

308917



UNIVERSIDAD PANAMERICANA

32
2ej.

Con estudios incorporados a la Universidad Nacional Autónoma de México

ESCUELA DE INGENIERÍA

**SISTEMA DE DESARROLLO DE
APLICACIONES DE ANÁLISIS DE
INFORMACIÓN MULTIDIMENSIONAL**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
INGENIERO MECÁNICO ELECTRICISTA
ÁREA MECÁNICO ELECTRICISTA
P R E S E N T A**

ALEJANDRO DEL MORAL PALACIO

DIRECTOR:

ING. ALFONSO G. LEAL GUAJARDO

MÉXICO, D.F.

1998

**TESIS CON
FALLA DE ORIGEN**

260107



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

CONTENIDO

| | |
|--|-----------|
| I. ANTECEDENTES | 4 |
| A. MATRICES MATEMÁTICAS..... | 4 |
| 1. Definición de matriz..... | 4 |
| 2. Matriz multidimensional..... | 5 |
| 3. Matrices esparcidas..... | 5 |
| B. CONCEPTOS DE MODELOS DE INFORMACIÓN MULTIDIMENSIONAL | 6 |
| 1. Matrices densas..... | 6 |
| 2. Matrices esparcidas..... | 8 |
| 3. Modelos multidimensionales de información | 10 |
| C. PROBLEMAS DE LOS SISTEMAS DE INFORMACIÓN TRADICIONALES..... | 12 |
| 1. Sistemas tradicionales..... | 12 |
| 2. Hojas de cálculo | 14 |
| 3. Bases de datos..... | 14 |
| D. SISTEMAS DE PROCESAMIENTO ANALÍTICO EN LÍNEA | 18 |
| 1. Las 12 reglas de OLAP | 18 |
| 2. Sistemas de análisis de información multidimensional..... | 22 |
| II. ANÁLISIS Y DISEÑO DEL SISTEMA..... | 24 |
| A. DEFINICIÓN DEL PROBLEMA..... | 24 |
| B. JUSTIFICACIÓN DE LA TESIS | 25 |
| C. HERRAMIENTAS UTILIZADAS..... | 28 |
| D. ARQUITECTURA DEL SISTEMA..... | 29 |
| 1. Arquitectura Interna..... | 29 |
| 2. Subsistema de almacenamiento de datos | 29 |
| 3. Subsistema de almacenamiento del modelo analítico..... | 35 |
| 4. Subsistema de cálculo del modelo..... | 38 |
| 5. Subsistema de cálculos personalizados..... | 42 |
| 6. Arquitectura al usuario..... | 46 |
| III. USO DEL SISTEMA..... | 49 |
| A. LA HERRAMIENTA DE DISEÑO..... | 49 |
| B. CREACIÓN DE UN NUEVO MODELO..... | 50 |
| 1. Modificación de la estructura de un modelo | 51 |
| C. LA PANTALLA DE EDICIÓN DEL MODELO MULTIDIMENSIONAL..... | 54 |
| IV. CONCLUSIONES | 61 |
| V. BIBLIOGRAFÍA..... | 65 |

TABLA DE FIGURAS

| | |
|---|----|
| Figura 1. Representación de una matriz..... | 4 |
| Figura 2. Representación de una matriz multidimensional..... | 5 |
| Figura 3. Número de elementos de una matriz | 5 |
| Figura 4. Ejemplo de una matriz densa | 7 |
| Figura 5. Ejemplo de una matriz tridimensional..... | 7 |
| Figura 6. Dimensiones del ejemplo de matriz esparcida..... | 9 |
| Figura 7. Tipos de información manejados por sistemas de información | 12 |
| Figura 8. Sistemas de información disponibles, por cantidad de información y cantidad de análisis | 13 |
| Figura 9. Diagrama general del sistema | 29 |
| Figura 10. Diagrama de un bloque..... | 31 |

| | |
|---|----|
| Figura 11. Diagrama del encabezado de un bloque..... | 31 |
| Figura 12. Dimensiones de la matriz de ejemplo | 32 |
| Figura 13. Estructura de bloque en el ejemplo..... | 32 |
| Figura 14. Ejemplo de un índice de bloques | 32 |
| Figura 15. Archivo de datos del ejemplo..... | 33 |
| Figura 16. Índice de bloques del ejemplo tras el cálculo de la dimensión regiones | 33 |
| Figura 17. Índice de bloques del ejemplo tras el cálculo de la dimensión clientes | 34 |
| Figura 18. Estructura del archivo de almacenamiento del modelo..... | 36 |
| Figura 19. Ejemplo de archivo de almacenamiento del modelo..... | 37 |
| Figura 20. Ejemplo de archivos de estructura de dimensión..... | 38 |
| Figura 21. Ejemplo de estructura de bloque para determinación de un cálculo..... | 40 |
| Figura 22. Mapa de bits del conjunto de celdas a calcular en el ejemplo de un cálculo | 40 |
| Figura 23. Bloques existentes al inicio del cálculo en el ejemplo de un cálculo | 41 |
| Figura 24. Estructura secuencial del índice de bloques en el ejemplo..... | 41 |
| Figura 25. Mapa de bits de bloques a calcularse en el ejemplo | 42 |
| Figura 26. Opciones de líneas disponibles en un cálculo | 43 |
| Figura 27. Orden seguido en el cálculo inteligente del ejemplo..... | 44 |
| Figura 28. Formato del archivo de almacenamiento de cálculo | 45 |
| Figura 29. Arquitectura al usuario del sistema..... | 47 |
| Figura 30. Estructura del objeto <i>TModel</i> | 47 |
| Figura 31. Estructura del objeto <i>TDynamicModViewer</i> | 48 |
| Figura 32. Icono del sistema..... | 49 |
| Figura 33. Menú de la pantalla principal | 50 |
| Figura 34. Cuadro de diálogo para creación de un nuevo modelo..... | 50 |
| Figura 35. Cuadro de diálogo para selección del directorio del nuevo modelo | 51 |
| Figura 36. Ventana del editor de estructura..... | 52 |
| Figura 37. Pantalla de edición de dimensiones | 52 |
| Figura 38. Cuadro de diálogo de edición de miembros | 53 |
| Figura 39. Funciones matemáticas disponibles | 54 |
| Figura 40. Ventana de edición del modelo..... | 55 |
| Figura 41. Pestañas de la ventana de edición del modelo..... | 55 |
| Figura 42. Ventana de creación de un nuevo modelo..... | 55 |
| Figura 43. Ventana de edición de cálculos | 56 |
| Figura 44. Ventana de edición de vistas | 57 |
| Figura 45. Acciones disponibles en la ventana de edición de vistas..... | 58 |
| Figura 46. Ventana de selección manual de miembros | 58 |
| Figura 47. Edición de vistas. Ventana de opciones de la tabla | 59 |
| Figura 48. Edición de vistas. Ventana de opciones de formato numérico | 60 |
| Figura 49. Características del modelo de la prueba comparativa de desempeño..... | 61 |
| Figura 50. Resultados de la prueba comparativa de desempeño..... | 63 |

INTRODUCCIÓN

En la actualidad, los sistemas de información han llegado a ocupar un lugar preponderante en muchas ramas del conocimiento. Esto se ha debido en alguna proporción a que son capaces de manejar una gran cantidad de información, liberando al ser humano de tareas tediosas y repetitivas. Sin embargo, este hecho ha dado como resultado al utilizarlos el que frecuentemente se dispone de un volumen considerable de datos, situación nueva y por lo que no se tiene la costumbre ni las técnicas para obtener de ella el mayor provecho, con el consiguiente problema de que es difícil analizar, controlar, monitorear y evaluar esta información. El estado presente en este entorno es el de una subutilización generalizada de la información.

En algunos casos, los sistemas de cómputo permiten aventurarse a realizar análisis o cálculos importantes, que anteriormente eran imposibles debido a que no existían los recursos necesarios para su consecución. Sin embargo, aunque ahora existe la capacidad de procesamiento de información, frecuentemente no existen los sistemas o herramientas que otorgan la facilidad de realizar estos análisis, o si los hay, requieren el aprendizaje de una gran cantidad de conceptos que no están relacionados con el problema a resolver.

El desaprovechamiento de información sucede en casi todas las áreas en las que se usan los sistemas de cómputo. El principal obstáculo es el esfuerzo que requiere el obtener la información necesaria para tomar una decisión adecuada, o para encontrar las causas de un fenómeno o un problema. Esto se puede atribuir a que las herramientas que actualmente se encuentran disponibles para el manejo de información han sido diseñadas con el fin principal de almacenar datos, no con un objetivo extractivo o analítico.

Obstáculos de menor magnitud y sin embargo, importantes, son la falta de disponibilidad de herramientas que sean capaces de realizar análisis de gran magnitud con la información que ya se posee. De igual manera es un problema el que las herramientas disponibles han evolucionado a partir del concepto de almacenamiento de la información y no de la necesidad de la consulta de ella. Esto es, que estos sistemas requieren un diseño estructural basado en la forma óptima de almacenar los datos y no al modelo conceptual de la información. Esto le da una complejidad tal a los sistemas que los usuarios en muchas ocasiones se abstienen de usarlos.

En el mercado actual, comienzan a aparecer sistemas orientados a la consulta de información de acuerdo a modelos analíticos que dan al usuario la capacidad de desplazarse de una manera más cómoda que en los sistemas tradicionales. Esto se logra por medio de diversos métodos, de acuerdo a la clase de información de la que se trate. En particular, hay una gran variedad de información que puede manejarse en base a matrices o modelos multidimensionales.

Dado que la aparición en el mercado de sistemas de análisis de información multidimensional es muy reciente, su evolución no es muy grande. En el presente, no se dispone aún de herramientas que brinden la capacidad de desarrollar aplicaciones que utilicen esta técnica. De hecho, los sistemas disponibles tienen una capacidad limitada o un costo que los hace en general inaccesibles. Igualmente restrictivo es que estos sistemas requieren la realización del modelado, almacenamiento y análisis de información en sistemas propietarios que no permiten al usuario añadirles nuevas capacidades.

Esta tesis tiene como objetivo el realizar un sistema que permita el almacenamiento, análisis y consulta de modelos de información multidimensional. Asimismo, se pretende que el sistema sirva como plataforma para el desarrollo de nuevas aplicaciones que utilicen las capacidades multidimensionales de este sistema sin tener que desarrollar toda la programación necesaria para lograr tal efecto. Es decir, que sirva como herramienta de desarrollo para aplicaciones de análisis de modelos de información multidimensional.

I. ANTECEDENTES

A. MATRICES MATEMÁTICAS

Los modelos multidimensionales que se tratan en este trabajo se basan en su más bajo nivel en matrices matemáticas. El objetivo no es el realizar un sistema que sea capaz de realizar el cálculo de operaciones matriciales complejas, sino que permita el manejo de información que presente la forma matricial.

1. Definición de matriz

Matriz: conjunto de números que puede representarse de manera tabular, como la siguiente:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1j} \\ a_{21} & a_{22} & \dots & a_{2j} \\ \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{ij} \end{pmatrix}$$

Figura 1. Representación de una matriz

Donde a_{ij} es un número. a_{ij} se define como un elemento de la matriz. El número de elementos de la matriz es igual a $n*m$.

2. Matriz multidimensional

Aquella matriz que tiene más de dos dimensiones. Cada dimensión es compuesta de n miembros. El cruce de un miembro de cada dimensión se define como un elemento.

Ejemplo:

$$\left| \begin{array}{cccc} a_{111} & a_{121} & \dots & a_{1j1} \\ a_{211} & a_{221} & \dots & a_{2j1} \\ \dots & \dots & \dots & \dots \\ a_{i11} & a_{i21} & \dots & a_{ij1} \end{array} \right| \left| \begin{array}{cccc} a_{112} & a_{122} & \dots & a_{1j2} \\ a_{212} & a_{222} & \dots & a_{2j2} \\ \dots & \dots & \dots & \dots \\ a_{i12} & a_{i22} & \dots & a_{ij2} \end{array} \right| \dots \left| \begin{array}{cccc} a_{11o} & a_{12o} & \dots & a_{j1o} \\ a_{21o} & a_{22o} & \dots & a_{2jo} \\ \dots & \dots & \dots & \dots \\ a_{i1o} & a_{i2o} & \dots & a_{ijo} \end{array} \right|$$

Figura 2. Representación de una matriz multidimensional

Si n es el número de dimensiones y si se define la magnitud de la dimensión i como D_i , se puede calcular el número de elementos E de la matriz como

$$E = \prod_{i=1}^n D_i$$

Figura 3. Número de elementos de una matriz

3. Matrices esparcidas

Una matriz esparcida es aquella que no contiene información en todos sus elementos, sino que sólo la tiene en algunos de ellos.

Una matriz no esparcida se denomina densa.

Supóngase que al cumplirse la existencia de información para un miembro de una dimensión, existe para los demás miembros de esa dimensión. En este caso, la dimensión se denomina densa. En caso contrario, se denomina esparcida. Una matriz esparcida está formada entonces, por dos o más dimensiones esparcidas y una cantidad variable de dimensiones densas. Una matriz esparcida puede no tener dimensiones densas.

En este caso, el número de elementos en la matriz será

$$E = r \prod_{i=1}^n D_i$$

Ilustración 1-1

Donde r es el número de cruces de miembros de todas las dimensiones esparcidas que contienen información y D_i es una dimensión densa.

Como se puede observar, las matrices sólo sirven para manejar información numérica, por lo que quedan excluidas de ciertas funciones como el manejo de documentos, imágenes, etcétera.

