

20/17



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

---

**MI: un Sistema de Base de Datos  
Relacional Orientado al Usuario.**

**T E S I S**

**PARA OPTAR POR EL TITULO DE :  
LICENCIADO EN MATEMATICAS  
QUE PRESENTA :  
MARIAURORA MOTA BRAVO**

**MEXICO, D.F.**

**1983**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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

## Introducción.

## Capítulo 1: Marco Teórico.

- 1.1) Crisis del Software.
- 1.2) ¿Computación tipo Batch o Interactiva?
- 1.3) Características de las Computadoras Personales.
- 1.4) Bases de Datos en Sistemas Pequeños.
- 1.5) Sistema orientado al usuario  
Implica Modelo de Datos Relacional.
- 1.6) Operaciones a Base de Comandos vs.  
Operaciones a Base de Menús.

## Capítulo 2: Diseño del Sistema.

- 2.1) Características Generales.
  - 2.1.1) Resultado del Marco Teórico.
  - 2.1.2) Filosofía de Diseño.
  - 2.1.3) Internas ó de Diseño.
- 2.2) Inicio de Sesión.
- 2.3) Visión Global del Banco de Datos.
- 2.4) Operaciones con Tablas.
  - 2.4.1) Crear Tabla.
  - 2.4.2) Eliminar Tabla.
  - 2.4.3) Proyectar Tabla.
  - 2.4.4) Clasificar Tabla.
  - 2.4.5) Multiplicar Tablas.

- 2.4.6) Unir Tablas.
- 2.4.7) Diferencia de Tablas.
- 2.5) Operaciones con Renglones.
  - 2.5.1) Insertar Renglones.
  - 2.5.2) Eliminar Renglones.
  - 2.5.3) Modificar Renglones.
  - 2.5.4) Copiar Renglones.
- 2.6) Consulta.
  - 2.6.1) Consulta por Rango.
  - 2.6.2) Consulta por LLave.
  - 2.6.3) Consulta con Condición.
- 2.7) Fin de Sesión.
- 2.8) Entrada Adelantada de Respuestas para Usuarios Experimentados.
- 2.9) Combinaciones Inter-Menús.

### Capítulo 3: Diseño de la Implementación.

- 3.1) Estructuras de Datos.
  - 3.1.1) Externas.
  - 3.1.2) Internas.
- 3.2) Diagramas de Warnier-Orr.
- 3.3) Estimados de Tiempo y Espacio.

Conclusión.

Bibliografía.

## INTRODUCCIÓN.

Los sistemas de manejo de base de datos, surgieron al final de la década de los 60's como una de las respuestas más importantes al problema de la crisis del software. Previo a la introducción de estos sistemas, era necesario que los usuarios de computadoras contaran con varios programadores que dedicaban gran parte de su tiempo al desarrollo de programas de actualización y consulta a la información. Para obtener respuesta a preguntas simples tales como: ¿que terrenos hay en Copilco o Tepepan cuyo costo es menor a un millón y medio de pesos?, era necesario codificar y probar un programa completo, un proceso que puede tomar varios días. Los sistemas de base de datos, al ofrecer lenguajes de consulta a la información, han resuelto en gran parte este problema, pero han introducido otros.

Uno de los problemas existentes es que los sistemas han ido creciendo en complejidad y "sofisticación", llegando al punto en que el usuario se ve inundado por un exceso de opciones y posibilidades, y se ve en la necesidad de estudiar una pila de manuales técnicos simplemente para determinar si le conviene hacer uso del sistema.

En años recientes, el surgimiento de las computadoras personales ha hecho pensar en las necesidades de información de usuarios tales como pequeñas empresas, tiendas,

estudiantes y profesionistas. Este tipo de usuarios requieren obviamente de sistemas interactivos, simples y fáciles de usar; sistemas que guíen y ayuden al usuario en todo momento, y a la vez suficientemente poderosos para satisfacer sus necesidades.

Esta tesis presenta el diseño de un sistema de base de datos para computadoras personales denominado MI1 (Modelo Interactivo 1). El trabajo se dividió en tres capítulos. El primer capítulo (Marco Teórico) presenta los fundamentos y filosofía de diseño del sistema. Partiendo de una explicación de la crisis del software, se pasa a analizar los sistemas por lotes (batch) y los sistemas interactivos. Estos últimos son más indicados para las computadoras personales. En seguida se discuten las características de estas máquinas y las características que debe tener un sistema de base de datos orientado a este tipo de computadoras. Una descripción de los tres modelos de base de datos --jerárquico, de redes y relacional-- lleva a escoger este último debido a su mayor facilidad de uso. Por último se justifica el modo de operación a base de menús.

El segundo capítulo presenta el diseño del sistema, enfocando especialmente la atención a la interfase del usuario. Se describen las operaciones que se utilizan para manipular tablas de información, para manejo de renglones dentro de las tablas, y para efectuar consultas simples a la

información. El énfasis del diseño es la simplicidad de uso y la ayuda continua que el sistema ofrece al usuario. El sistema incluye un conjunto de funciones simples y fáciles de aprender que pueden combinarse de muchas formas. Al final del capítulo se muestra como es posible realizar operaciones relativamente complejas combinando de manera lógica conjuntos de operaciones simples.

En el tercer capítulo se describe el diseño de la implementación incluyendo las estructuras de datos y los diagramas de Warnier-Orr.

Aunque se hizo un esfuerzo para utilizar terminología en español hasta donde fuera posible, a lo largo del texto aparecen vocablos que aun no han sido aceptados oficialmente en nuestro idioma. Estos vocablos son de dos tipos: las palabras definitivamente inglesas y las palabras en español derivadas de vocablos ingleses. Ambos tipos se dejaron así, porque ya forman parte del lenguaje coloquial de los expertos en computación.

Este trabajo se desarrollo bajo la dirección del Doctor Mario Magidin Matluk.

## Capítulo 1: Marco Teórico.

En este capítulo se explica -primeramente- por qué el software ha caído en una etapa crítica. Tomando esto como un hecho indiscutible, se analizan varias metodologías y tendencias relacionadas con el desarrollo del software y se deducen algunas de las características que debe tener todo sistema que quiera prevalecer durante la próxima década.

### 1.1. Crisis del Software.

La transformación constante de la sociedad es un hecho bien establecido y analizado; lo que ahora resulta sorprendente es el rápido cambio que se registra con respecto a la técnica, y en especial a la computación.

Existen en la actualidad mas de un millón de sistemas funcionando, siete veces más de los que operaban en 1974 [BW1]. En un futuro no muy lejano, los expertos predicen que habrá una computadora en cada casa y sobre cada escritorio.

¿Que habría pensado Alexander Graham Bell, si se hubiera imaginado que 65 años después de que él inventara el teléfono, existiría uno de estos aparatos sobre cada escritorio?

La revolución de la computación, sin embargo, esta llegando a un cuello-de-botella que comienza a detener su fuerza. El

obstáculo es el software: la larga lista de comandos o instrucciones que dicen a la computadora qué debe hacer, ya que a esta se le tiene que decir con agonizante detalle, exactamente qué pasos ejecutar para convertir su poder bruto en energía orientada.

En 1973 el Air Force Command de los EUA completó un estudio, CCIP-85, que proyectaba los planes futuros de la fuerza aérea, así como las necesidades de procesamiento de información en la década de los 80's [Boehm]. Aunque muchas de las conclusiones son específicas de la fuerza aérea, hay muchos puntos que son de interés para este trabajo.

Básicamente, el estudio muestra que para casi todas las aplicaciones, el software (opuesto al hardware, displays, arquitectura, etc.) es la mayor fuente de problemas para lograr una operación eficiente.

Un hecho convincente es el factor económico. En la fuerza aérea se gastó en desarrollo de software entre US\$1000 y US\$1500 millones en 1972, aproximadamente tres veces más que el gasto en hardware. Porcentajes semejantes se encuentran por todos lados. El World Wide Military Command and Control System estimó sus gastos de hardware entre US\$50 y US\$100 millones y de software en US\$722 millones. Una estimación reciente de la NASA señala gastos de US\$100 millones para hardware y US\$200 millones para software. [Boehm] (Figura

1).

