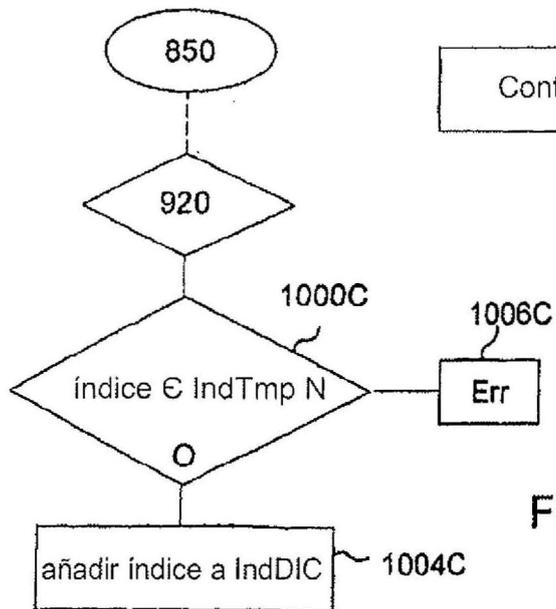


Fig.10C

Fig.11

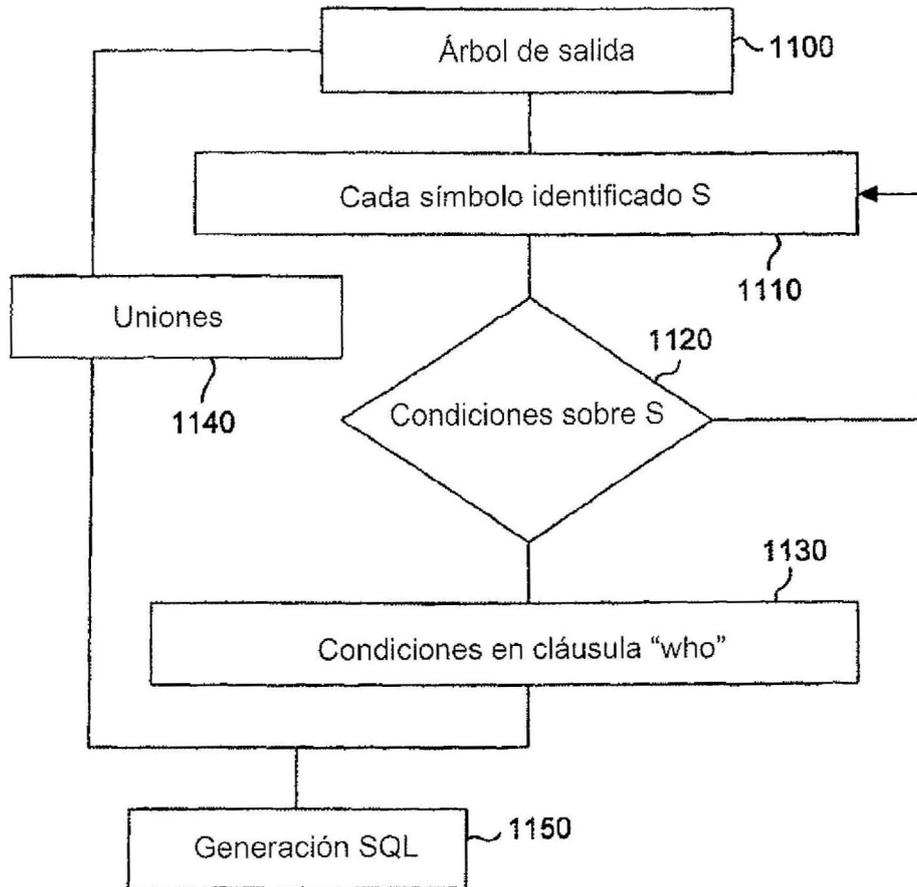


Fig.12

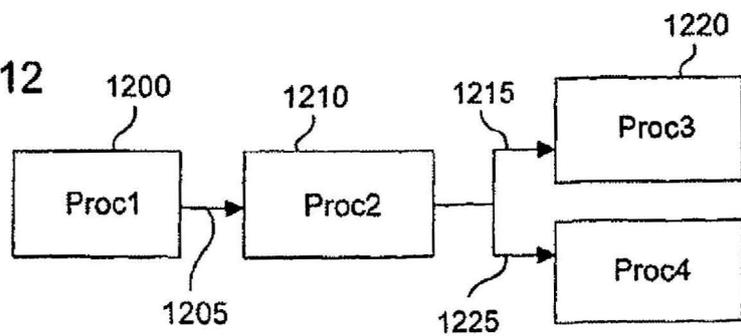
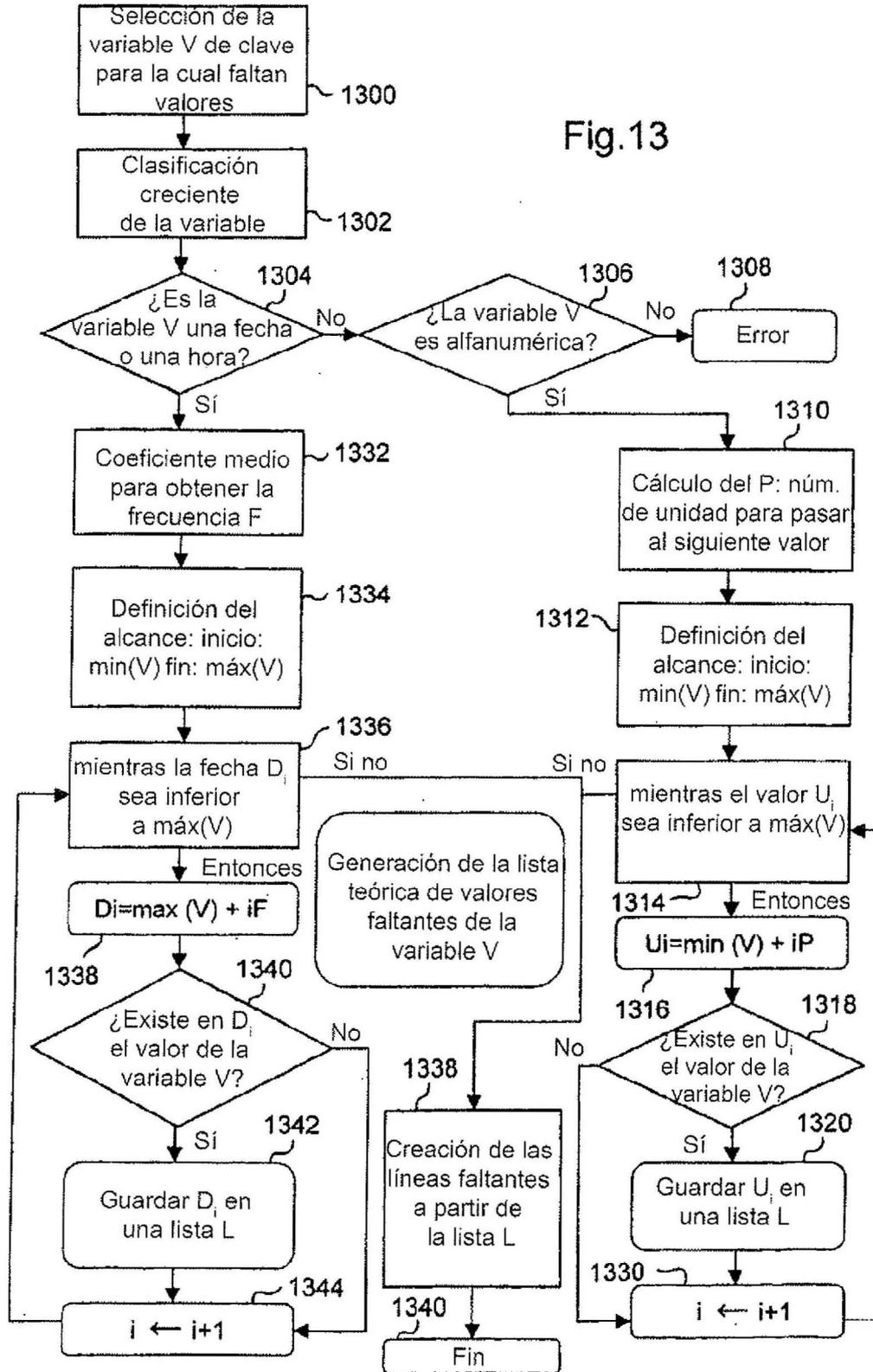


Fig.13



DOCUMENTOS INDICADOS EN LA DESCRIPCIÓN

En la lista de documentos indicados por el solicitante se ha recogido exclusivamente para información del lector, y no es parte constituyente del documento de patente europeo. Ha sido recopilada con el mayor cuidado; sin embargo, la EPA no asume ninguna responsabilidad por posibles errores u omisiones.

5 Documentos de patente indicados en la descripción

- US 20040133567 A1 [0004]