B. CONCEPTOS DE MODELOS DE INFORMACIÓN MULTIDIMENSIONAL

Muchos de los sistemas de información que se utilizan actualmente manejan información que puede ordenarse de manera tabular. Los casos más evidentes son las bases de datos y las hojas de cálculo, que se basan fundamentalmente en tablas de datos. En el caso específico de que la información sea puramente numérica, ella puede representarse como una matriz. Estas matrices ocurren de manera bastante frecuente en todos los campos del conocimiento. Como ejemplos de información matricial se tienen desde Estados de Resultados, hasta las conocidas tablas de propiedades físico - químicas que se usan como consulta en la ciencia.

1. Matrices densas

Úsese como ejemplo alguna tabla de propiedades, como la tabla de propiedades de algún líquido hipotético dada alguna presión:

| Temperatura [°C] | 100°C | 110°C | 120°C | 130°C |
|-----------------------|-------|-------|-------|-------|
| Entalpía (H) | 123 | 133 | 143 | 153 |
| Entropía (S) | 24 | 34 | 44 | 54 |
| Densidad (δ) | 5.0 | 4.5 | 4.0 | 3.5 |

Figura 4. Ejemplo de una matriz densa

Es posible saber algunas de las propiedades de algún líquido dadas una cierta presión y temperatura. La tabla anterior muestra ciertas propiedades a una presión específica. Como puede verse, puede tratarse la tabla anterior como una matriz de dos dimensiones. Una dimensión es la Temperatura. A la otra dimensión se le llamará "variables". En el ejemplo anterior, la dimensión variables tiene los miembros entalpía, entropía y densidad. La magnitud de esta dimensión es 3, pues tiene 3 miembros.

Esta misma tabla se puede obtener para varias presiones. Esta información se puede ver como una matriz de tres dimensiones, como se muestra en la siguiente figura:

| Temperatura [°C] | 100°C | 110°C | 120°C | 130°C |
|-----------------------|-------|-------|-------|-------|
| Entalpía (H) | 123 | 133 | 143 | 153 |
| Entropía (S) | 24 | 34 | 44 | 54 |
| Densidad (δ) | 5.0 | 4.5 | 4.0 | 3.5 |

Figura 5. Ejemplo de una matriz tridimensional

En la figura anterior se aprecia una matriz de tres dimensiones. La tercera dimensión es la presión. De la misma manera, se podría obtener la información de la matriz tridimensional anterior para una cierta cantidad de materiales. Esta información sería la misma en su forma para cada uno de estos materiales. Es posible, pues, definir "materiales" como la cuarta dimensión.

De la explicación anterior se puede observar que hay cierta clase de información que puede analizarse como una matriz multidimensional de información. Con gran frecuencia y en casi cualquier ámbito se maneja información numérica que corresponde al patrón de una matriz multidimensional.

2. Matrices esparcidas

Hay una gran cantidad de casos en que las matrices de información que se obtienen son esparcidas. Esto es, que no todos los elementos de esa matriz contienen información. Es importante hacer notar la diferencia entre que no existe información y que la información sea cero. El cero es un dato existente, es decir, es un dato igual a cero. En cambio, el caso estudiado es en el que no se posee información para un elemento en específico.

Tómese como ejemplo la información comercial de una compañía A. Esta compañía A se dedica a la venta de sus productos a ciertos clientes. Estos clientes son distribuidores de venta directa al público. La compañía A tiene muchos clientes, productos y áreas de distribución de sus productos. A la dirección de la compañía A le interesa tener ciertas cantidades para propósitos de toma de decisión. Estas son:

- Venta
- Gasto de envío
- Utilidad Bruta
- Margen de Operación
- Costo de Venta
- Impuestos
- Utilidad Neta

La dirección de la compañía quisiera conocer estos datos por mes, semestre y año. Estas cifras se generan cada vez que se realiza una venta. No obstante, no todos los clientes compran todos los productos. Asimismo, no todos los clientes compran en todas las regiones, aunque puede darse el caso de que un cliente compre en más de una región, pues tiene varias sucursales. Si se hacen las siguientes suposiciones:

- Hay 100 productos
- Hay 1000 clientes
- Hay 10 regiones en la organización

Se puede llegar a la definición de una matriz multidimensional. Se observan las siguientes dimensiones de la matriz, con sus respectivas magnitudes:

| Dimensión | Magnitud | Tipo de dimensión |
|-----------|-------------------------------------|-------------------|
| Variables | 7 | Densa |
| Tiempo | 12 meses + 2 semestres + 1 año = 15 | Densa |
| Región | 10 | Esparcida |
| Cliente | 1000 | Esparcida |
| Producto | 100 | Esparcida |

Figura 6. Dimensiones del ejemplo de matriz esparcida

Es posible saber qué dimensiones son densas y cuáles son esparcidas con el siguiente método:

Tómese la primera dimensión: variables. Se sabe que en el momento de generarse una venta, se generan las demás variables. Es decir, que la existencia de un miembro (ventas) hace cumplir la existencia de todos los demás miembros.

La segunda dimensión no parece ser densa de acuerdo a lo anteriormente escrito. Sin embargo, en realidad se sabe que siempre existirán todos los meses en el tiempo. En este caso, es posible garantizar la existencia de información aun cuando no exista información en ningún miembro. Ninguna de las condiciones anteriores se cumple para las demás dimensiones. Por lo tanto, son esparcidas.

De acuerdo a la fórmula, es posible saber la cantidad total de elementos en la matriz. Supóngase que la matriz está llena en aproximadamente el 10% de su total. El número de elementos es el producto del número de miembros de las dimensiones multiplicada por 0.1. Se puede decir que la Densidad de Información de la Matriz es de 0.1.

$$E=7*15*10*1000*100*0.1=10,500,000$$

3. Modelos multidimensionales de información

En los ejemplos anteriores se explicó el manejo de información en forma de matrices multidimensionales. Sin embargo, no se prestó atención a los cálculos que se realizan en ellas. Tómese el ejemplo de la matriz esparcida. Puede observarse que algunos miembros de una dimensión pueden calcularse a partir de otros miembros.

En la dimensión variables pueden definirse las siguientes reglas:

- Utilidad Bruta = Venta - Costo de Venta - Gasto de Envío
- Impuestos = Utilidad Bruta * 0.35
- Utilidad Neta = Utilidad Bruta - Impuestos
- Margen de Operación = Utilidad Neta / Ventas * 100.0

De igual manera, se definen otras reglas para las restantes dimensiones:

- Primer Semestre = Enero+Febrero+Marzo+Abril+Mayo+Junio
- Segundo Semestre = Julio+Agosto+Septiembre+Octubre+Noviembre+Diciembre
- Total Anual = Primer Semestre + Segundo Semestre
- Total Regiones = Región 1 + Región 2 + ... + Región 10
- Total Clientes = Cliente 1 + Cliente 2 + ... + Cliente 1000
- Total Productos = Producto 1 + Producto 2 + ... + Producto 100

De lo anterior se desprende que a partir de un conjunto de información definido por ciertos miembros de cada dimensión puede obtenerse el resto de la matriz. Es importante observar que las reglas definidas son conceptuales. No relacionan elementos de la matriz, sino miembros.

La unión de la matriz multidimensional de información con las reglas que la rigen dan como resultado un Modelo Multidimensional de Información.

De acuerdo a lo anterior, puede hacerse una distinción entre los elementos de la matriz que son resultado de alguna operación, y aquellos que son necesarios para obtener la matriz. Los primeros se denominan como "resultados" y los restantes como "datos".

Al realizar todas las operaciones descritas por las reglas, a partir de los datos del modelo para obtener los resultados, se resuelve el modelo, o dicho de otra forma, se obtiene su solución.

C. PROBLEMAS DE LOS SISTEMAS DE INFORMACIÓN TRADICIONALES

Uno de los principales retos que presenta el procesamiento de información en el presente es el procesamiento de las cada vez más grandes cantidades de información, en modelos cada vez más complejos, sin sacrificar el tiempo de respuesta. Existen muchas herramientas que ayudan a manejar información. Actualmente, sobresalen los sistemas que, con la ayuda de computadoras, facilitan el manejo de estos datos. Éstos, comúnmente, se han clasificado de acuerdo al tipo de información que manejan. Como ejemplos más notorios, están los siguientes:

| Sistema | Tipo de información que maneja |
|-----------------------------|--|
| Procesadores de palabras | Documentos (textos, imágenes, etcétera) |
| Bases de datos relacionales | Información tabular, de diverso tipo (textos, números, fechas, etcétera). Las tablas se relacionan entre sí. |
| Hojas de cálculo | Información principalmente numérica, tabular. |

Figura 7. Tipos de información manejados por sistemas de información

1. Sistemas tradicionales

Como se acaba de señalar, los sistemas se clasifican y se desarrollan de acuerdo al tipo de información de la que se trata. No obstante, parece que hay una cierta clase de información que pertenece al ámbito de las bases de datos relacionales y a las hojas de cálculo: información numérica tabular. Tradicionalmente, se han usado las hojas de cálculo cuando esta información es relativamente poca, y se desea hacer un gran número de análisis con ella. Por el contrario, se han utilizado las bases de datos relacionales cuando la cantidad de información es mayor, y los análisis (o cálculos) a realizar son menos. En la siguiente gráfica se observa el ámbito que alcanzan cada uno de ellos.

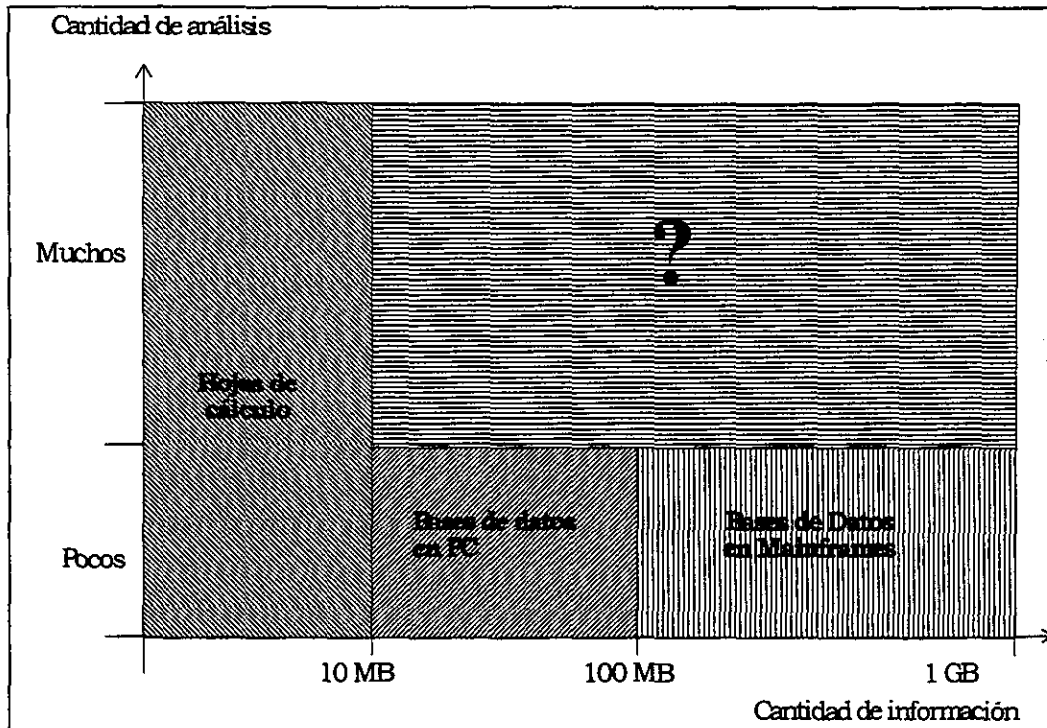


Figura 8. Sistemas de información disponibles, por cantidad de información y cantidad de análisis

En la gráfica anterior se observa un espacio vacío. Estos sistemas parecen no ser adecuados para el manejo de un gran volumen de información que impliquen una cierta cantidad de cálculos con ella. En específico, las hojas de cálculo, son las que más se acercan a los requerimientos, pues el realizar cualquier cantidad razonable de cálculos sobre una base de datos relacional implica un esfuerzo de programación generalmente costoso en recursos humanos.

2. Hojas de cálculo

El principal problema que aqueja a las hojas de cálculo en el contexto de modelos de información multidimensional es que cuando la información excede cierto límite tienden a saturarse. En particular, los avances que se tienen sobre los sistemas de cómputo han logrado un aumento constante sobre el tamaño de las hojas de cálculo que se pueden manejar. No obstante, esta misma razón ha hecho que la información que se tenga que analizar sea más. Es razonable predecir en las tendencias que las hojas de cálculo en el mediano plazo no serán útiles en el manejo de volúmenes considerables de información.

3. Bases de datos

Recientemente, las organizaciones han dejado de utilizar sistemas de manejo de bases de datos relacionales (RDBMS, por sus siglas en inglés) para toda la gama de sus aplicaciones de bases de datos. Ha llegado a ser cada vez más aparente la existencia de ciertas categorías de aplicaciones que no son servidas adecuadamente por los RDBMS. Como ejemplo de esto, la compañía Oracle recientemente comenzó a comercializar un sistema totalmente nuevo para el manejo de aplicaciones de multimedios.

Otra categoría de aplicaciones es el procesamiento analítico en línea (OLAP, por sus siglas en inglés). Las aplicaciones OLAP tienen características de consulta y tiempo de respuesta que las posicionan en un lugar aparte de las aplicaciones de procesamiento de transacciones en línea (OLTP, por sus siglas en inglés). Los servidores especializados de OLAP están diseñados para proveer a los analistas el tiempo de respuesta y las capacidades funcionales de los sistemas de computación personal con el soporte a múltiples usuarios y grandes bases de datos que requieren.