Si la diferencia de costos entre software y hardware era ya tan amplia en 1972, ¿que podemos decir hoy en día? O, aun peor, que sucederá en el futuro cuando el costo del hardware disminuya mas y mas y el costo del software aumenta continuamente? La figura 2 muestra una proyección de los gastos estimados de software para la fuerza aerea de los Estados Unidos llegando al 90!! porciento del costo total en 1985. [Boehm]

No obstante este impresionante incremento del costo del software, hace poco más de una década, la idea de vender software hubiera parecido absurda. El software venía integrado con la computadora y su costo se incluía en el precio total del sistema. Gradualmente fueron surgiendo casas de software que son compañías dedicadas exclusivamente al desarrollo y venta de software.

Dewey F. Manzer de Honeywell dice: "El progreso en el futuro estará limitado por nuestra habilidad para concebir y desarrollar software; a menos de que se encuentre un método de desarrollo de software mas barato y eficiente, será imposible para nosotros continuar expandiendo el uso de las

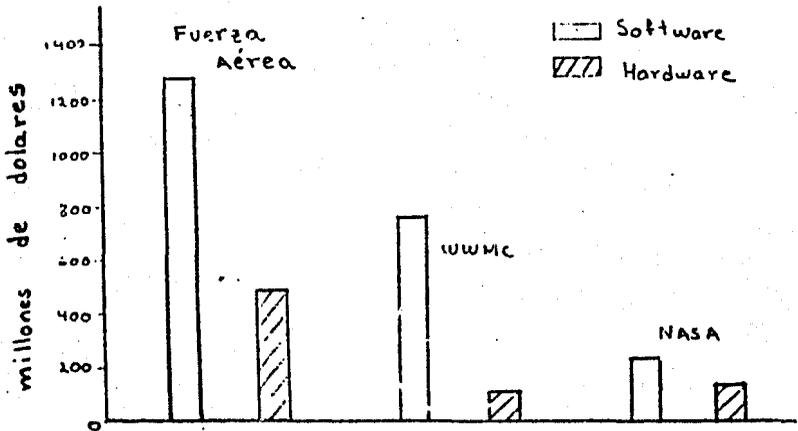


Fig.1 Comparacion de Costos entre Hardware y Software.

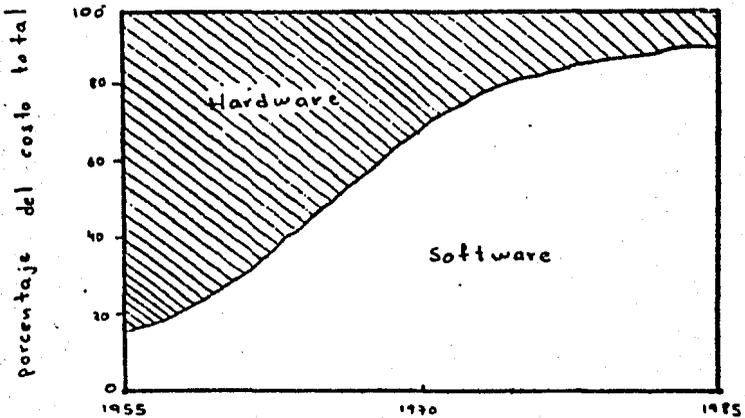


Fig.2 Tendencias de Costo en Hardware/Software.

computadoras". [BW1]

Este oscuro porvenir proviene del hecho de que escribir software es un proceso costoso, que consume tiempo y que ha demostrado resistencia inflexible a la reducción de costos que ha caracterizado al hardware. Mientras que los precios del hardware han descendido fuertemente en un factor de 1000 desde 1955, los costos de software han permanecido relativamente estables. [BW1]

Aunado a este problema esta la creciente demanda de gente experta en escribir software. La necesidad de programadores puede alcanzar 1.5 millones en 1990; más del triple del numero que labora actualmente. Muchos constructores de computadoras destinan ya el 50% de sus recursos al desarrollo de software.

"Acostumbrabamos escribir software para vender hardware, pero a la larga construiremos hardware con objeto de vender software" [BW1], dice Helbert J. Richman, vice-presidente de Data General Corp. y es quizá la frase que envuelve toda la problemática en la que esta sumergido el software.

Al diseñar un microprocesador o computadora en un chip, los fabricantes descubren a menudo que deben gastar US\$200 000 en desarrollar el software que corre en un chip de US\$10.

El desarrollo del software sigue siendo una labor intensa e

impredecible, por más intentos que se han hecho de transformarlo de arte en ciencia. "La creación de software no es como la ingeniería que conocemos; es más parecida a la creatividad del arte" [BW2].

Los esfuerzos puestos en hacer de la programación un proceso más eficiente y productivo solo han tenido un éxito limitado. Técnicas como la programación estructurada y el diseño estructurado han contribuido un poco a la solución de estos problemas; sin embargo estas técnicas apenas han cambiado el modo en que actualmente se escribe software.

Los fabricantes de computadoras están desarrollando sistemas de software completamente nuevos con objeto de facilitar su uso, de tal forma que hasta un niño sea capaz de interactuar directamente con la computadora y eliminar así la necesidad de un programador habilitado. Cada vendedor sabe que hacer los sistemas fáciles de usar será la clave de las ventas.

En las siguientes secciones se discuten algunas características importantes de los sistemas y se decide cuales de ellas deben alentarse para sacar al negocio del software de la crisis que vive actualmente.

## 1.2. ¿Computación Tipo Batch o Interactiva?

Antes de los 60's las computadoras estaban generalmente orientadas al proceso por lotes (batch). Los programas se perforaban en tarjetas, los lotes de tarjetas se introducían a la computadora y los resultados se imprimían y entregaban al usuario. Éste sistema aun se sigue usando en muchas instalaciones.

En la década de los 60's se comenzaron a desarrollar sistemas operativos capaces de soportar terminales en tiempo compartido en computadoras de gran tamaño, y así se llegó a lo que hoy conocemos como computación interactiva.

La computación interactiva ofrece nuevas posibilidades; el usuario puede ahora interactuar directamente con la computadora utilizando un teclado y una impresora ó una pantalla de rayos catódicos. Los programas y/o los datos pueden teclarse directamente en la computadora y los resultados se obtienen casi inmediatamente.

La esencia del problema es lograr una comunicación efectiva entre dos sistemas un tanto distintos. Uno, con un pensamiento creativo, correlaciona a gran velocidad y toma decisiones bajo ambigüedades, pero es lento para los cálculos, la lectura y escritura y es propenso al error. El otro, tiene poderes intelectuales limitados, pero hace cálculos, lee y escribe rápidamente y rara vez se equivoca.

Se ha preferido la computación tipo interactiva para muchas aplicaciones debido entre otras cosas a la obtención instantánea de resultados, la poderosa capacidad de edición y la flexibilidad de algunos sistemas para auxiliar al usuario. La enseñanza asistida por computadora ha llegado a ser una realidad y han surgido lenguajes especializados como PILOT que facilita el desarrollo de lecciones y exámenes.

El principal obstáculo para la popularización del modo interactivo, había sido su alto costo en comparación con el tipo batch, pero esta diferencia ha ido disminuyendo desde principios de los 70's.

Nos parece que la interacción hombre-máquina va a ser el camino para resolver muchos problemas humanos, y que de esta forma la computadora, o más específicamente los sistemas de software, llegarán a ser un medio necesario para la comunicación y ampliación de las limitaciones del hombre, tal como lo son en la actualidad el teléfono y la máquina de escribir.