Los sistemas RDBMS y sus aplicaciones asociadas tienen muchas limitaciones cuando son requeridas a realizar procesamiento de información en línea, entre ellas, un manejo de relaciones multidimensionales ineficiente, capacidades analíticas propietarias muy limitadas y consolidación de información engorrosa. Estas limitantes se detallan a continuación.

a) Manejo de relaciones multidimensionales

Estructura bidimensional. Las bases de datos relacionales son fundamentalmente incapaces de manejar estructuras de información multidimensional. En cambio, deben almacenar la información en tablas bidimensionales interrelacionadas que deben ser representadas como estructuras multidimensionales.

Confusión del usuario final con las herramientas de consulta SQL. Un gran conjunto de tablas relacionales representando múltiples dimensiones es difícil para el usuario final de entender y usar. Los usuarios deben imaginar cuál información se encuentra en qué tabla; cómo están ligadas las tablas entre sí; y cómo construir la consulta SQL adecuada que realizarán la unión de las tablas.

Uniones múltiples y desempeño pobre llegan a ser un problema. Los usuarios deben preocuparse en eficientar las consultas para obtener un desempeño aceptable. Algunos cuantos usuarios realizando consultas OLAP pueden llevar a sistemas incluso muy grandes a su punto de saturación.

Las bases de datos denormalizadas consumen los recursos. Debido a que uniones múltiples absorben costosos recursos de las máquinas y son difíciles de entender por los usuarios, la mayoría de los productores de RDBMS recomiendan a los usuarios que creen bases de datos especiales denormalizadas para el procesamiento de análisis. Los principales problemas que esta práctica acarrea son los siguientes:

- Consumo de los recursos del sistema
- Desempeño menor
- Creación de bases de datos mayores debido a la esparcidad de los datos
- Mayor entrada y salida de unidades de almacenamiento y construcción de índices

Las alternativas presentadas normalmente a los usuarios son la adquisición de equipos más poderosos y costosos, o el uso de procesamiento en paralelo de consultas. Sin embargo, la mayoría de las consultas no pueden ser divididas en trabajos paralelos, por lo que esta alternativa no tiene beneficios generalizados.

b) Capacidad de análisis

Los RDBMS no tienen las capacidades inherentes para manejar las operaciones OLAP como consolidaciones, "drill-down", etcétera. Los RDBMS tienen un conjunto de funciones limitados que sólo pueden operar en una dirección.

Limitaciones funcionales. El lenguaje SQL puede realizar operaciones de suma y promedio (*SUM* y *AVG*) en renglones (registros) pero no a través de columnas. Los usuarios pueden crear expresiones que representan diferentes campos en un registro en una sola tabla, pero esta expresión no puede hacer referencia a otras tablas a menos que se haya creado primero una vista.

Herramientas para usuario final. Como resultado de la limitación anterior, las operaciones que van más allá de las capacidades de la herramienta deben ser manejadas por la aplicación del usuario final. Esto representa varios problemas:

- Programación y personalización
- Problemas de redes locales y desempeño de los sistemas
- Transferencia de grandes cantidades de información
- Capacidad de los sistemas

c) Consolidación de información en RDBMS

El procesamiento analítico en línea requiere información consolidada, pero las bases de datos relacionales almacenan la información en forma detallada. Esto tiene los siguientes problemas:

- Requieren recursos de cómputo de forma intensa
- Requieren mucho tiempo de procesamiento
- No son amigables al usuario
- La actualización de la información interfiere con la operación normal
- Acceso del usuario restringido

D. SISTEMAS DE PROCESAMIENTO ANALÍTICO EN LÍNEA

El procesamiento analítico en línea está primordialmente involucrado con la consulta y agregación de grandes grupos de información diversa. Al contrario de las aplicaciones OLTP, las aplicaciones OLAP involucran muchos elementos de información (frecuentemente miles o incluso millones) que están involucrados entre sí mediante complejas relaciones. El objetivo de un sistema OLAP es el análisis de estas relaciones y la búsqueda de patrones, tendencias y condiciones de excepción.

1. Las 12 reglas de OLAP

El padre de la revolución relacional, el Dr. E. F. Codd, acuñó el término "Procesamiento analítico en línea" (OLAP, por sus siglas en inglés) en un reporte de agosto de 1983. Gracias al éxito previo en su definición del modelo relacional, el Dr. Codd especificó 12 reglas para OLAP. Acusó al SQL y los RDBMS actuales por su pobre soporte para un análisis flexible y promovió la necesidad de servidores de análisis especializados. A continuación se presentan las 12 reglas del OLAP. Cabe señalar que en la presente tesis no se pretende cubrir todos los aspectos señalados por las 12 reglas, sino utilizarles como una pauta de acción en las tareas que sean pertinentes.

a) Vista conceptual multidimensional.

La información desde el punto de vista del usuario es multidimensional en naturaleza. Por esto, los modelos OLAP deben ser multidimensionales. Esta vista multidimensional facilita el análisis y diseño de modelos, así como cálculos interdimensionales e intradimensionales, a través de un modelo analítico intuitivo.

b) Transparencia

El hecho de que OLAP es parte del producto del usuario debe ser transparente para él. El producto OLAP debe existir en una arquitectura de sistemas abierta y ser capaz de que la herramienta analítica sea utilizada o insertada en donde el usuario lo desee sin afectar de manera adversa a la funcionalidad del servidor de aplicaciones.

Adicionalmente, el usuario no debe preocuparse por el origen de la información, así haya sido extraída de un ambiente homogéneo o heterogéneo de bases de datos.

c) Accesibilidad

La herramienta OLAP debe estructurar su propio esquema lógico al almacenamiento de datos físicos heterogéneos, acceder a la información y realizar cualquier conversión necesaria para presentar una sola vista, coherente y consistente al usuario. Más aún, la herramienta y no el usuario debe preocuparse acerca de dónde o de qué tipo de sistemas proviene la información.

d) Desempeño consistente de ejecución de reportes

El usuario de OLAP no debe observar una degradación del desempeño en los reportes cuando aumenta el número de dimensiones.

e) Arquitectura Cliente/Servidor

La mayor parte de los sistemas que requieren OLAP almacenan su información en *Mainframes* y son accedidos vía computadoras personales. Es mandatorio pues, que los productos OLAP operen en un ambiente cliente/servidor. Para este fin, es imperativo que el componente servidor de las herramientas OLAP sea suficientemente inteligente para que clientes variados puedan ser integrados con un mínimo de esfuerzo y programación de integración.

f) Dimensionalidad Genérica

Cada dimensión de información debe ser equivalente en su estructura y capacidad operacional. Debe existir una sola estructura lógica para todas las dimensiones. Asimismo, cualquier función aplicada a una dimensión debe poder aplicarse a otra dimensión. La estructura básica de información, fórmulas y formato de reportes no debe estar predispuesta hacia ninguna dimensión de información.

g) Manejo Dinámico de Matrices Esparcidas

La estructura física del servidor OLAP debe manejar óptimamente matrices esparcidas. Una matriz esparcida es aquella en la que no todo bloque o celda en la matriz contiene datos. Cuando puesto a usar una matriz esparcida, el sistema debe ser capaz de deducir la distribución de los datos y la forma de almacenarla eficientemente. La herramienta OLAP debe poseer alguna técnica para la configuración de la estructura de la información.

h) Soporte a multiusuarios

Las herramientas deben proveer acceso simultáneo (consulta y escritura), integridad y seguridad para soportar usuarios que trabajen concurrentemente con el mismo modelo analítico o creen diferentes modelos de la misma información.

i) Operaciones dimensionales cruzadas irrestrictas

En el análisis de información multidimensional, todas las dimensiones son creadas y tratadas de igual forma. Por ejemplo, el usuario debe poder realizar las mismas acciones en diferentes dimensiones. La herramienta OLAP debe manejar cálculos entre dimensiones y no requerir que el usuario defina cuáles operaciones serán éstas.

j) Manipulación de datos intuitiva

La manipulación de los datos, como la reorientación de la ruta de consolidación, la realización del "*drill-down*" a través de las columnas y renglones o la expansión de la información, debe realizarse a través de una acción directa en las celdas del modelo analítico. Esta manipulación no debe requerir el uso de un menú o múltiples viajes a través de la interface al usuario. Las dimensiones definidas en el modelo analítico deben poseer toda la información que el usuario necesita para ejecutar cualquier acción inherente.

k) Reportes flexibles

Usando el servidor de OLAP y sus herramientas, un usuario debe manipular, analizar y observar la información de cualquier manera que lo desee, incluyendo la colocación de las columnas, renglones y celdas cerca las unas de las otras o arreglarlas en grupos lógicos. Las facilidades para realización de reportes deben emular esta flexibilidad y presentar información sintetizada de cualquier manera que el usuario quiera verla.

1) Dimensiones y niveles de agregación ilimitados

Investigación acerca de cuántas dimensiones posiblemente un modelo analítico requiere muestra que un usuario puede requerir hasta 19 dimensiones concurrentes. Cualquier herramienta OLAP debe ser capaz de manejar al menos 15 y preferentemente 20 dimensiones en un modelo analítico común. Asimismo, cada una de estas dimensiones genéricas debe ser capaz de manejar un número esencialmente ilimitado de niveles de agregación definidos por el usuario en cualquier ruta de consolidación.

2. Sistemas de análisis de información multidimensional

En la actualidad, existen algunos sistemas de análisis de información multidimensional, y los hay de diferentes capacidades. Como ejemplos notables son los productos tales como Improv, de Lotus; Commander Prism, de Comshare; y Essbase, de Arbor Software. Cada uno de ellos tiene ciertas deficiencias y virtudes. Estos productos se clasifican en dos grupos principales, que son los servidores de bases de datos multidimensionales, y las hojas de cálculo multidimensionales.

Los servidores de bases de datos multidimensionales (como Essbase, ProdeaBeacons) son sistemas que brindan la capacidad de realizar aplicaciones con acceso a la información que en ellas se almacena. Sin embargo, estos sistemas tienen dos desventajas muy importantes:

- El costo de estos sistemas es muy alto, generalmente en el orden de los cientos de miles de dólares. Esto los hace inaccesibles para la comunidad científica.

- Las aplicaciones deben realizarse en sistemas propietarios, como los sistemas de información ejecutivos (EIS, por sus siglas en inglés), que no poseen la flexibilidad y no son tan robustos como las herramientas de desarrollo normales. Esto implica la compra de estas herramientas y el aprendizaje de su uso.

Las hojas de cálculo multidimensionales son herramientas más comunes (como Lotus Improv, Commander Prism, Advance) que a su vez presentan las siguientes deficiencias:

- No proveen de alguna interface para el desarrollo de aplicaciones que utilicen su información. Es decir, que el uso que se les puede dar está limitado al que tiene su propia interface al usuario.
- Se alcanza su saturación rápidamente. Con unas cuantas dimensiones o al cruzar la barrera de algunas decenas de megabytes, el sistema se satura.

II. ANÁLISIS Y DISEÑO DEL SISTEMA

A. DEFINICIÓN DEL PROBLEMA

Analizados los antecedentes, se pretende realizar un sistema que sea capaz del almacenamiento, análisis y consulta de modelos de información multidimensional.

Asimismo, se pretende que el sistema entregue esta funcionalidad para el desarrollo de nuevas aplicaciones que utilicen las capacidades de este sistema sin tener que desarrollar toda la maquinaria necesaria para lograr tal efecto. Es decir, que sirva como herramienta de desarrollo de aplicaciones que utilicen modelos de información multidimensional.

B. JUSTIFICACIÓN DE LA TESIS

Es posible asegurar un mercado importante en la comunidad de sistemas de información para un producto que cumpla con estas características dado que no existen en el mercado paquetes que cubran este segmento, mismo que existe, vista la proliferación en tiempos recientes de productos de análisis de información multidimensional. Ninguno de ellos cumple con el requerimiento de ser un sistema de desarrollo abierto, sino que más bien, obligan al usuario a utilizar el conjunto completo de aplicaciones ofrecidos por el fabricante, sin que éste permita el desarrollo viable de sistemas con herramientas de programación ni la interacción directa con la base de datos.

Como su puede observar a partir de lo escrito en el capítulo anterior, existe la necesidad de un sistema de procesamiento de información multidimensional, con las siguientes características.

- Procesamiento de información multidimensional
- Orientado a desarrollo de aplicaciones que requieran análisis de información multidimensional
- Procesamiento analítico en línea, con sus consiguientes restricciones, de acuerdo a las reglas de OLAP
- Plataforma abierta
- Capacidad de correr en computadoras tipo PC
- Tiempo de respuesta aceptable

De acuerdo a los requerimientos anteriores, se considera factible la realización de un sistema con las siguientes características:

- El sistema funcionará en MS-Windows 95

- Se desarrollará con las herramientas *Borland Pascal with Objects 7.0* y *Borland Delphi*. La programación se realizará de forma tal que la necesidad de volver a escribir código se mantenga en un mínimo
- Se utilizarán las técnicas de programación orientada a Objetos (OOP, por sus siglas en inglés)
- Se proveerá al usuario de componentes DCC¹ para el desarrollo de sus aplicaciones propietarias.

El sistema, sin embargo, dado el alcance del proyecto se restringirá a ciertas áreas primordiales de un sistema OLAP, de acuerdo a las doce reglas vistas anteriormente:

- ✓ Vista multidimensional conceptual
- ✓ Transparencia. Esto será logrado al proporcionar herramientas de desarrollo. Éstas tendrán la tarea de unirse al sistema de información multidimensional. El usuario no debe tener conocimientos de esto
- ✓ Accesibilidad
- ✓ Desempeño consistente
- ✓ Manejo dinámico de matrices esparcidas
- ✓ Dimensionalidad genérica
- ✓ Arquitectura Cliente - Servidor²

¹ Los componentes DCC (Delphi Custom Controls) son objetos que cumplen con el estándar COM de Microsoft. Esta es la base para otras plataformas, como *ActiveX* y el producto *MS Transaction Server* Son generados por la herramienta *Borland Delphi*, y permiten todos los atributos de la programación orientada a objetos. Por lo mismo, el usuario tiene la posibilidad de utilizar estos objetos en el desarrollo de un sistema propio, y modificar o aumentar el comportamiento de los objetos. Asimismo, estos objetos pueden ser compilados de tal forma que resulten objetos que cumplen con el estándar *ActiveX* de Microsoft. Por lo mismo, son compatibles con la mayor parte de los sistemas de desarrollo disponibles actualmente. Estos objetos pueden ser utilizados en la implementación de aplicaciones de Internet, por lo que una aplicación que los utilice puede ser ejecutada en cualquier navegador de Internet que cumpla con el estándar.

- ✓ Soporte a multiusuario³
- ✓ Operaciones dimensionales cruzadas irrestrictas
- ✓ Manipulación de datos intuitiva. Ésta será lograda al desarrollar la aplicación del usuario.
- ✓ Creación de reportes flexible
- ✓ Niveles de agregación y dimensiones ilimitadas

Como se puede observar, no se pretende realizar una aplicación universal de manejo de información multidimensional, sino proporcionar las herramientas necesarias para el desarrollo de aplicaciones propias.

² La arquitectura cliente - servidor se refiere a la existencia de dos niveles en la programación. Una, el cliente, que es encargado de interactuar con el usuario, y realizar los requerimientos necesarios al servidor para que sus órdenes sean cumplidas. El servidor no tiene conocimiento del usuario, y debe poder controlarse completamente a través de un cliente. Esto permite que a un sistema fácilmente se le agreguen capacidades de multiusuario, o de seguridad. Lo anterior también implica que hay diferentes tipos de clientes, cada uno con una función específica. Para el usuario, representa la ventaja de tener una sola verdad de la información para todos los clientes.

Sin embargo, la arquitectura cliente - servidor no implica que el sistema esté habilitado para el uso de varias personas a través de una red. Esta es una extensión natural y bastante normal a la arquitectura, pero no suficiente ni necesaria.

Esta tesis no pretende el realizar un sistema de comunicaciones por red que permita un sistema multiusuario. Existen varias herramientas que hacen transparente la comunicación vía red, que pueden integrarse al sistema. Sin embargo, son herramientas costosas por lo que no se tienen disponibles en este trabajo. No obstante, debido a que el resultado se basa en objetos que cumplen con el estándar COM, es posible suponer que el sistema se ejecuta sin problemas en este entorno sin cambio alguno, aunque la prueba no se haya realizado.

³ Aunque no se realizó la prueba de soporte a varios usuarios simultáneos, se supone que ésta debe realizarse sin problemas gracias al estándar COM.

Es importante notar que los puntos anteriores tienen como consecuencia el desarrollo de un sistema que en su mayor parte estará formado por componentes u objetos que entregan la capacidad de realizar una aplicación propietaria. Sin embargo, debe proveerse una aplicación que permita el desarrollo de los modelos de información. Con esto se logra la separación de los pasos del diseño de aquellos del uso de la aplicación.

C. HERRAMIENTAS UTILIZADAS

El problema presentado puede resolverse de manera exitosa con la ayuda de ciertas herramientas de desarrollo de sistemas. Se plantea la solución con la ayuda de los paquetes *Borland Pascal with Objects (BP)* y *Borland Delphi (Delphi)*. Estas son dos herramientas basadas en el lenguaje de programación Pascal.

El primero, BP, es un lenguaje que brinda el beneficio de la programación orientada a objetos (OOP, por sus siglas en inglés). Estas son la encapsulación, la herencia y el polimorfismo. Gracias a las técnicas de la OOP, se obtienen varios beneficios, entre ellos, el reuso del código y una fácil actualización de versiones de los sistemas. Esto es importante, pues evita el volver a escribir código inútilmente. Si a esto se agrega una planeación cuidadosa del sistema, es posible el cambio de plataformas sin un esfuerzo considerable.

El segundo, Delphi, es una herramienta de desarrollo rápido de aplicaciones (RAD, por sus siglas en inglés), que presenta una evolución en su área tal que permite avocarse a la funcionalidad de la aplicación, y obtener una presentación de manera rápida. Es posible utilizar código escrito en BP de manera eficiente, con sólo algunos cambios menores.

D. ARQUITECTURA DEL SISTEMA

Este sistema tiene como principal finalidad el proveer al usuario una herramienta con la que pueda desarrollar un sistema de análisis de información multidimensional. Se eligió a Borland Delphi como herramienta para el usuario por tres características:

- Desempeño excelente de la aplicación resultante
- Facilidad de desarrollo de aplicaciones complejas
- Apertura sobresaliente a desarrollo de componentes adicionales

1. Arquitectura Interna

a) Diagrama general

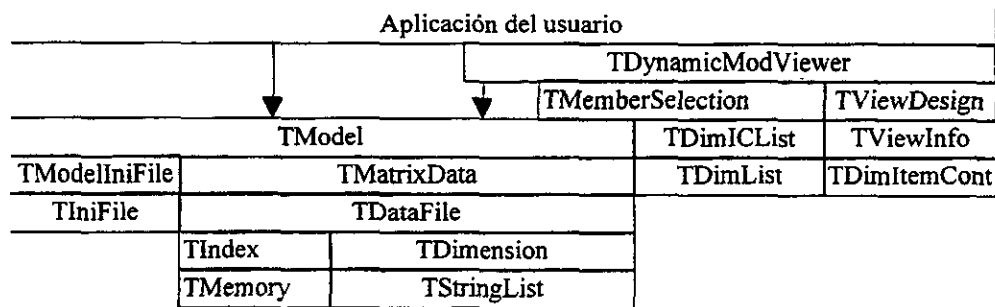


Figura 9. Diagrama general del sistema

2. Subsistema de almacenamiento de datos

a) Definición del problema

El almacenamiento de matrices esparcidas no es una tarea trivial, por lo que se debe utilizar un método que tenga la capacidad de realizar la tarea con un mínimo de efectos secundarios. El problema reside en que no todas las celdas de la matriz se almacenan, y el disponer de espacio para todas las posibles celdas sería un desperdicio de espacio innecesario.

b) Solución empleada

Recuérdese que existen dimensiones que se saben son densas. Esto significa que la existencia del valor en uno de sus miembros es condición suficiente para garantizar la existencia de algún valor en todos sus demás miembros. Esto es útil porque se puede entonces saber que la existencia de celdas en la matriz será determinada por las dimensiones esparcidas.

Se llama a la combinación de valores definidos por la combinación de todos los miembros de todas las dimensiones densas "bloque". Si una sola de sus celdas existe, es posible entonces garantizar que todas las demás celdas existen. Esto indica que es posible almacenar la matriz en bloques, los cuales estarán siempre llenos de información. El número de bloques que existan depende de las combinaciones entre las dimensiones esparcidas que ocurran. El almacenamiento por bloques entonces, permite asegurar que la matriz será almacenada sólo en las partes que sea necesario, y no habrá desperdicio de espacio.

Debido a que no se sabe cuáles son los bloques que existan en algún momento, es necesario proveer algún método de almacenamiento de bloques en cualquier orden. Esto se puede lograr por medio de un "índice" que controle el almacenamiento, creación y destrucción de los bloques.

El índice es una parte fundamental en el acceso a la información. Por lo mismo, se realiza un mapeo del archivo de índice al espacio de memoria de la aplicación de manera que el acceso a la información que en él se encuentra sea el más rápido posible. En Windows 95 el espacio disponible para esta memoria es de hasta 2GB, por lo que no representa ningún problema.

Cada bloque entonces será almacenado en el disco duro sólo en el momento en el que sea necesario, es decir, cuando se escriba algún dato en una celda perteneciente a este bloque o en el momento en el que una celda de este bloque sea escrita como resultado de un cálculo. La estructura de un bloque es la siguiente:

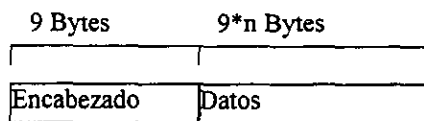


Figura 10. Diagrama de un bloque

Como se observa, el tamaño de un bloque en el disco duro es de 9 Bytes * (n+1), donde n es el número de celdas del bloque. El encabezado del bloque contiene la siguiente información:

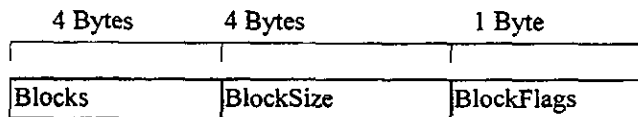


Figura 11. Diagrama del encabezado de un bloque

Al crear un nuevo modelo, el archivo de datos no contiene ningún bloque. Al escribirse datos en alguna combinación, se crea el bloque. Supóngase el siguiente ejemplo: existen dos dimensiones densas, variables y tiempo; y dos esparcidas, clientes y productos, como se ilustra:

| Variables | Tiempo | Regiones | Cientes |
|-----------|-------------|----------|----------|
| Ventas | Total anual | TotalR | TotalC |
| Precio | Ene | Norte | Cliente1 |
| Unidades | Feb | Sur | Cliente2 |
| | Mar | Este | Cliente3 |
| | Abr | Oeste | Cliente4 |
| | May | | |
| | Jun | | |
| | Jul | | |
| | Ago | | |
| | Sep | | |
| | Oct | | |
| | Nov | | |
| | Dic | | |

Figura 12. Dimensiones de la matriz de ejemplo

Al capturarse información en algún bloque, éste se creará. Los bloques son formados por las combinaciones de la dimensión regiones y clientes. El bloque tendrá un tamaño de $(5*5)+1 = 26$ celdas, que corresponden a $26*8 = 208$ Bytes. La estructura del bloque se muestra en la siguiente figura:

| | | | | | |
|------------|-----------------------|-----------------------|-------------------------|---------------|-----------------|
| Encabezado | Ventas Total anual | Precio Total anual | Unidades Total anual | Ventas Ene | Precio Ene |
| ... | Precio Nov | Unidades Nov | Ventas Dic | Precio Dic | Unidades Dic |

Figura 13. Estructura de bloque en el ejemplo

Supóngase que primero se captura información del "cliente1" en la región "norte"; después se captura información del "cliente4" en la región "sur"; y así sucesivamente. Por lo tanto, el índice quedará como se muestra:

| | Total | Norte | Sur | Este | Oeste |
|----------|-------|-------|-----|------|-------|
| Total | 0 | 0 | 0 | 0 | 0 |
| Cliente1 | 0 | 1 | 0 | 3 | 0 |
| Cliente2 | 0 | 5 | 0 | 0 | 4 |
| Cliente3 | 0 | 0 | 0 | 0 | 0 |
| Cliente4 | 0 | 0 | 2 | 0 | 0 |

Figura 14. Ejemplo de un índice de bloques

Las celdas del índice marcadas con cero son bloques que no existen. Los bloques marcados con un número son aquéllos que existen, y el número contenido es el número de bloque en el archivo de datos. Es decir, que el orden en el que se capturan es el mismo que el orden en el que se escriben. El archivo de datos contendrá los siguientes bloques:

| | | | | |
|----------|----------|----------|----------|----------|
| Cliente1 | Cliente4 | Cliente1 | Cliente2 | Cliente2 |
| Norte | Sur | Este | Oeste | Norte |

Figura 15. Archivo de datos del ejemplo

Así, el archivo de datos es de $208 \text{ Bytes} * 5 \text{ Bloques} = 1040 \text{ Bytes}$, contra los $208 * 5 * 5 = 5200 \text{ Bytes}$ que hubiesen sido utilizados para almacenar esta información de haber creado todo el espacio necesario para almacenarla. Conforme aumenta el número de dimensiones esparcidas, se incrementa la utilidad de este modelo de almacenamiento.

Al realizar el cálculo de este modelo, se crearán nuevos bloques, en los totales de las dimensiones esparcidas. En este cálculo, el orden de las operaciones es el siguiente: primero se calcula la dimensión de regiones, pues es la primera en la estructura. El único cálculo a realizar es el de "TotalR". Debido a que aún no hay información en "TotalC", no hay resultados en sus celdas. El orden en el que se calcula "TotalR" es: "Cliente1", "Cliente2", "Cliente3", "Cliente4" y por último "TotalC". El resultado del archivo de índice es ilustrado de la siguiente forma:

| | TotalR | Norte | Sur | Este | Oeste |
|----------|--------|-------|-----|------|-------|
| TotalC | 0 | 0 | 0 | 0 | 0 |
| Cliente1 | 6 | 1 | 0 | 3 | 0 |
| Cliente2 | 7 | 5 | 0 | 0 | 4 |
| Cliente3 | 0 | 0 | 0 | 0 | 0 |
| Cliente4 | 8 | 0 | 2 | 0 | 0 |

Figura 16. Índice de bloques del ejemplo tras el cálculo de la dimensión regiones

Nótese que no se generó un bloque para la combinación "TotalR" de "Cliente3", pues este no tiene información. Tras haber hecho este cálculo se procede a hacer el cálculo de la dimensión clientes, en donde el único cálculo a realizar es el de "TotalC". El orden en el que se calcula es semejante al ejemplo anterior: "Norte", "Sur", "Este", "Oeste" y por último, "TotalR". Así, el archivo de índice queda ilustrado como sigue:

| | <i>TotalR</i> | <i>Norte</i> | <i>Sur</i> | <i>Este</i> | <i>Oeste</i> |
|----------|---------------|--------------|------------|-------------|--------------|
| TotalC | 13 | 9 | 10 | 11 | 12 |
| Cliente1 | 6 | 1 | 0 | 3 | 0 |
| Cliente2 | 7 | 5 | 0 | 0 | 4 |
| Cliente3 | 0 | 0 | 0 | 0 | 0 |
| Cliente4 | 8 | 0 | 2 | 0 | 0 |

Figura 17. Índice de bloques del ejemplo tras el cálculo de la dimensión clientes

c) Funciones

El subsistema de almacenamiento realiza las funciones de almacenamiento, control y mantenimiento de los bloques que componen a las matrices esparcidas. Provee de una abstracción de su funcionamiento de manera tal que los otros módulos no requieren conocer el funcionamiento del módulo para utilizarlo.