### 1.3. Características de las Computadoras Personales.

Jack M. Nilles ha descrito una computadora personal (CP) como "...un sistema de computación de propósito general que contiene uno ó más microprocesadores y que es

adquirido y/o operado por un individuo ó un pequeño grupo (hasta 5) de personas. Requiere interacción a un nivel de conversación con un operador por lo menos una parte del tiempo que esta funcionando, (es decir no se usa solamente para control de procesos ó funciones similares no interactivas). Su costo es tal que el propietario puede justificar su compra en base a los beneficios recibidos: personales, financieros ó de otro tipo. El propietario de la computadora no es necesariamente un profesional en computación". [Nilles]

En la figura tres se muestra el lugar que ocupan las computadoras personales en el universo de las computadoras pequeñas. Como se puede ver, aún con las restricciones impuestas por la definición de Nilles, no se puede evitar una intersección entre las computadoras personales y algunos otros dispositivos utilizados en el área.

Esto se debe a que el concepto es muy joven. Hace apenas diez años se comenzó la producción en masa de computadoras personales, cuando en 1974 la firma MITS Inc. sacó a la venta la computadora personal ALTAIR 8800. Esta computadora empleaba el microprocesador 8080 y se vendía en US\$400 sin equipo periférico; un mínimo de dispositivos periféricos incrementaba el costo a US\$2000.

Esta computadora y otras similares que salieron al mercado

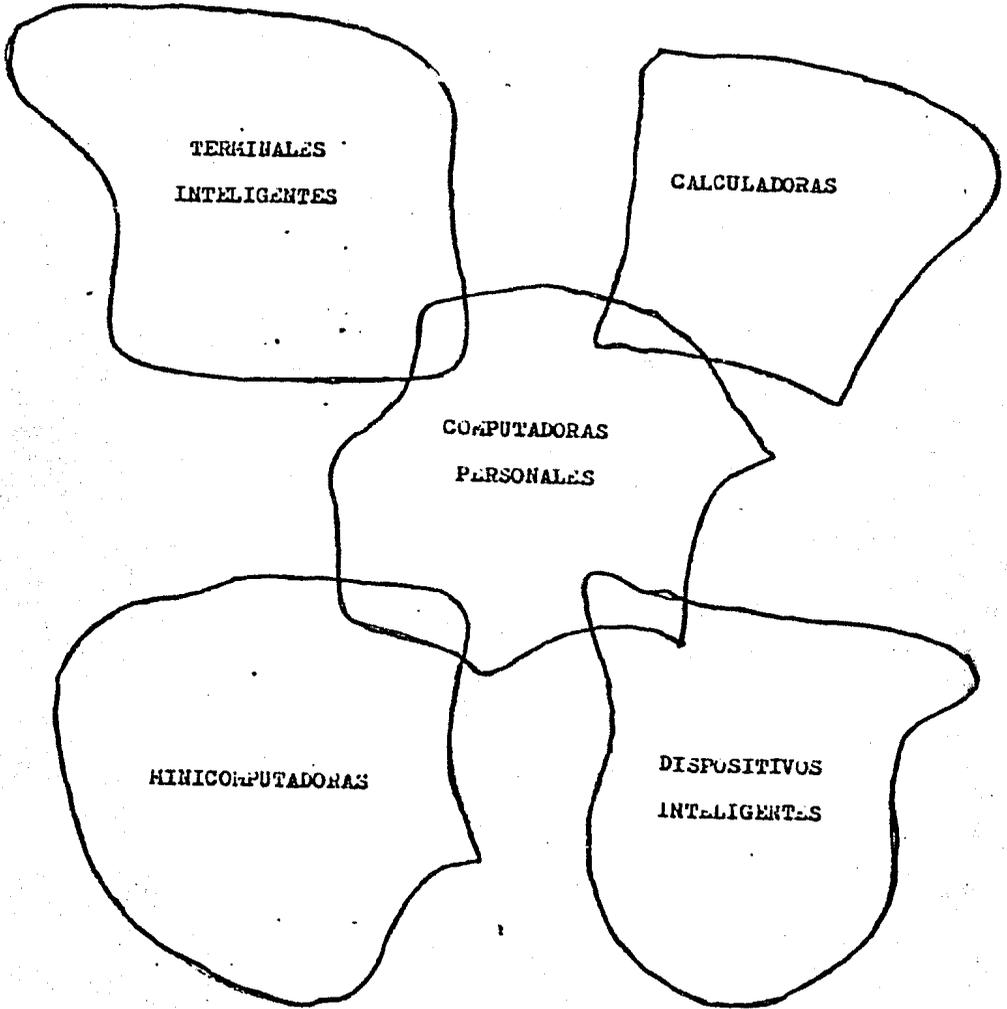


Fig.3 Las Computadoras Personales en el Universo de las Computadoras Pequeñas.

poco después, fueron pensadas principalmente para personas aficionadas a la electrónica; sin embargo, pronto se descubrió que existía un mercado más amplio, incluyendo a personas que usaban la computadora para planeación financiera, entretenimiento ó la administración de pequeños negocios.

El precio de las computadoras personales disminuyó hasta un punto en que un PC con los equipos periféricos indispensables podía adquirirse en 1979 en US\$600.

Actualmente, un sistema típico de computadora personal tiene la siguiente estructura: la CPU esta diseñada alrededor de un microprocesador ampliamente usado, tal como el 280 compatible con 8080, el Motorola 6800 ó el MOS Technology 6502, todos con palabras de 8 bits. Se necesitan por lo menos 4k bytes de memoria RAM (random access memory) para operación. La CPU y la RAM están construídas con circuitos LSI<sup>A</sup> y normalmente residen juntas, en ocasiones con el teclado. Los dispositivos periféricos son normalmente versiones, a bajo costo de dispositivos usados en las computadoras grandes. Los cassettes de cinta magnética usados en grabadoras comunes proporcionan un medio de almacenamiento secundario muy barato. Algo más caro pero con una alta capacidad de transferencia de datos es el floppy disk (disco blando). El teclado y las facilidades de impresión pueden ser combinadas en forma de una máquina de A Large Scale Integration.

escribir eléctrica. En algunos sistemas , la terminal de video es un aparato de televisión. El bus (la colección de líneas y señales que permiten a los dispositivos periféricos comunicarse con la CPU o con la memoria) que se ha establecido como estándar es el S-100, que fue usado originalmente en la computadora ALTAIR.

En cuanto al software, se puede decir que se suministra en cantidad limitada. Sin embargo, han surgido muchos proveedores independientes de paquetes especializados, en particular para aplicaciones en pequeños negocios.

BASIC es el lenguaje de alto nivel usado con mayor frecuencia debido a:

- Sencillez, que trae como consecuencia facilidad de uso
- Requerimientos modestos de memoria
- Factores históricos: se estableció como el lenguaje de programación de las primeras microcomputadoras.
- Popularidad: en la actualidad existe una gran cantidad de software escrito en BASIC para computadoras personales

En un proyecto de investigación iniciado en 1979 por la Universidad del Sur de California se ha concluido entre otras cosas que en 1990 habrá mas de 40 millones de computadoras personales funcionando. [Gray]

En este estudio, se dividió el mercado potencial en 3 tipos: la casa, los pequeños negocios y las grandes organizaciones.

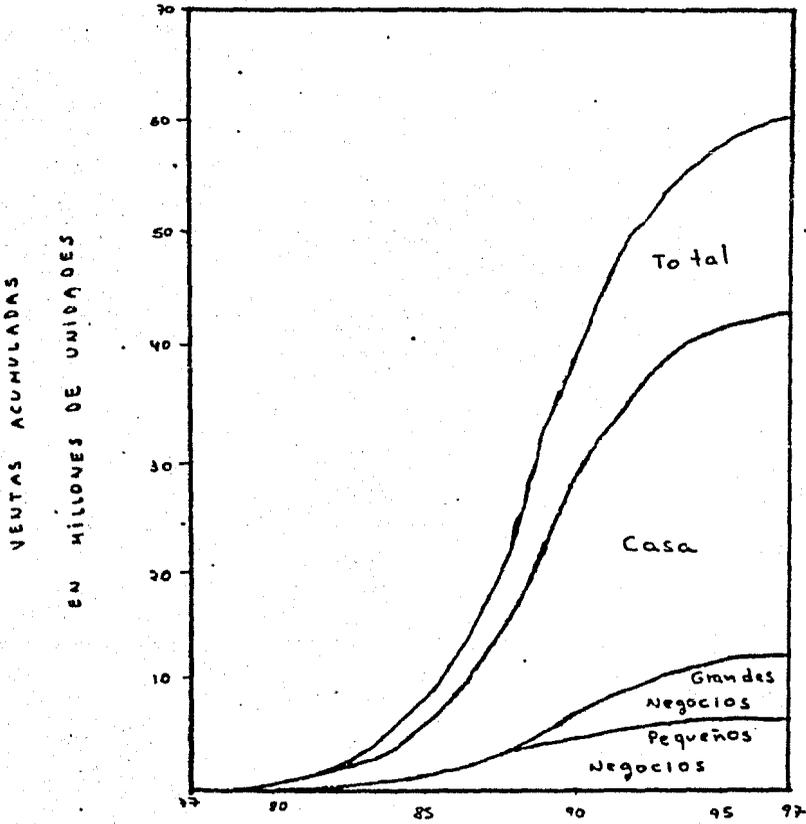


Fig.4 Pronóstico de Venta de Computadoras Personales en Tres Submercados.