Aún más importante, provee de una abstracción del almacenamiento por bloques y de la esparcidad de las matrices. De esta manera, a la vista de los otros módulos, la matriz aparece como completamente densa, y no representa problemas para su programación. En su funcionamiento interno, sin embargo, sólo se almacena la información que se requiere.

3. Subsistema de almacenamiento del modelo analítico

a) Definición del problema

El modelo contiene datos no numéricos que son necesarios. Como se mostró en la sección anterior, el archivo de datos sólo contiene los números. Por lo mismo, es necesaria una estructura en donde se almacenen las descripciones de los miembros de las dimensiones, su orden y sus atributos (densa o esparcida).

b) Solución empleada

Para el almacenamiento del modelo se utiliza un archivo tipo "INI" ⁴ en donde se definen todos los datos generales del modelo en un archivo de estructura mostrada en la siguiente tabla:

⁴ Los archivos tipo "INI" son archivos estándares utilizados de forma corriente en la plataforma MS Windows en todas sus versiones. Esta misma plataforma provee funciones específicas para su manejo, por lo que es posible utilizarlas de forma que no es necesaria la definición de una técnica de almacenamiento propia.

| <i>Sección</i> | <i>Identificador</i> | <i>Descripción</i> | <i>Ejemplo</i> |
|-------------------|-----------------------------|---|-----------------------------------|
| Model | Name | Nombre del modelo. Máximo 8 letras. | MAGNETO |
| Model | Description | Descripción del modelo | Análisis de magnetismo en sólidos |
| Model | Path | Ruta de almacenamiento de los archivos | C:\MAGNETO\ |
| Model | Data file | Archivo de datos | C:\MAGNETO\MAGNETO.PAG |
| Model | Index | Archivo de índice de bloques | C:\MAGNETO\MAGNETO.IDX |
| Model | Hasdata | Bandera de existencia de información | 0 |
| Model | Createddata | Bandera de existencia de archivo de datos | 1 |
| Dense Dimensions | <Nombre de las dimensiones> | Orden y nombre de dimensión | Variables=0 |
| Sparse Dimensions | <Nombre de las dimensiones> | Orden y nombre de la dimensión | Escenarios=3 |
| Views | <Nombre de la vista> | Ruta de acceso a la vista | C:\MAGNETO\Campos.VIW |
| Calcs | <Nombre del cálculo> | Ruta de acceso al cálculo | C:\MAGNETO\ONLYMAG.CLC |

Figura 18. Estructura del archivo de almacenamiento del modelo

De acuerdo con la tabla anterior, un archivo típico se ejemplifica en la siguiente

figura:


```

[MODEL]
NAME=Nombre
DESCRIPTION=Pruebas para la tesis
PATH=E:\MODELOS\SPARSE\
INDEX=E:\MODELOS\SPARSE\Model.IDX
HASDATA=0
CREATEDDATA=0
DATAFILE=E:\MODELOS\SPARSE\Model.DAT

[DENSE DIMENSIONS]
VARIABLE=0
TIEMPO=1

[SPARSE DIMENSIONS]
REGION=2
PRODUCTO=3

[VIEWS]
VISTA1=E:\MODELOS\SPARSE\ASD.VIW
VISTA2= E:\MODELOS \SPARSE\FIRST.VIW

[CALCS]
CALCALL= E:\MODELOS\SPARSE\CALCALL.CLC
OTROCALC=E:\MODELOS\SPARSE\OTRO.CLC

```

Figura 19. Ejemplo de archivo de almacenamiento del modelo

La estructura de cada dimensión se almacena en la ruta que se encuentra en el campo "PATH", en tres archivos diferentes. El nombre del archivo es el mismo que el de la dimensión. El archivo <Nombre>.DIM contiene los nombres de los miembros. El archivo <Nombre>.DES contiene las descripciones de los miembros. El archivo <Nombre>.MAT contiene las fórmulas de los miembros. Estos son archivos ASCII⁵, en los que la generación del miembro se encuentra definida por el número de caracteres tabulador que se encuentran antecediéndole. Un ejemplo de estos archivos, para la dimensión variables es el siguiente, en donde el carácter ">" representa al tabulador:

⁵ Archivos planos formados exclusivamente por caracteres numéricos, alfabéticos, tabuladores y de cambio de renglón.

| <i>Archivo Variable.DIM</i> | <i>Archivo Variable.DES</i> | <i>Archivo Variable.MAT</i> |
|-----------------------------|-----------------------------|------------------------------|
| MOP | M.O.P. | Utilidades % Ingresos |
| >UTIL | >Utilidades | > Ingresos - Costos - Gastos |
| >>ING | >>Ingresos | >>@SUMCHILDS(Ingresos) |
| >>>VTA | >>>Ventas | >>> |
| >>>OTROS | >>>Otros Ingresos | >>> |
| >>CTO | >>Costos | >>@SUMCHILDS(Costos) |
| >>>CTODIR | >>>Costos indirectos | >>> |
| >>>CTOIND | >>>Costos directos | >>> |
| >>GTO | >>Gastos | >> |

Figura 20. Ejemplo de archivos de estructura de dimensión

c) Funciones

El subsistema de almacenamiento del modelo actúa como un contenedor de los datos de los demás subsistemas (cálculo, almacenamiento e interface al usuario) y por lo mismo, es el encargado de cargar y grabar a disco los datos pertenecientes a los demás subsistemas cuando es creado o destruido. Asimismo, contiene todos los datos almacenados en el archivo de estructura del modelo y los archivos de estructura de las dimensiones.

4. Subsistema de cálculo del modelo

a) Definición del problema

El cálculo de un modelo multidimensional esparcido representa un reto considerable, debido a los siguientes factores:

- Se deben ejecutar operaciones matemáticas diversas entre los miembros de las dimensiones
- Es indispensable proveer al usuario una forma de realizar cálculos explícitos, con los cuales cambiar el orden de los cálculos; calcular una sola parte de la base de datos y realizar cálculos cualquier número de veces
- Se deben poder realizar operaciones dentro de los bloques y entre bloques

- La creación de los bloques debe ser determinada por la existencia de los bloques de los que dependen, es decir, que sólo debe ser creado un bloque cuando realmente sea necesario.

La conjunción de los factores anteriores da como resultado que la ejecución de un cálculo sea una operación particularmente complicada. En especial, consentir al usuario el realizar el cálculo de una dimensión sobre sólo algunos miembros de las demás dimensiones (sean éstas densas o esparcidas) y brindarle la capacidad de realizar el cálculo de sólo algunos miembros de una dimensión complica de manera substancial los algoritmos de cálculo.

b) Solución empleada

Se dio solución a la problemática expuesta creando estructuras u objetos "bloque" inteligentes, es decir, que contienen en ellos mismos la lógica para grabar en disco sus datos en el momento en que sea necesario; que sepan cómo hacer operaciones matemáticas dentro de ellos y que sepan cómo hacer operaciones matemáticas entre ellos. Estos objetos son de tipo *TBlock*.

Para este efecto se creó una estructura semejante a un *buffer*, en donde los bloques son cargados de disco al *buffer*, el cual tiene una capacidad máxima de bloques que puede almacenar. Esta estructura tiene una arquitectura "último usado - primero en salir", es decir, que en el momento en que se necesite el espacio para un nuevo bloque, el último utilizado será el primero en salir del *buffer* y grabarse a disco. De esta manera, no será necesario el leer de disco los datos cada vez que se necesiten. Esto se hace debido a que la probabilidad de volver a usar un bloque que acaba de ser usado es muy alta. De esta manera, se minimiza el acceso a disco para los cálculos. Este es el clásico esquema llamado "cache".

Para resolver el problema que representa el hacer cálculos sólo sobre algunos miembros, se utilizan "mapas de bits" para representar aquellos miembros que habrán de ser calculados. Se presenta un ejemplo a continuación.

Supóngase que se quiere hacer sobre el ejemplo expuesto en la sección anterior un cálculo de la variable ventas sólo sobre los periodos "Ene", "Feb" y "Mar"; y sólo sobre las regiones "TotalR", "Norte" y "Sur". El primer paso es definir dentro del bloque aquellas celdas que serán resultados. El bloque puede representarse de la siguiente manera:

| | Total Anual | Ene | Feb | Mar | Abr | May | Jun | Jul | Ago | Sep | Oct | Nov | Dic |
|----------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Ventas | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| Precio | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| Unidades | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 |

Figura 21. Ejemplo de estructura de bloque para determinación de un cálculo

Donde el número en cada celda es el número de la celda vista en forma secuencial dentro del bloque. Como es sabido que sólo se quiere calcular "Ventas" sobre "Ene", "Feb" y "Mar", es posible hacer una tabla en la que se muestren las celdas que serán calculadas:

| | Total Anual | Ene | Feb | Mar | Abr | May | Jun | Jul | Ago | Sep | Oct | Nov | Dic |
|----------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Ventas | | x | x | x | | | | | | | | | |
| Precio | | | | | | | | | | | | | |
| Unidades | | | | | | | | | | | | | |

Figura 22. Mapa de bits del conjunto de celdas a calcular en el ejemplo de un cálculo

La anterior tabla puede ser almacenada en una estructura formada por "ceros" y "unos", en donde los "unos" representan las celdas a calcular. De esta manera, la estructura ocupará $13 \times 3 = 39$ bits = 5 Bytes. Teniendo este mapa de bits, habrán de calcularse los miembros de la dimensión variables para las celdas marcadas independientemente de su posición. De igual manera, es posible construir un mapa de bits para los bloques a calcular.

En la siguiente figura se muestran los bloques que existen originalmente:

| | TotalR | Norte | Sur | Este | Oeste |
|----------|--------|-------|-----|------|-------|
| TotalC | 0 | 0 | 0 | 0 | 0 |
| Cliente1 | 0 | 1 | 0 | 3 | 0 |
| Cliente2 | 0 | 5 | 0 | 0 | 4 |
| Cliente3 | 0 | 0 | 0 | 0 | 0 |
| Cliente4 | 0 | 0 | 2 | 0 | 0 |

Figura 23. Bloques existentes al inicio del cálculo en el ejemplo de un cálculo

Sin embargo, lo que se desea es determinar cuáles son los bloques que habrán de ser calculados. Para esto, es posible mostrar de forma secuencial el índice de bloques como se muestra en la siguiente figura:

| | TotalR | Norte | Sur | Este | Oeste |
|----------|--------|-------|-----|------|-------|
| TotalC | 0 | 1 | 2 | 3 | 4 |
| Cliente1 | 5 | 6 | 7 | 8 | 9 |
| Cliente2 | 10 | 11 | 12 | 13 | 14 |
| Cliente3 | 15 | 16 | 17 | 18 | 19 |
| Cliente4 | 20 | 21 | 22 | 23 | 24 |

Figura 24. Estructura secuencial del índice de bloques en el ejemplo

Los bloques que habrán de ser calculados son los pertenecientes a "TotalR", "Norte" y "Sur", por lo que es posible tener un mapa de bits resultante como el que se muestra:

| | <i>TotalR</i> | <i>Norte</i> | <i>Sur</i> | <i>Este</i> | <i>Oeste</i> |
|----------|---------------|--------------|------------|-------------|--------------|
| TotalC | x | x | x | | |
| Cliente1 | x | x | x | | |
| Cliente2 | x | x | x | | |
| Cliente3 | x | x | x | | |
| Cliente4 | x | x | x | | |

Figura 25. Mapa de bits de bloques a calcularse en el ejemplo

Debido a que no hay ninguna restricción sobre la dimensión clientes, se supone que habrá de calcularse sobre todos los miembros de esa dimensión. Cuando se realice el cálculo, entonces, sólo queda hacerlo sobre los bloques marcados que existan, es decir, aquellos que estén marcados con un "uno" en el mapa de bits y cuyo valor en el índice de bloques sea diferente a cero. Debido a que el cálculo se está realizando sobre una dimensión densa, ningún bloque se creará. Por el contrario, si el cálculo fuese en una dimensión esparcida, se crearían nuevos bloques, por lo que debe garantizarse la existencia de al menos un bloque del que sea dependiente el bloque resultado y que, al realizarse el cálculo, el bloque resultante contenga al menos una celda diferente de "Vacío".