En la figura 4 se observan los pronósticos de ventas de computadoras personales en tres submercados.

Las computadoras personales adquiridas para el hogar ocupan un lugar preponderante dentro del mercado potencial. Este dato es de interés, pues MII está orientado principalmente para el hogar.

Ésto nos da un lineamiento a seguir para desarrollar nuevos sistemas: deberán ser suficientemente claros y sencillos a la vista del usuario para que este se decida a usarlos sin temor; los conceptos vertidos deberán ser generales y no requerir de especialistas. La documentación deberá ser clara y concisa, auxiliada por ejemplos y evitando vocablos técnicos.

#### 1.4. Bases de Datos en Sistemas Pequeños.

Las computadoras personales, como vimos en la sección anterior, están siendo adquiridas principalmente para utilizarse en el hogar o en pequeños negocios. Las amas de casa, ingenieros, hombres de negocios, estudiantes, etc. tienen en la computadora personal una herramienta muy útil.

Las necesidades de información del hombre moderno son múltiples. Un jefe sindical requiere un registro de obreros afiliados; un astrónomo tabula día a día sus observaciones; un profesor registra las calificaciones de sus alumnos; pe-

queñas empresas mantienen registros de cuentas por pagar y cobrar, etc.

Para que esta información pueda consultarse, debe almacenarse en forma ordenada y actualizada; así encontramos en el mercado los directorios telefónicos, las agendas, los libros de ingresos y egresos y toda suerte de documentos que pueden contener información.

El problema, sin embargo, no ha sido totalmente resuelto: la información no está en un lugar central sino que se encuentra por todos lados, algunos datos son difíciles de actualizar y no siempre resulta sencillo o eficiente localizar la información deseada.

Un sistema de base de datos para una computadora personal viene a resolver todos estos problemas al proporcionar una forma sencilla y rápida de almacenar y actualizar información y un medio para acceder de manera eficiente los datos requeridos.

Formalmente, se define un sistema manejador de base de datos (DBMS, Data Base Management System) como un sistema de software o conjunto de programas que permitan a una o varias

personas interactuar con un conjunto (ó banco) de datos, como se puede ver en la figura 5.

No obstante la definición formal, el principal objetivo de un DBMS es permitir al usuario interactuar con los datos en términos abstractos, más que como datos almacenados en una computadora. En este sentido, el DBMS actúa como un filtro, permitiendo al usuario especificar qué es lo que se debe hacer, con poca ó nula atención en los algoritmos de almacenamiento ó en la representación interna de los datos.

Los DBMS para computadoras grandes han sido dados a conocer y adquiridos desde los años 70's. Entre los paquetes más conocidos tenemos TOTAL, System 2000, DMS170, IMS, Adabas, etc. Muchos centros de cómputo en México y en el extranjero cuentan ya con paquetes de base de datos.

Los DBMS para computadoras personales, están en una etapa de pleno desarrollo. En Abril de 1981, la revista BYTE, dedicada a computadoras personales, anunció cuatro paquetes de base de datos listos para instalarse. Otra prueba es el hecho de que en Octubre de 1981, ACM llevó a cabo una reunión de trabajo titulada "Bases de Datos en Sistemas Pequeños" en donde se expusieron las características de los sistemas existentes, se presentaron nuevos y se dieron lineamientos para desarrollos futuros.

Uno de los lineamientos fue la sencillez y facilidad de uso:

un ama de casa, un hombre de negocios ó un estudiante no tienen interés, ni tiempo para aprender el funcionamiento de un sistema complejo. La sencillez y facilidad de uso sólo se logrará si se considera el punto de vista del usuario; esto es, se toman decisiones de diseño bajo el criterio "¿Que es mejor para el usuario?" y no bajo el argumento "elige siempre la opción que resulte más fácil de implementar ó la más eficiente". Esto es un concepto muy importante por lo que le dedicaremos la siguiente sección.

### 1.5. Sistema Orientado al Usuario

#### Implica Modelo de Datos Relacional.

Uno de los problemas más inquietantes en la ingeniería de software consiste en crear sistemas que puedan ser usados por gente común ó personal conocedor de un campo profesional particular pero que están poco enterados de la operación de una computadora.

En los sistemas de software más sofisticados, la simplicidad de uso y el punto de vista del usuario deberían aparecer al principio de la lista de prioridades de diseño. Si ésto no sucede, el resultado será que muchos sistemas altamente competentes no tendrán éxito ni popularidad; muchos usuarios potenciales se resistirán a usarlos debido a la experiencia que se requiere para operar un sistema complejo en forma eficiente. Otros usuarios abandonarán los sistemas porque no sabrán como continuar ó como recuperarse de un mensaje de

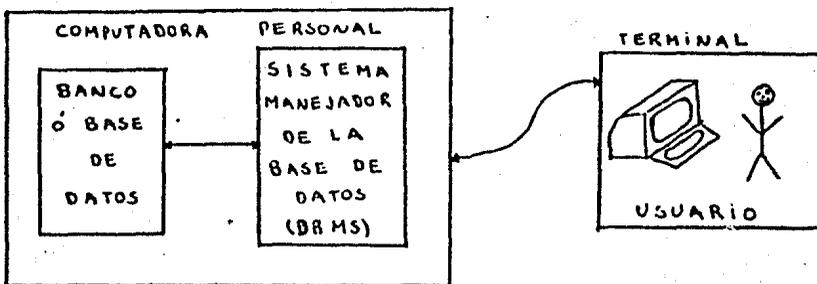


Fig. 5 Vista Gráfica de un Sistema Manejador de Base de Datos.

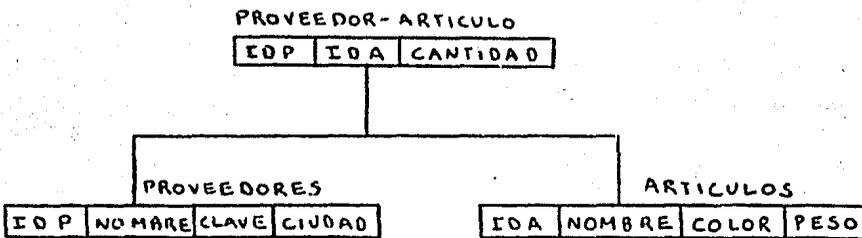


Fig.6 Modelo Jerárquico de Base de Datos.

error.

Un ejemplo sorprendente de esta deficiencia puede ser extraído de un sistema de recuperación de información legal altamente sofisticado desarrollado en los EU [Slon]. El sistema fue aclamado técnicamente desde su diseño y potencialmente era de gran valor para los abogados. La organización patrocinadora, sin embargo, no fue capaz de convencer a muchos abogados a adoptar el sistema como una herramienta profesional. Las mejoras graduales en el funcionamiento interno del sistema no modificaron este hecho. Un nuevo administrador reconoció, afortunadamente, que la razón de la impopularidad del sistema no era el sistema en sí, sino la dificultad que se le presentaba al usuario cuando estaba aprendiendo a operarlo. La nueva administración inició el diseño de una terminal funcional, muy adaptada a las necesidades de los abogados. En meses, el número de usuarios se incrementó en cientos. Ejemplos como éste se ven por todas partes y seguramente todo usuario de computadoras se ha enfrentado a un sistema que requiere de aprendizaje y tiempo para poder hacer uso eficiente de él.

En el caso de un sistema de base de datos, el problema continúa: cómo lograr que el sistema sea sencillo de operar, de tal forma que un gran número de personas pueda hacer uso eficiente de él?

Una característica que hace a una base de datos fácil o difícil de usar, es la forma en que el usuario "ve" la información almacenada en el banco de datos. En la actualidad existen tres modos de visualizar la información: en forma jerárquica, como una red ó vista como tablas.

A continuación se mencionan las características de estos tres modelos, y las ventajas y desventajas que tiene cada uno de ellos desde el punto de vista del usuario.

**Modelo jerárquico:** por razones históricas, el modelo jerárquico es el mas popular. Esta compuesto por una serie de registros ó nodos y ligas que los unen formando un árbol. La figura 6 muestra un ejemplo.

Como se puede ver en el dibujo, cada nodo esta ligado a un registro padre y ademas no existen ligas entre nodos hermanos.

Por la forma en que se encuentran las ligas entre los registros, insertar y borrar registros es una tarea que requiere de mucho cuidado. Cuando se quiere insertar un nuevo registro, debe encontrarse primero un nodo padre del cual pueda depender. Al borrar un registro, se debe tener especial cuidado en no eliminar a los nodos dependientes, sino pasarlos a depender de otro registro.

La recuperación de información en este modelo se hace

exclusivamente en forma vertical. La consulta entre nodos hermanos es muy difícil ya que no existe forma directa de llegar de uno a otro.

A pesar de todas estas limitaciones, dos de los sistemas de base de datos más conocidos: IMS (Information Management System) y System 2000 han sido desarrollados sobre el modelo jerárquico.

Una ventaja de este modelo es que dentro de la administración, parece muy natural, pues la estructura jerárquica es análoga a la estructura administrativa. En algunas otras aplicaciones sin embargo, resulta difícil adaptarse a este modelo y solo se logra duplicando la información.

Analizando este modelo desde el punto de vista del usuario común, se puede deducir que no es conveniente ya que la estructura de la información debiera diseñarse cuidadosamente teniendo previo conocimiento de las consultas que se harán.

Modelo de redes: este modelo es una extensión del modelo jerárquico, si se remueven las limitaciones que existen para ligar registros. En un modelo de redes no hay restricciones en las relaciones representadas por las ligas. Un ejemplo se muestra en la figura 7.

La información almacenada se puede acceder por medio de un lenguaje de datos que permita insertar, modificar, borrar y

PROVEEDOR

IDP	NOMBRE	CLAVE	CIUDAD
P1	Celis	10	Puebla

P2	Cano	15	Torreón
----	------	----	---------

P3	Santos	35	Oaxaca
----	--------	----	--------

P4	Zavala	12	Tuxtla
----	--------	----	--------

ARTICULOS

IDA	NOMBRE	COLOR	PESO
A1	Tornillo	Rojo	12

A2	Tuerca	Verde	17
----	--------	-------	----

A3	Perno	Negro	14
----	-------	-------	----

A4	Rondana	Azul	15
----	---------	------	----



Fig.7 Modelo Reticular de Base de Datos.

ARTICULOS

IDA	NOMBRE	COLOR	PESO
A1	Tornillo	Rojo	12
A2	Tuerca	Verde	17
A3	Perno	Negro	14
A4	Rondana	Azul	15

PROVEEDOR

IDP	NOMBRE	CLAVE	CIUDAD
P1	Celis	10	Puebla
P2	Cano	15	Torreón
P3	Santos	35	Oaxaca
P4	Zavala	12	Tuxtla

Fig.8 Modelo Relacional de Base de Datos.

recuperar registros y crear o destruir las conexiones entre registros. En el modelo de redes, a diferencia del jerárquico, un registro puede unirse a otro registro por medio de muchas ligas. Así, cuando un nuevo registro se inserta, puede conectarse a muchos registros ya existentes. Cuando se borra un registro, no se borran necesariamente todos los registros conectados a él.

El lenguaje de datos reticular debe contener dos facilidades básicas de selección de datos: localizar directamente algunos registros, y, localizar registros basándose en las conexiones entre ellos.

Desde el punto de vista del usuario, este modelo resulta complicado ya que para poder tener acceso a los datos, el usuario debe saber cómo está almacenada la información: qué tipo de dato contiene cada registro y qué representan cada una de las ligas que unen a cada par de registros. Para insertar un registro, se deben crear cada una de las ligas que lo unen a otros registros. Todas estas operaciones implican que el usuario tenga mucho conocimiento de la estructura interna de la información y por tanto es poco conveniente para un usuario promedio.

Modelo relacional o la información vista como tablas: éste modelo fue el último en aparecer. Introducido en 1970 por Codd, utiliza tablas para visualizar la información. Una

tabla esta compuesta por un conjunto de columnas como se puede ver en la figura 8.

Cada tabla tiene una llave: una columna cuyos valores identifican plenamente a cada renglón. En el ejemplo, la columna-llave de la tabla ARTICULOS es IDA. Gracias a la columna-llave es posible detectar información duplicada.

La forma tabular de ver la información no requiere de ligas como los otros dos modelos y se basa en estructuras bien conocidas por todos: tabla y llave.

Un sistema de base de datos estructurado sobre el modelo relacional, debe contar con un lenguaje de datos relacional que le permita llevar a cabo operaciones sobre tablas: crear, destruir y modificar, incluir o eliminar registros, algunas funciones para realizar consulta y otras operaciones mas elaboradas como proyección y multiplicación de tablas.

Desde el punto de vista del usuario, el modelo relacional es el mas natural y sencillo. La gente esta acostumbrada a manipular y leer información tabular en su vida diaria -desde una simple lista de nombres y teléfonos hasta tablas de estados financieros y contables. No es necesario aprender conceptos ajenos como "liga", "red", "arbol", etc.

La consulta al banco de datos, se hace tambien de manera muy natural pues el usuario tiene una idea clara del tipo de

información almacenada y de las relaciones existentes entre las tablas.

Por todo lo anterior, considero que el modelo de datos relacional es el mas adecuado para un sistema de base de datos orientado al usuario.

### 1.6. Operaciones a Base de Comandos vs. Operaciones a Base de Menús.

Se han expuesto en las secciones precedentes algunas características que deben tener los sistemas computacionales para combatir la crisis que vive el software actualmente: la computación interactiva es superior en muchas aplicaciones, a la computación tipo batch; desarrollar sistemas para computadoras personales ha adquirido sentido a partir de 1975 cuando se extendió y popularizó el uso de estos equipos; una base de datos es una herramienta indispensable dentro de una computadora personal para el almacenamiento y recuperación de información en forma sencilla y rápida; se ha preferido el modelo de datos relacional por considerarlo mas adecuado desde el punto de vista del usuario.

Por último, se discuten las características de los sistemas interactivos a base de comandos y la operación a base de menús, se presentan ventajas y desventajas de estos dos modos de operación y se opta por uno como base de nuestro sistema.

**Operación a base de comandos:** en este modo, el usuario introduce en el teclado el comando que desea ejecutar, por ejemplo L para listar, C para compilar, E para ejecutar, etc. El mnemónico por lo general va acompañado de una serie

de parámetros que deben introducirse en un cierto orden. El usuario sabe de antemano en qué etapa se encuentra, cuáles funciones se pueden llevar a cabo desde esa etapa, los mnemónicos de las funciones y la sintaxis que acompaña a cada función. Si el usuario olvida o no conoce de esta información, no puede operar con el sistema.

Este tipo de operación en realidad implica que el usuario aprenda un lenguaje de comandos nuevo para él, lo que es particularmente difícil para usuarios no familiarizados con computadoras. Antes de iniciar una sesión con un sistema que opera a base de comandos, el usuario debe invertir una cantidad considerable de tiempo en estudiar los manuales, con objeto de conocer los comandos y las reglas de sintaxis que el sistema exige. Esto se convierte en una tarea aún más difícil cuando los manuales son confusos y voluminosos. Sin embargo, si el usuario ha aprendido de memoria todos los manuales, este modo de operaciones es conveniente pues el usuario realizará su trabajo en forma rápida y concisa.