5. Subsistema de cálculos personalizados

a) Definición del problema

El usuario debe tener la facilidad de almacenar cálculos personalizados. Estos son procedimientos predeterminados por él, que consisten de pasos que deben ejecutarse secuencialmente. El usuario debe poder seleccionar el subconjunto de los miembros que deben calcularse, y los miembros de las demás dimensiones sobre los que debe calcularse. Debe brindársele alguna forma de editar este procedimiento y de ejecutarlo.

b) Solución empleada

El usuario cuenta con el editor de cálculos, con el que se pueden crear, añadir, borrar y modificar líneas de un cálculo. Esto tiene el objeto de que el cálculo se ejecute línea por línea, de manera síncrona. El usuario puede modificar las líneas de manera que seleccione con precisión la operación que desea realizar. Las opciones que tiene disponibles se muestran en la siguiente tabla:

| <i>Dimensión a calcular</i> | <i>Opciones / miembros</i> | <i>Selección de miembros</i> | <i>Miembros fijos</i> | <i>Descripción</i> |
|-----------------------------|----------------------------|------------------------------|-----------------------|--|
| Todas | (Smart) | N/D | N/D | Calculará todos los miembros de todas las dimensiones |
| | (All members) | N/D | N/D | Calculará jerárquicamente todas las celdas del modelo |
| Dimensión | All | N/D | Normal | Calculará todos los miembros de la dimensión |
| | Single | Un miembro | Normal | Calculará sólo un miembro de la dimensión |
| | Multiple | Varios miembros | Normal | Calculará algunos miembros de la dimensión |
| | Child | Un miembro | Normal | Calculará los miembros hijos de un miembro de la dimensión |
| | Descendants | Un miembro | Normal | Calculará los descendientes de un miembro de la dimensión |

N/D: No disponible. Esta opción no se muestra dado que no tiene aplicación para el cálculo seleccionado

Figura 26. Opciones de líneas disponibles en un cálculo

En las líneas en las que es posible seleccionar los miembros de las otras dimensiones a los que aplica el cálculo, se presenta la opción, y se les llaman "miembros fijos". Existe una opción para calcular toda la base de datos, que es llamada "Calc all smart". Esta opción realiza el cálculo de todos los miembros que tienen alguna fórmula, pero sólo en los miembros de las otras dimensiones en los que aplica. Tómese el siguiente ejemplo para mostrar su utilidad:

Considérese el modelo de ejemplo utilizado en los anteriores subcapítulos. Este modelo tiene las dimensiones variables, periodos, clientes y regiones. Debido a que el orden de cálculo es: variables, periodos, regiones y clientes, es posible observar que los cálculos del miembro "Ventas" que se realicen sobre las celdas pertenecientes al miembro "Total anual" son inútiles, pues habrán de ser vueltas a escribir cuando se realice el cálculo del miembro "Total anual". Por lo mismo, es posible inferir cuáles son las celdas que deben ser calculadas en el cálculo de cada dimensión. Este orden de cálculo se muestra en la siguiente figura:

| | | TotalC | | | | Cliente1 | | | | | ... | | | |
|---------|----------|--------|-------|-----|-----|----------|-----|-------|-------|-----|-----|-----|-----|-----|
| | | Total | Anual | Ene | Feb | ... | Dic | Total | Anual | Ene | Feb | ... | Dic | ... |
| TotalR | Ventas | 4 | | 4 | 4 | | 4 | 3 | | 3 | 3 | | 3 | ... |
| | Precio | 4 | | 4 | 4 | | 4 | 3 | | 3 | 3 | | 3 | ... |
| | Unidades | 4 | | 4 | 4 | | 4 | 3 | | 3 | 3 | | 3 | ... |
| Region1 | Ventas | 4 | | 4 | 4 | | 4 | 2 | | 1 | 1 | ... | 1 | ... |
| | Precio | 4 | | 4 | 4 | | 4 | 2 | | 0 | 0 | ... | 0 | ... |
| | Unidades | 4 | | 4 | 4 | | 4 | 2 | | 0 | 0 | ... | 0 | ... |
| Region2 | Ventas | 4 | | 4 | 4 | | 4 | 2 | | 1 | 1 | ... | 1 | ... |
| | Precio | 4 | | 4 | 4 | | 4 | 2 | | 0 | 0 | ... | 0 | ... |
| | Unidades | 4 | | 4 | 4 | | 4 | 2 | | 0 | 0 | ... | 0 | ... |
| Region3 | Ventas | 4 | | 4 | 4 | | 4 | 2 | | 1 | 1 | ... | 1 | ... |
| | Precio | 4 | | 4 | 4 | | 4 | 2 | | 0 | 0 | ... | 0 | ... |
| | Unidades | 4 | | 4 | 4 | | 4 | 2 | | 0 | 0 | ... | 0 | ... |
| Region4 | Ventas | 4 | | 4 | 4 | | 4 | 2 | | 1 | 1 | ... | 1 | ... |
| | Precio | 4 | | 4 | 4 | | 4 | 2 | | 0 | 0 | ... | 0 | ... |
| | Unidades | 4 | | 4 | 4 | | 4 | 2 | | 0 | 0 | ... | 0 | ... |

Figura 27. Orden seguido en el cálculo inteligente del ejemplo

Como se observa en la figura, las celdas marcadas con un "cero" no requieren ser calculadas, pues no contienen ningún cálculo. Por ello, no representan ningún tiempo extra en el cálculo. Las celdas marcadas con un "uno" serán las que se calcularán en el turno correspondiente a la dimensión "variables". Las celdas que tienen un "dos" serán calculadas en el turno de la dimensión "periodos". Las marcadas con un "tres", lo harán en el turno de la dimensión "regiones". Por último, las marcadas con un "cuatro" serán calculadas en el turno de la dimensión "clientes". Como se observa, el mayor tiempo del cálculo estará ocupado por la última dimensión del modelo. Pero, recuérdese que éstas son típicamente dimensiones esparcidas, por lo que sólo los bloques realmente ocupados son los que se calcularán.

Estas estructuras de cálculo se almacenan en archivos tipo "Ini", en donde las secciones e identificadores del mismo se muestran en la siguiente figura:

| Sección | Identificador | Valor (Ejemplo) | Descripción |
|----------------|---------------|--|--|
| Calcule | Description | Calculo XYZ | Descripción del cálculo |
| | Lines | 6 | Número de líneas del cálculo |
| Line# | Dimension | (All) / Dimensión | Selecciona todas o alguna dimensión |
| | MemberOp | Child | Selecciona los hijos de un miembro de la dimensión |
| | | Descendants | Selecciona los descendientes de un miembro de la dimensión |
| | | All | Selecciona todos los miembros de la dimensión |
| | | Multiple | Selecciona sólo algunos miembros de la dimensión |
| | Single | Selecciona sólo un miembro de la dimensión | |
| LineXSelection | Miembro | Norte | Selección de los miembros fijos |

Figura 28. Formato del archivo de almacenamiento de cálculo

Al crear el archivo de cálculo, el usuario puede ejecutarlo. Para esta ejecución, el sistema procesa el archivo en dos pasos: en el primero, lo transforma en una estructura de tipo *TCalculate*, en el que se identifican todos los miembros que se especifican en el archivo, y es equivalente a una corrección de sintaxis. Los miembros se transforman a un identificador de 32 bits, que es el orden en el que se encuentran en la dimensión. Las selecciones de miembros se transforman en listas de identificadores de 32 bits. Las fórmulas, por último se transforman de cadenas de caracteres en estructuras binarias de tipo *TCompiledFunction*, que es una estructura que los objetos *TBlock* reconoce y puede ejecutar.

El segundo paso de esta transformación consiste en crear estructuras de tipo *TCompiledCalculate* para el cálculo de la matriz en sí. Esto es, crea caches de cálculo, y los mapas de bits necesarios para identificar los miembros a calcular.

6. Arquitectura al usuario

Al usuario se le presenta una arquitectura para el desarrollo de sistemas sin que tenga la necesidad de conocer las partes internas que le proveen de esta funcionalidad. Las partes que se le proveen para interactuar son dos objetos: el *Tmodel* y el *TDynamicModViewer*. Estos objetos son "Delphi Custom Controls" (DCC) que tienen un comportamiento semejante al de los *OLE custom controls (OCX)* de Microsoft, con algunas características extras (como la extensibilidad) que habilitan al usuario para añadir comportamientos específicos a los objetos provistos. Esto es particularmente útil, porque no se limita al usuario con un comportamiento predeterminado, sino que tiene la posibilidad de modificarlo a su gusto.

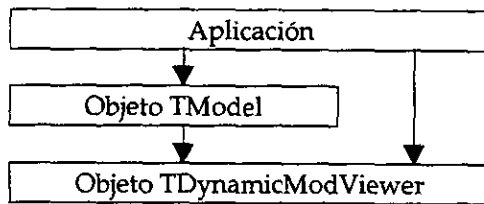


Figura 29. Arquitectura al usuario del sistema

Estos objetos están disponibles a través de la paleta de objetos que proporciona Delphi. Las propiedades y métodos que proveen cada uno de ellos son accesibles a través del inspector de objetos. Algunos objetos agrupan a otros, de manera que contengan su funcionalidad. Por ejemplo, el objeto *TModel* agrupa a los objetos *TMatrixData* (que es el que maneja el almacenamiento y cálculo del modelo), al objeto *TDimensions* (que contiene los nombres de los miembros, sus descripciones, fórmulas y atributos), etcétera. Las relaciones entre objetos del modelo multidimensional se muestran en la siguiente figura:

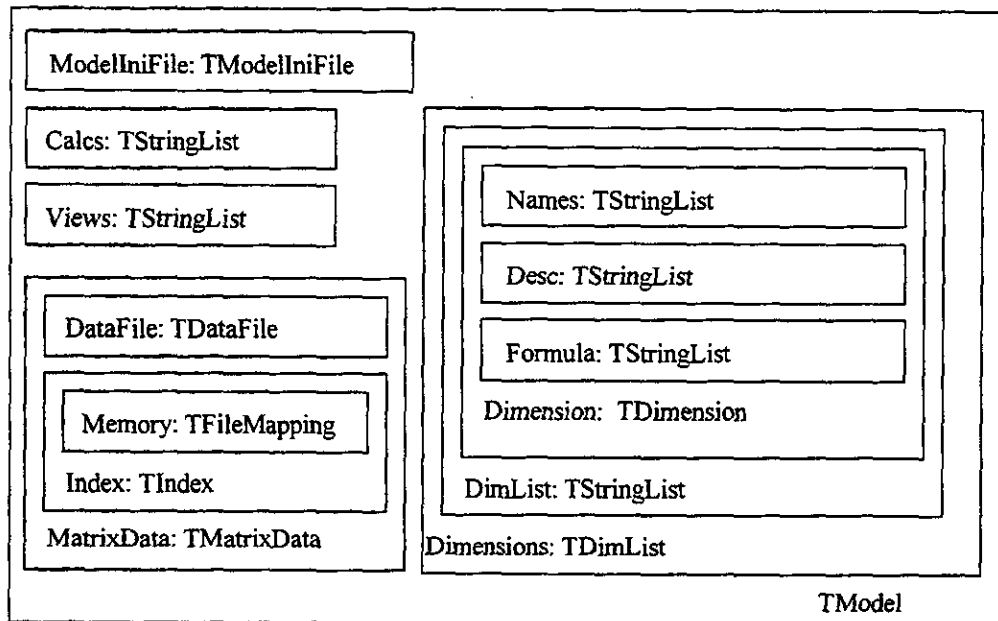


Figura 30. Estructura del objeto *TModel*

El objeto *TMatrixData* es el que se encarga de realizar todos los cálculos matemáticos dentro del modelo, así como las funciones de almacenamiento del mismo.

a) Objeto *TDynamicModViewer*

Este objeto presenta al usuario una interface para interacción con la información contenida en el modelo matemático. De esta manera, para la construcción de una aplicación basta con definir las propiedades del mismo. La estructura de este objeto se muestra en la siguiente figura:

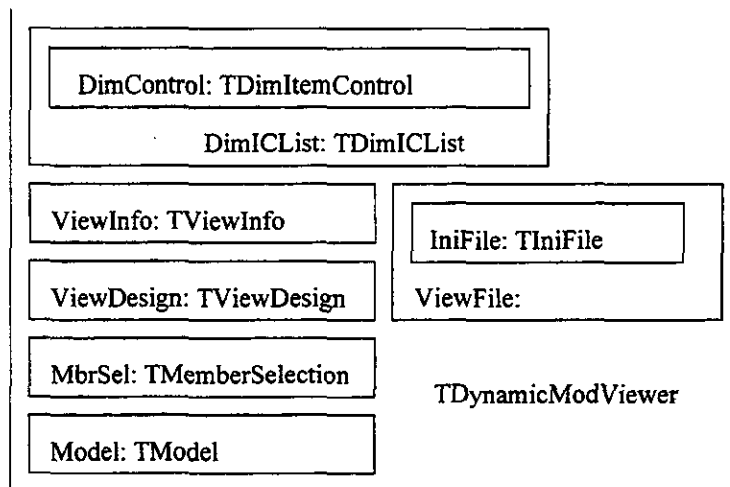


Figura 31. Estructura del objeto *TDynamicModViewer*

III. USO DEL SISTEMA

El sistema puede ser utilizado de dos formas diferentes: como una aplicación sola, para análisis multidimensional, o como una herramienta de desarrollo para aplicaciones que tengan entre sus funciones el análisis de información multidimensional. En ambas formas, el diseño y construcción del modelo multidimensional puede ser realizado con la ayuda del sistema, pues éste provee las herramientas necesarias para tal efecto. Una aplicación puede ser construída posteriormente con los modelos creados con el sistema. Este capítulo tiene por objeto el explicar el funcionamiento del sistema, de forma que pueda ser utilizado como una herramienta individual o como herramienta de desarrollo.