Operación a base de menús: en este modo de operación, el usuario selecciona, de entre una lista, la función que desea ejecutar. La lista es proporcionada por el sistema, que checa también que la respuesta del usuario sea una de las opciones posibles.

La opción elegida por el usuario puede desencadenar una

serie de acciones y preguntas específicas para llevar a cabo una función ó desplegar en la pantalla otro menú.

Las ventajas que tiene este tipo de operación es que el usuario no necesita saber ni recordar las funciones que puede llevar a cabo en cada etapa, ni conocer los mnemónicos de las funciones ni su sintaxis; el sistema le proporciona la información requerida en todo momento. Un ejemplo de menú se muestra a continuación:

#### MENÚ DE OPERACIONES

L LISTA LA INFORMACIÓN QUE TIENES ALMACENADA  
C COMPILA EL PROGRAMA  
E EJECUTA EL PROGRAMA OBJETO  
M MENÚ PARA ACCESAR LA BASE DE DATOS

Una posible desventaja de este tipo de operación es que el tiempo necesario para especificar una función es un poco mayor que en el anterior, ya que para cada instrucción el sistema debió haber desplegado un menú. Si el usuario tiene ya cierta experiencia con el sistema, y conoce las opciones de todos los menús, entonces el despliegue de menús será una operación tediosa que consume tiempo innecesario. Como se verá mas adelante, esta desventaja puede eliminarse fácilmente si se permite al usuario hacer entrada adelantada de opciones.

Con la operación a base de menús, el sistema proporciona al usuario toda la información que éste necesita. De este modo el uso del sistema no está restringido a un grupo de

personas que han invertido mucho tiempo estudiando los manuales.

De acuerdo a lo expuesto anteriormente, consideramos que la operación a base de menús es preferible sobre la operación a base de comandos desde el punto de vista del usuario.

## Capítulo 2: Diseño del Sistema.

En el diccionario de Webster se define "diseño" como un esquema preliminar que muestra las principales características de algo que va a ser ejecutado. En términos de informática, y en especial en este trabajo, el diseño incluye: decidir qué funciones se van a llevar a cabo, agrupar las funciones en módulos -basado en algún criterio- y especificar los detalles con respecto a lo que entra al sistema (ENTRADA), las manipulaciones y operaciones que se efectúan (PROCESO) y los resultados generados por el sistema (SALIDA).

### 2.1) Características Generales.

Las características del sistema pueden considerarse de tres tipos: aquellas que se deducen del marco teórico y que son en cierta medida impuestas al sistema desde fuera; son características que se le piden al sistema mucho antes de empezar el diseño detallado; la filosofía general de diseño, es decir el supuesto bajo el que se desarrolló el sistema y las características internas que fueron surgiendo al desarrollar en forma mas precisa el diseño. En las siguientes secciones, se enumeran los elementos de cada uno de estos grupos.

### 2.1.1) Resultado del Marco Teórico.

A continuación se detallan las principales características externas (ó surgidas como resultado del marco teórico) de MI1.

- MI1 es un sistema, que se ejecutará en forma interactiva en una computadora personal.

- MI1 es un sistema de base de datos que permitirá almacenar y recuperar información.

- MI1 es un sistema totalmente orientado al usuario, es decir, el usuario no tendrá que preocuparse del funcionamiento interno del sistema pues esta "muy alejado" de la máquina. La información estará almacenada por medio de tablas, usando el modelo relacional de base de datos.

- El usuario será conducido por MI1 en todo momento. El sistema despliega menús y hace preguntas concisas para obtener los valores ó instrucciones para cada función. Las opciones a los menús se identifican con mnemónicos.

- Los usuarios experimentados podrán adelantar respuestas haciendo mas rápida la operación.

- Las líneas desplegadas por el sistema estarán precedidas por un signo de "mas" (+) con objeto de que no surja confusión entre los renglones del sistema y los del usuario.

A continuación se muestra una tabla comparativa entre las características de algunos sistemas de base de datos y MI1.

## Otros sistemas

## MI1

A menudo se requiere fuerte motivación para usarlos ya que los manuales son voluminosos, en ocasiones incompletos o confusos.

El usuario sólo deberá teclear MI1 y a partir de allí, se le darán instrucciones para creación, actualización y consulta del banco de datos.

Como casi todos los sistemas han sido diseñados e implementados en EU, los manuales están generalmente en inglés, limitando su uso en México.

Es un sistema diseñado en México, con texto y manual en español.

El usuario debe tener algún conocimiento en computación; además, es necesario leer manuales técnicos.

No requiere que el usuario tenga conocimiento en computación; sólo debe entender los conceptos de tabla y llave.

Manejan un lenguaje de descripción de datos que el usuario tendrá que aprender. Para llevar a cabo la consulta se usa el lenguaje de consulta.

Fig.9 Tabla Comparativa entre MI1 y Otros Sistemas:

### 2.1.2) Filosofía de Diseño.

El sistema se diseñó bajo la siguiente hipótesis:

MI1 es un sistema de base de datos relacional que servirá única y exclusivamente para almacenar y recuperar información en forma interactiva por un solo usuario.

Se evitó de esta forma diseñar un sistema que pretenda tener muchos usos y por lo mismo resulte tan barroco y complicado que sea difícil de usar y aprender. Entre los lenguajes de programación, hay algunos ejemplos de esta situación: PL/1

mezcla dentro de sus elementos, características de lenguajes de tipo administrativo (como COBOL), con facilidades de lenguajes de tipo científico, como pueden ser FORTRAN y ALGOL y otras inspiradas en lenguajes ensambladores. Esto dio como resultado un lenguaje de computación muy sofisticado y completo en el que "todo se puede hacer", pero en el cual es difícil hacerlo. Como contraste citaremos PASCAL, que cuenta con un conjunto básico de características sencillas y elegantes que pueden combinarse en varias formas.

Tomando en cuenta esta hipótesis y, para evitar caer en una sofisticación excesiva y redundante, se decidió que en MI1 habrá solamente un conjunto sencillo pero poderoso de operaciones básicas. Para llevar a cabo funciones más complicadas, será necesario combinar algunas operaciones incluidas en el conjunto básico. Al final del capítulo se dan varios ejemplos de estas combinaciones.

### 2.1.3) Internas ó de Diseño.

Se detallan a continuación las características propias del diseño de MI1.

Todos los menús se desplegarán con un formato semejante en el espacio destinado a este fin, en la parte superior de la pantalla. Como encabezado de despliegue aparecerá el título

del menú. Cada opción aparece precedida por un mnemónico de una o dos letras. La respuesta a la pregunta formulada por el sistema se puede hacer en un período de tiempo corto, consideración muy importante para un sistema interactivo donde la velocidad de respuesta es un factor determinante; se tiene la ligera desventaja de que una respuesta cuyo mnemónico es X para el menú n, no cause el mismo efecto que la respuesta X en el menú m. Éste detalle guarda una implicación de fondo. Los mnemónicos no tienen un significado absoluto; su función depende de la posición en que se encuentren; en este sentido se dice que tienen un valor posicional.

La última opción de cada menú es una función para salir de ese nivel y regresar al menú inmediato superior; en el caso del menú principal, la última opción termina la sesión.

Existen dentro del sistema cinco tipos de respuesta.

1) Opción al menú: el usuario puede responder con cualquiera de las letras que le ofrece el menú. Por ejemplo:

```
+          MENÚ DE VISION GLOBAL
+
+  N  NOMBRE DE TABLAS          G  GENERALIDADES DE U/TABLA
+  F  FIN DE VISION GLOBAL
```

El usuario tiene la opción de elegir las letras N, G ó F. Si la respuesta no es una de las posibles opciones, se despliega el siguiente mensaje:

```
+  ERROR: X , NO ES UNA DE LAS POSIBLES OPCIONES
```

Esto se repite hasta que la opción seleccionada por el usuario es igual a una de las opciones aceptadas por el sistema.

En todas las opciones de menú que se refieren a mostrar alguna información, los resultados se despliegan por default en la pantalla; es posible imprimir los resultados agregando la letra L (listar) a cada una de las opciones del menú.  
Ejemplo:

```
+ N(L) NOMBRE DE TABLAS
```

Con la opción N, se despliegan en pantalla los nombres de las tablas almacenadas en disco; introduciendo NL en la consola, las tablas se listan en impresora.

2) Una lista o pareja de números que indican un rango: con este tipo de respuesta el usuario indica al sistema los objetos que quiere analizar a partir de una lista ordenada. El rango de validez de la respuesta se despliega en el momento de formular la pregunta, por ejemplo:

```
+ RANGO DE RENGLONES? (MIN 1, MAX 13)
```

1-10 indicando renglones numerados del uno al diez o  
5,7 refiriéndose al cinco y siete

Si la respuesta del usuario no está comprendida en el rango indicado, se despliega un mensaje de error:

```
+ ERROR: 1-15 , NO ES UN RANGO ACEPTABLE
```

```
+ RANGO DE RENGLONES? (MIN 1, MAX 13)
```

3) **Un nombre seleccionado de una lista:** el usuario debe escoger un nombre; este nombre sin embargo, debe formar parte de una lista, por ejemplo: el nombre de una tabla que se encuentra almacenada en disco. El sistema verifica esta respuesta comparando el nombre dado por el usuario contra cada uno de los nombres incluidos en la lista. Si el nombre no casa, se envía un mensaje de error. Por ejemplo:

```
+      NOMBRE DE LA TABLA?
NAGICIO
+      ERROR: NAGICIO , NO ES EL NOMBRE DE UNA TABLA
+      NOMBRE DE LA TABLA?
```

Este ciclo se repite hasta tres veces si el usuario no ha dado una respuesta correcta. Después del tercer fracaso, se descarga la opción. Esto se hace con objeto de evitar que la sesión caiga en un "loop" indeseable cuando el usuario no recuerda la información solicitada.

4) **Un nombre arbitrario:** se refiere al nombre que el usuario escoge para llamar a una nueva tabla ó a una nueva columna. El sistema verifica que el nombre no sea duplicado y que cumpla con las reglas impuestas a su tipo; en caso contrario envía un mensaje de error. Ejemplo:

```
+      NOMBRE DE LA TABLA?
TELEFONO
+      ERROR: TELEFONO , ES UN NOMBRE DE TABLA DUPLICADO
```

5) **Afirmación ó Negación:** el usuario tiene posibilidad de contestar SI ó NO a preguntas formuladas por el sistema. El

sistema verifica y manda un mensaje de error en caso necesario. Ejemplo:

```
+ MODIFICACIONES? (S , V)
POR
+ ERROR: POR , NO ES UNA RESPUESTA VALIDA
+ MODIFICACIONES? (S , N)
SI
```

Se ha hecho un esfuerzo para que la operación de las funciones sea tan uniforme y consistente como sea posible. Esto comprende lo siguiente:

- Uniformidad en los despliegues.
- Claridad y uniformidad en los mensajes de error.
- Consistencia en la agrupación de las funciones.

En la descripción del sistema usaremos "listar" indicando escribir en la impresora y "desplegar" para mostrar en la pantalla.

## 2.2) Inicio de Sesión.

Las operaciones que el sistema es capaz de efectuar se han agrupado en cinco categorías: visión global, operaciones con tablas, operaciones con renglones, consulta y fin de sesión.

Cada uno de estos grupos corresponde a una opción del menú principal como se observa en la figura 10.

El menú principal es el siguiente:

```

+           MENÚ PRINCIPAL
+   V  VISION GLOBAL       T  TABLAS: OPER.       C  CONSULTA
+   R  RENGLONES: OPER.   F  FIN DE SESION
  
```

## 2.3) Visión Global del Banco de Datos.

Con el conjunto de operaciones agrupadas bajo el nombre V VISION GLOBAL se puede dar respuesta a la siguientes preguntas:

- ¿Cuántas tablas hay almacenadas?
- ¿Cuántos renglones tiene la tabla T?
- ¿Se dispone de espacio en disco para almacenar una nueva tabla?
- ¿Cuántas columnas tiene la tabla T?
- ¿Cuál es el nombre de las columnas de la tabla T?
- ¿Cuántos caracteres se pueden almacenar en cada columna de la tabla T?
- ¿Qué tipo de datos se pueden almacenar en cada columna de

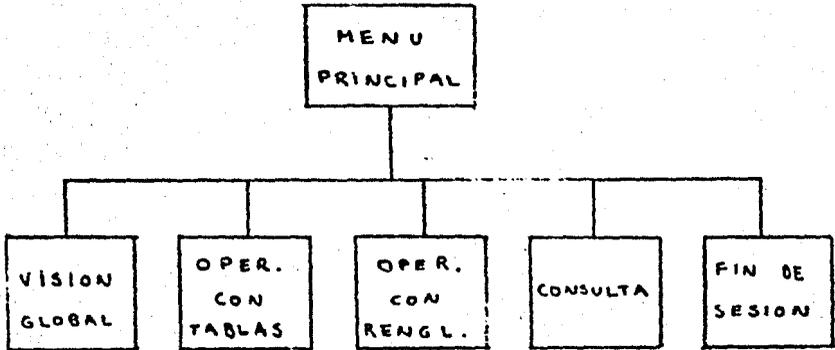


Fig.10 Menú Principal

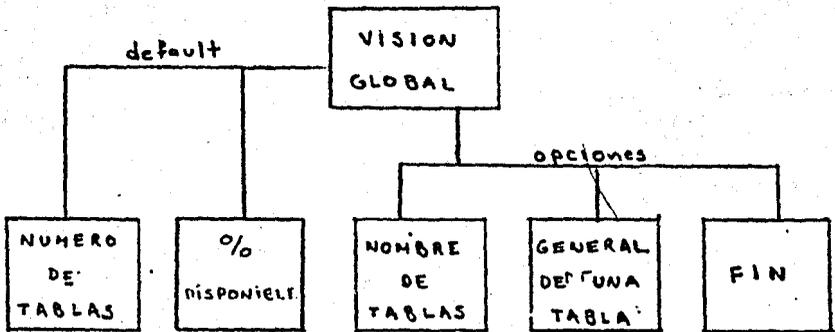


Fig.11 Menú de Vision Global.

la tabla T?

- ¿Cuál es la primera tabla que se almacenó?
- ¿Cuál es la última tabla que se almacenó?
- ¿Cómo se llama la tabla almacenada en el lugar n?
- ¿En qué lugar se almacenó la tabla T?

Al elegir esta opción, el usuario tiene acceso por default a la siguiente información:

- Numero de tablas almacenadas y
- Porcentaje de espacio disponible en disco.

Por medio del menú tiene la opción de conocer el nombre y algunas características de las tablas, obtener generalidades de alguna tabla o regresar al menú principal como se ve en la figura 11.

El menú de vision global es así:

```

+           MENU DE VISION GLOBAL
+
+  N(L)  NOMBRE DE TABLAS           G(L)  GENERALID. DE U/TABLA
+
+           F  FIN DE VISION GLOBAL
+
+  EXISTEN nn TABLAS ALMACENADAS
+  ESTA DISPONIBLE EL nn % DE ESPACIO EN DISCO
  
```

La opción H NOMBRE DE TABLAS, muestra además del nombre de las tablas, el lugar que ocupan cronológicamente y el número de renglones almacenados en cada una.