A. LA HERRAMIENTA DE DISEÑO

Esta es una aplicación normal compilada para Win32, por lo que se puede ejecutar tanto en Windows 95 como en Windows NT. El icono se muestra en la siguiente figura:



Figura 32. Icono del sistema

B. CREACIÓN DE UN NUEVO MODELO

La pantalla principal muestra al inicio el menú que se muestra a continuación:

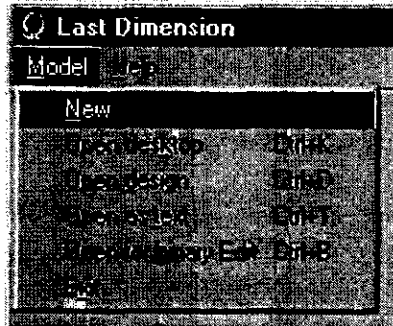


Figura 33. Menú de la pantalla principal

La primera opción se utiliza para crear un nuevo modelo multidimensional. Al seleccionarla, aparece el siguiente cuadro de diálogo:

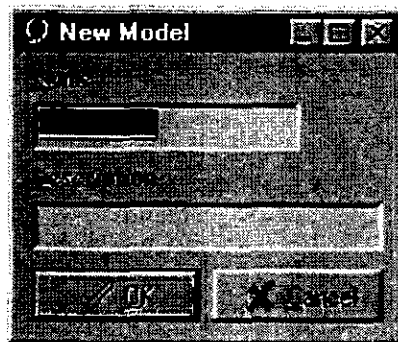


Figura 34. Cuadro de diálogo para creación de un nuevo modelo

En este cuadro de diálogo debe escribirse el nombre del nuevo modelo, que no debe pasar de 8 caracteres y su descripción, que puede ser lo que sea. Tras haber hecho esto y presionar el botón "OK", aparecerá el siguiente cuadro de diálogo:

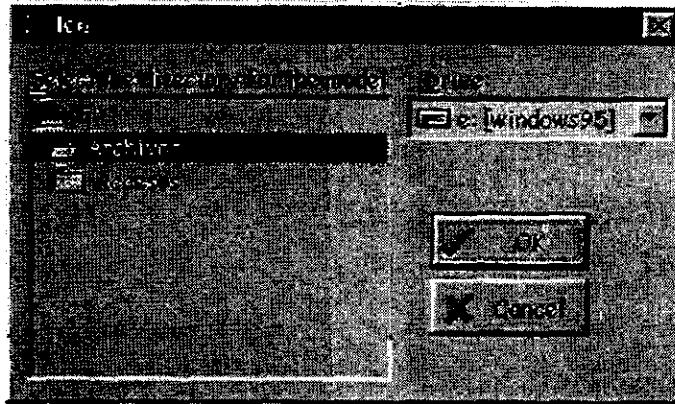


Figura 35. Cuadro de diálogo para selección del directorio del nuevo modelo

En este cuadro de diálogo se pregunta el directorio en donde deberá ser grabado el nuevo modelo. Tras seleccionarlo, y presionar el botón "OK", se creará el nuevo modelo. Nótese que esto no lo abrirá, sino que simplemente lo creará en disco duro.

1. Modificación de la estructura de un modelo

Para agregar o borrar dimensiones, definir dimensiones, y definir la densidad o esparcidad de las dimensiones se utiliza el "editor de estructura" del sistema. Para abrirlo, se selecciona la opción "Open design" del menú principal, y se selecciona el modelo que se desea abrir. Tras hacerlo, aparecerá en pantalla la siguiente ventana:

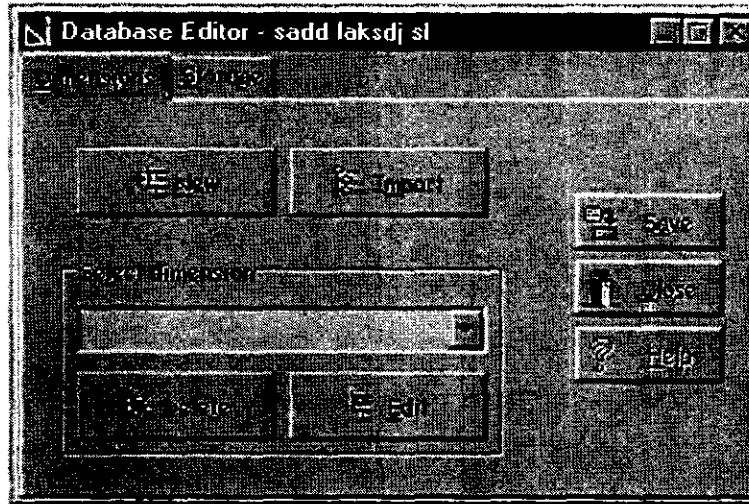


Figura 36. Ventana del editor de estructura

En esta ventana existen dos pestañas: la primera, "Dimensions" sirve para agregar, borrar y editar dimensiones. Al presionar el botón "New", o presionar el botón "Edit" con alguna dimensión seleccionada, aparecerá el editor de dimensiones, que es una pantalla como la que se muestra:

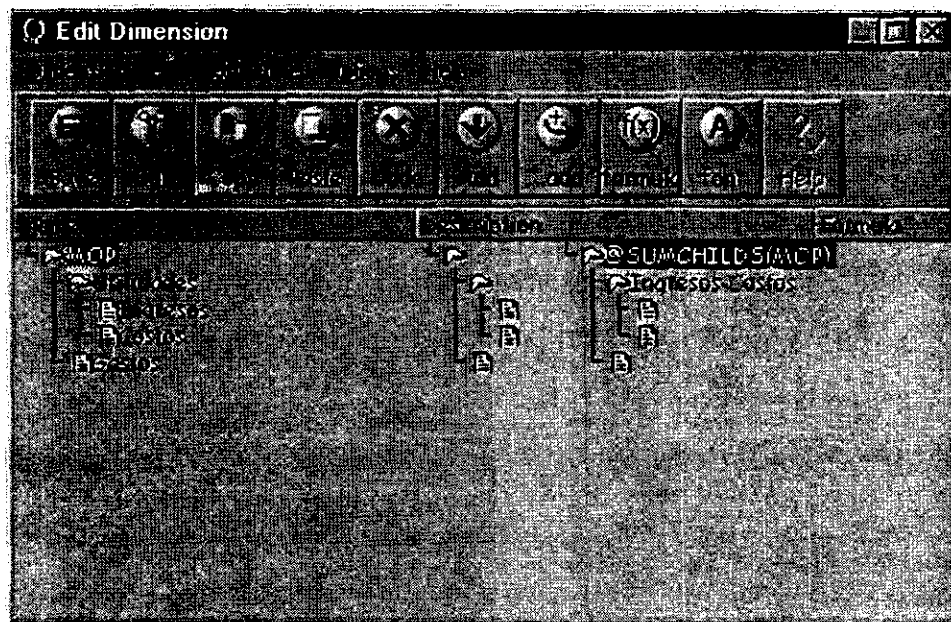


Figura 37. Pantalla de edición de dimensiones

En esta pantalla se encuentra una barra de herramientas para las operaciones más frecuentes, así como acceso a todas las funciones vía los menús. Asimismo, el botón derecho del ratón activará un menú de contexto para las mismas funciones. Al presionar el botón "Formula" o seleccionar el menú de contexto "Properties" o seleccionar el menú "Structure - edit properties", aparecerá el siguiente cuadro de diálogo:

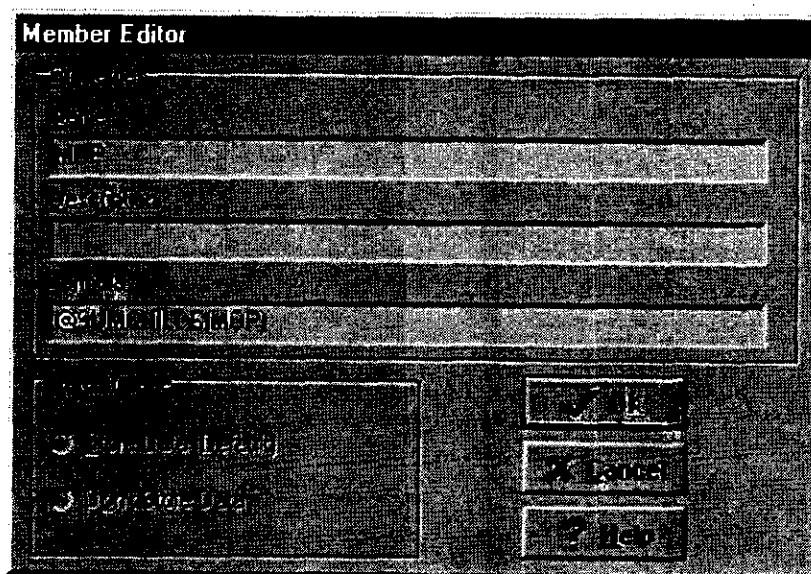


Figura 38. Cuadro de diálogo de edición de miembros

En este cuadro de diálogo se puede cambiar el nombre, descripción y fórmula de algún miembro. Igualmente se pueden cambiar estos campos presionando el botón derecho del ratón sobre algún miembro y seleccionando "edit" del menú de contexto. Las funciones disponibles para su inclusión en fórmulas se muestran en la siguiente tabla:

| <i>Función</i> | <i>Descripción</i> | <i>Cálculo</i> |
|----------------|--|----------------|
| @avg(x,y) | Promedio de x con y | $(x+y)/2$ |
| @sin | Seno de x | Sen(x) |
| @cos | Coseno de x | Cos(x) |
| @tan | Tangente de x | Tan(x) |
| @sqr | Cuadrado de x | $x*x$ |
| @sqrt | Raíz cuadrada de x | \sqrt{x} |
| @pwr | x elevado a la y | x^y |
| @ln | Logaritmo natural de x | ln(x) |
| @exp | Exponencial de x | e^x |
| @VarPer | Variación porcentual de x con respecto a y | $(x+y)/y*100$ |
| * | Multiplicación | |
| - | Resta | |
| % | Porcentaje | $x/y*100$ |
| / | División | |
| + | Suma | |
| @pi | Pi | 3.1415... |
| @sumchild | Suma de los hijos del miembro | |

Figura 39. Funciones matemáticas disponibles

C. LA PANTALLA DE EDICIÓN DEL MODELO MULTIDIMENSIONAL

Para realizar funciones que no tienen que ver con la estructura de las dimensiones, sino que más bien son operaciones con los datos que contiene el modelo, se utiliza el editor del modelo, seleccionando el menú "Open desktop" en la pantalla principal, tras lo que aparece la siguiente ventana:

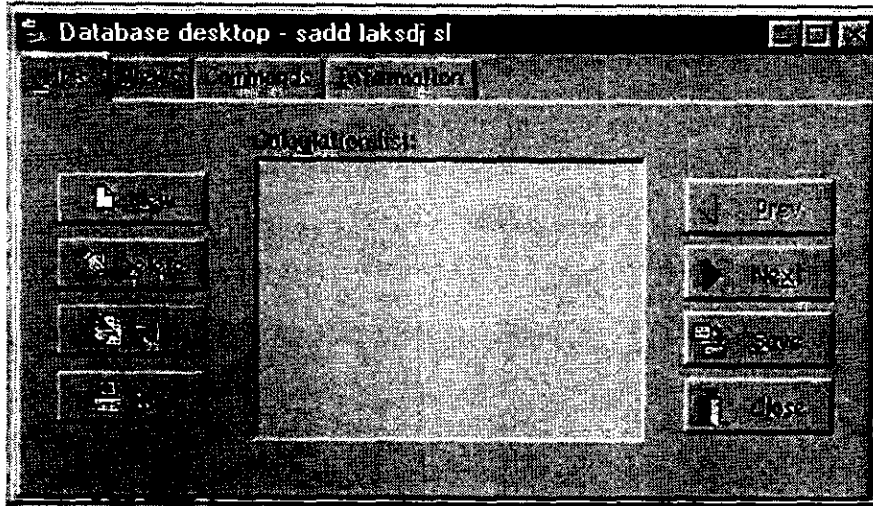


Figura 40. Ventana de edición del modelo

Esta ventana tiene cuatro pestañas, descritas en la siguiente tabla:

| Pestaña | Uso |
|-------------|---|
| Calcs | Creación, modificación, cálculo y borrado de cálculos |
| Views | Creación, modificación y borrado de vistas |
| Commands | Operaciones con la información del modelo |
| Information | Información del modelo |

Figura 41. Pestañas de la ventana de edición del modelo

Al seleccionar la edición de un cálculo existente, o crear uno nuevo, aparece la siguiente pantalla, con la que se crea un cálculo *Ad-hoc*:

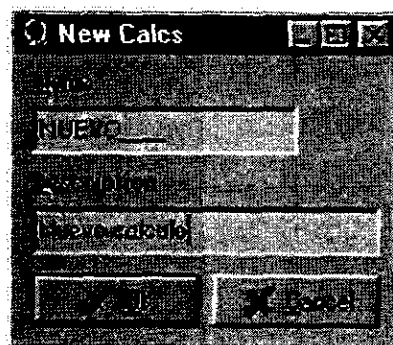


Figura 42. Ventana de creación de un nuevo modelo

Al escribir el nombre y descripciones deseadas, y presionar el botón "OK", se presenta la siguiente ventana, en donde es posible editar el cálculo:

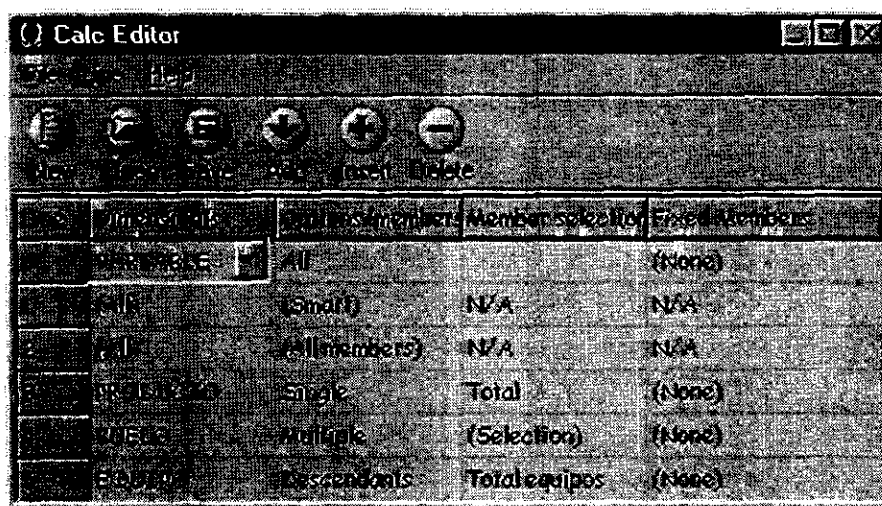


Figura 43. Ventana de edición de cálculos

En la ventana de edición de cálculos es posible eliminar o añadir líneas, las cuales se modifican presionando el botón izquierdo del ratón en alguna de sus celdas, con lo que se activarán las opciones disponibles para esa línea. Estas líneas se ejecutarán de manera secuencial, de forma que su cálculo sea síncrono.

De igual forma, es posible editar las vistas de información con la pantalla de edición de vistas. El procedimiento para crear o modificar una vista de información es el mismo que el de un cálculo, en la pestaña de vistas de la pantalla de edición del modelo multidimensional. Al seleccionar la edición de una vista de información, se muestra al usuario la siguiente pantalla

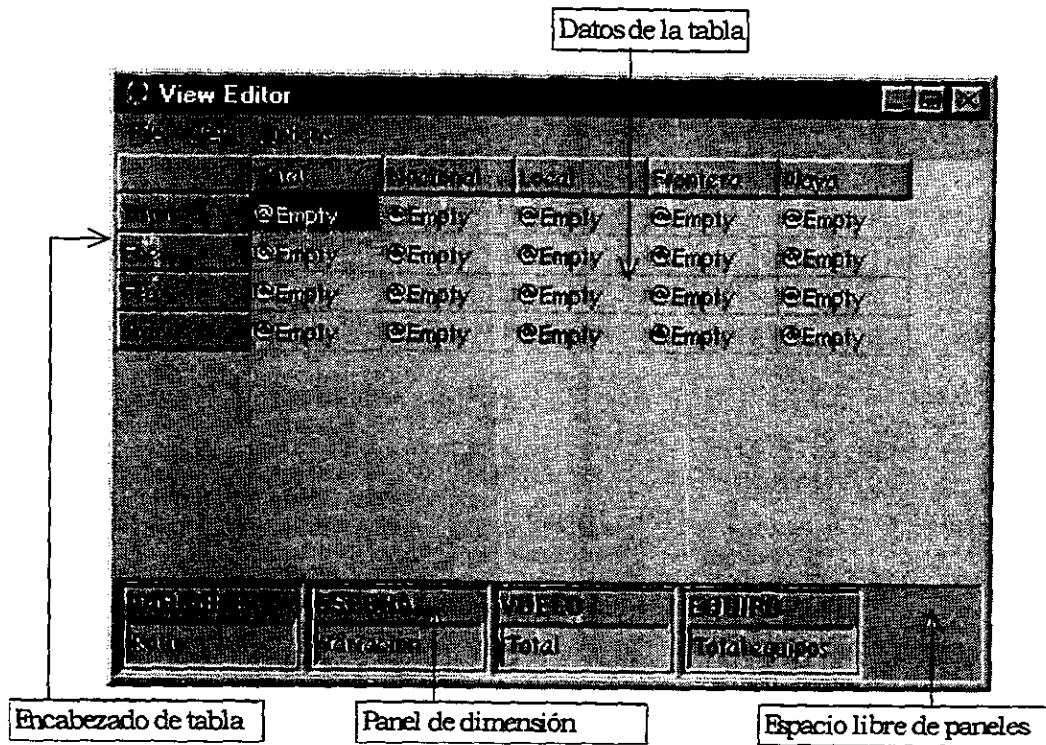


Figura 44. Ventana de edición de vistas

Las acciones que se pueden realizar dentro de esta ventana para modificar la vista se muestran en la siguiente tabla:

| Acción | Procedimiento | Descripción |
|---|---|--|
| Pivoteo (1) | "Drag and drop" de un panel de dimensión a la primera columna o al primer renglón de los datos de la tabla | Se insertará la dimensión seleccionada en la tabla |
| Pivoteo (2) | "Drag and drop" del encabezado de la tabla al espacio libre de paneles | Se removerá la dimensión seleccionada de la tabla |
| Rotación | "Drag and drop" de una columna o renglón del encabezado de la tabla a otra columna o renglón del encabezado de la tabla | Se intercambiarán las dimensiones |
| "Drill-down" | Doble click en alguna celda del encabezado de la tabla | Se expandirá el miembro seleccionado, mostrando a sus hijos |
| "Drill-up" | Doble click derecho en alguna celda del encabezado de la tabla | Se colapsará el miembro seleccionado, ocultando a sus hijos |
| Selección manual de miembros de la tabla | Click derecho en alguna celda del encabezado de la tabla, seleccionar el menú "Select members" | Se mostrará el cuadro de diálogo de selección de miembros, en donde se realiza la selección uno a uno de los miembros deseados |
| Selección manual de miembros de los paneles | Doble click en algún panel de dimensión | Se mostrará el cuadro de diálogo de selección de miembros, para la selección del miembro deseado |

Figura 45. Acciones disponibles en la ventana de edición de vistas

Al hacer la selección manual de miembros, aparece la siguiente ventana:

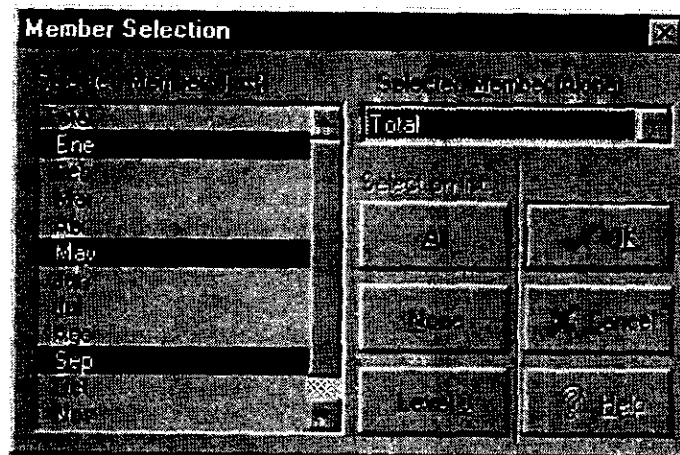


Figura 46. Ventana de selección manual de miembros

En la ventana anterior se realiza la selección de los miembros que aparecen en la dimensión en los casos en los que se muestre en la tabla o como un panel. Asimismo, es posible cambiar el formato de los números mostrados en la tabla, seleccionando el menú "Options -grid options", tras lo que aparece la siguiente ventana:

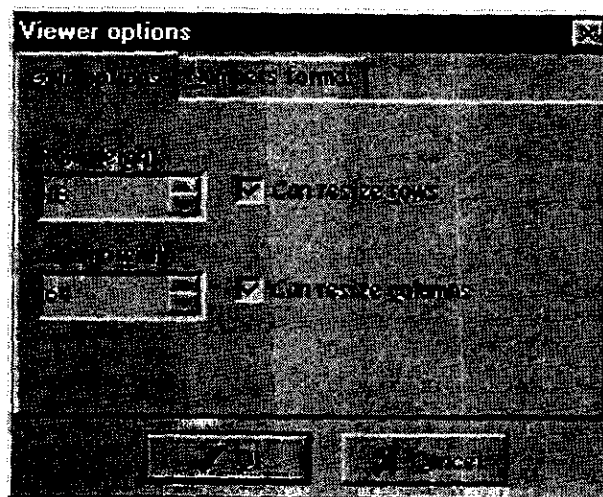


Figura 47. Edición de vistas. Ventana de opciones de la tabla

En la primera pestaña se realiza el cambio del formato de la tabla, como el ancho y la altura de los renglones y las columnas. En la segunda pestaña, se efectúa el cambio del formato de los números mostrados en la tabla, como se muestra:

**ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA**

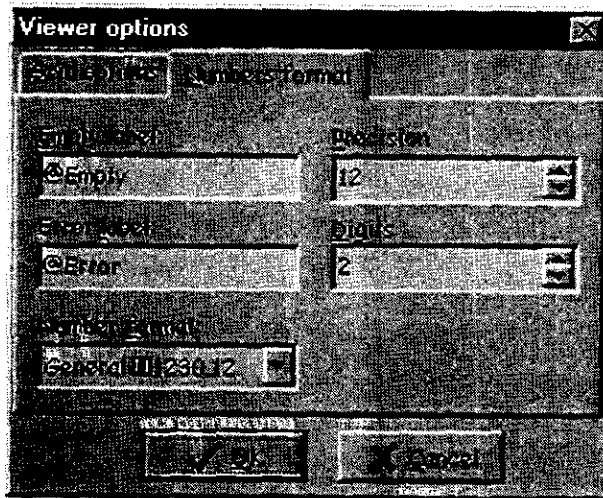


Figura 48. Edición de vistas. Ventana de opciones de formato numérico

IV. CONCLUSIONES

Con el objeto de obtener una medida del desempeño del sistema, se realizaron pruebas comparativas entre el producto final y otro producto que cumple con las especificaciones OLAP. Se eligió el sistema Essbase, producido por la compañía Arbor Software, debido a que éste es el sistema de mayor desempeño en el mercado, así como porque es el estándar a nivel mundial. Para la realización de la prueba se diseñó un modelo multidimensional típico, con las siguientes características:

| | |
|--------------------------------------|---------------------------------|
| Número de dimensiones | 6 |
| Dimensiones densas | Variables, periodos, escenarios |
| Dimensiones esparcidas | Productos, Equipos, Canales |
| Número de miembros de las variables | 7 |
| Número de miembros de los periodos | 13 |
| Número de miembros de los escenarios | 5 |
| Número de miembros de los productos | 28 |
| Número de miembros de los equipos | 21 |
| Número de miembros de los canales | 23 |
| Tamaño de bloque | 455 celdas |
| Densidad de bloque | 100% |
| Índice de esparcidad inicial | 9.52% |
| Tamaño teórico mínimo final | 17,035,200 Bytes |

Figura 49. Características del modelo de la prueba comparativa de desempeño

La prueba se realizó en una computadora *HP Vectra*, con un procesador Pentium de 100MHz y 48 MB de RAM. El sistema operativo fue Windows NT 3.51, pues Essbase no soporta la plataforma Windows 95. Se realizaron tres pruebas, que muestran las capacidades del sistema en diferentes puntos, que se describen a continuación:

- Velocidad de copiado de datos. Esta prueba mide la velocidad del copiado de información de una parte a otra de la base de datos. Ésta es una operación muy utilizada en modelos de simulación. Para realizarla, se copian datos de un miembro de cada dimensión esparcida a varios miembros más de la misma dimensión.
- Velocidad de consolidación o agregación. Con esto se mide el desempeño de las operaciones de consolidación en dimensiones esparcidas. Este tipo de funciones son utilizadas en virtualmente todos los modelos multidimensionales esparcidos. Se realiza calculando las dimensiones esparcidas.
- Velocidad de cálculo. Esto toma una medida de la capacidad de realizar cálculos matemáticos sobre una cierta cantidad de información. Este es el punto que típicamente provoca los cuellos de botella en los sistemas multidimensionales, y que es la mayor consideración de desempeño para la selección de una herramienta OLAP. Se efectúa calculando las dimensiones densas, que típicamente contienen las fórmulas complejas.

En añadidura a las pruebas anteriores, es útil conocer el espacio que requiere cada una de las herramientas para almacenar la información. Los resultados obtenidos de las anteriores pruebas son los siguientes:

Se ha logrado el objetivo de construir un sistema abierto, es decir, que otorga la capacidad de desarrollar aplicaciones que utilicen técnicas de análisis multidimensional con la ayuda de herramientas diversas. El sistema entregado incluye objetos COM que pueden ser utilizados con múltiples productos, lo que libera a los usuarios de la necesidad de recurrir a sistemas propietarios que no se ajustan a sus necesidades por completo. Gracias a ello, no es necesario para las áreas de desarrollo de sistemas la implementación de la solución completa, sino que podrán dedicar los recursos a su problemática específica.

El sistema presentado puede ser utilizado como una herramienta para el desarrollo de aplicaciones en las que se involucre el análisis de información multidimensional. Esto brinda la oportunidad de incrementar la base actual de los sistemas de información en nuevas áreas para resolver los problemas a los que se enfrentan las comunidades científica y de negocios.

V. BIBLIOGRAFÍA

(1) CODD, E.F.; CODD, S.B.; SALLEY, C.T. Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. EUA, 1982

(2) FINKELSTEIN, Richard. Understanding the Need for On-Line Analytical Servers. President, Performance Computing, Inc. EUA, 1993

(3) THOMSEN, Erik. Olap solutions: Building Multidimensional Information Systems. John Wiley & Sons. EUA, 1997