Con la opción G, GENERALIDADES DE UNA TABLA, el usuario conocerá el lugar que ocupa la tabla en la lista, el nombre,

longitud y tipo de las columnas de la tabla, y el nombre de las columnas que forman la llave. Ejemplo:

```

+ GENERALIDADES DE UNA TABLA
+ NOMBRE DE LA TABLA?
DESPENSA

+ GENERALIDADES DE LA TABLA: DESPENSA
+ COLUMNA      NOMBRE      LONGITUD  TIPO
+ 1            XPRODUCTO  15        A
+ 2            XMARCA     10        A
+ 3            PRECIO     4         D
+ 4            CANTIDAD   3         N
+ 5            EMPAQUE    10        A
  
```

El signo de porcentaje que precede a algunos nombres de columna, indica que la columna es llave.

Para regresar al menú principal el usuario elige la opción F.

#### 2.4) Operaciones con Tablas.

Con esta opción se pueden crear, eliminar, proyectar, clasificar, multiplicar, unir ó sacar diferencia entre tablas. Operaciones con tablas es la única opción que permite modificar la estructura de las tablas. Gráficamente tenemos:

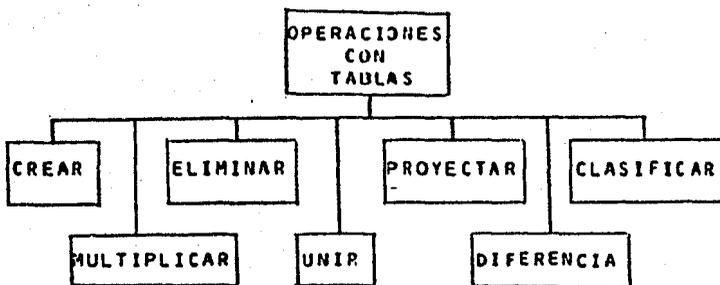


Fig. 12 Menú de Operaciones con Tablas.

En las últimas cinco opciones el usuario puede ver el resultado de la operación que acaba de efectuar agregando la letra D (despliegue) a la opción correspondiente.

El menú para operar con tablas es el siguiente:

```

+      MENU PARA OPERAR CON TABLAS
+  C CREAR      E ELIMINAR      P(D) PROYECTAR      S(D) CLASIFICAR
+  M(D) MULTIPLICAR      U(D) UNIR      D(D) DIFERENCIA      F FIN
  
```

#### 2.4.1) Crear Tabla.

Esta opción se refiere a crear la estructura de una nueva tabla. La información necesaria es:

- nombre de la tabla
- nombre, longitud y tipo de cada columna
- nombre de las columnas que forman la llave

No se admiten dos tablas con el mismo nombre. Únicamente los 8 primeros caracteres son significativos en el nombre de tabla y de columna.

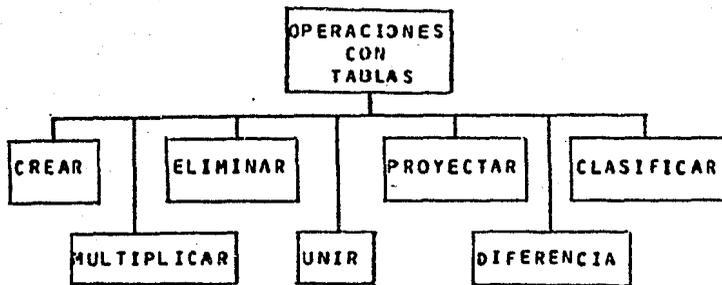


Fig. 12 Menú de Operaciones con Tablas.

En las últimas cinco opciones el usuario puede ver el resultado de la operación que acaba de efectuar agregando la letra D (despliegue) a la opción correspondiente.

El menú para operar con tablas es el siguiente:

```

+      MENU PARA OPERAR CON TABLAS
+  C CREAR      E ELIMINAR    P(D) PROYECTAR  S(D) CLASIFICAR
+  M(D) MULTIPLICAR  U(D) UNIR      D(D) DIFERENCIA  F FIN
  
```

#### 2.4.1) Crear Tabla.

Esta opción se refiere a crear la estructura de una nueva tabla. La información necesaria es:

- nombre de la tabla
- nombre, longitud y tipo de cada columna
- nombre de las columnas que forman la llave

No se admiten dos tablas con el mismo nombre. Únicamente los 8 primeros caracteres son significativos en el nombre de tabla y de columna.

La información contenida en cada columna puede ser de tipos A (alfanumérico), N (numérico) ó D (dinero). El tipo A contiene caracteres alfabéticos, numéricos ó especiales. El tipo N dígitos. El tipo D dispone de dígitos enteros; un punto y dos decimales se asumen.

Toda tabla debe tener una llave; no se permiten renglones duplicados ó con la misma llave.

Toda la información que el usuario proporciona es verificada por el sistema, que envía mensajes de error en caso necesario. Un ejemplo de diálogo entre sistema y usuario se muestra a continuación:

```

+          CREACION DE TABLA
+  NOMBRE DE LA TABLA QUE SE VA A CREAR?
DESPENSA
+  PARA TERMINAR HANDA UNA LINEA EN BLANCO
+
+  NOMBRE          TIPO (A, N, D)          LONGITUD
+  COLUMNA 1?     PRODUCTO ,          A          ,          15
+  COLUMNA 2?
MARCA , A , 35
+  COLUMNA 3?
PRECIO , D , 8

+  NOMBRE DE LAS COLUMNAS-LLAVE?
PRODUCTO , PRECIO

+  MODIFICACIONES? (S , N)
NO
+  TABLA CREADA: DESPENSA

```

#### 2.4.2) Eliminar Tabla.

Con esta opción es posible hacer desaparecer la información

contenida en una tabla y la estructura de la tabla. Por seguridad y protección, el usuario debe introducir dos veces el nombre de la tabla que desea borrar. Un ejemplo:

```

+           ELIMINACION DE TABLA
+
+   NOMBRE DE LA TABLA QUE SE PREVEE ELIMINAR?
DEPENSA
+   LA TABLA DEPENSA TIENE 24 RENGLONES
+   NOMBRE DE LA TABLA QUE SE ELIMINA DEFINITIVAMENTE?
DEPENSA
+
+   TABLA ELIMINADA: DEPENSA

```

Si el nombre de la tabla que se elimina definitivamente no es el mismo que el nombre de la tabla que se "prevee eliminar", el sistema repite nuevamente la segunda pregunta.

#### 2.4.3) Proyectar Tablas.

Con esta opción del menú, es posible crear la estructura de una nueva tabla semejante a la estructura de una tabla ya existente.

Proyectar se refiere a escoger ciertas columnas de la tabla, y con ellas formar una nueva estructura. La tabla original no se modifica. La información almacenada en las columnas seleccionadas pasa a formar parte de la nueva tabla. Gráficamente tenemos:

AMIGOSMD: creada a partir de AMIGOS  
 proyectando NOMBRE y TELEFONO

		AMIGOS	
AMIGOSMD		AMIGOS	
%NOMBRE	TELEFONO	%NOMBRE	EDAD SEXO TELEFONO

Fig.13 Proyeccion de una Tabla.

La nueva tabla contiene tambien una llave formada por una ó varias columnas que son seleccionadas por el usuario tomando en cuenta que el sistema elimina automaticamente los renglones duplicados.

Para llevar a cabo una proyeccion, el sistema necesita el nombre de la tabla, el nombre y orden de las columnas sobre las que se va a proyectar, el nombre de la tabla resultante y los nombres de las columnas-llave. Un ejemplo del dialogo se muestra a continuacion:

```

+          PROYECCION DE TABLA
+          NOMBRE DE LA TABLA QUE SE VA A PROYECTAR?
AMIGOS
+          COLUMNAS DE: AMIGOS
+          %NOMBRE          SEXO
+          %EDAD           TELEFONO
+          NOMBRE DE LAS COLUMNAS QUE SE VAN A PROYECTAR?
NOMBRE, TELEFONO
+          NOMBRE DE LAS COLUMNAS-LLAVE DE LA NUEVA TABLA?
NOMBRE
+          NOMBRE DE LA NUEVA TABLA?
AMIGOSMD
+
+          TABLA NUEVA: AMIGOSMD , CREADA A PARTIR DE: AMIGOS
+          PROYECTANDO : %NOMBRE, TELEFONO
+
+          AMIGOSMD TIENE 10 RENGLONES
  
```

Con esta opción, también es posible cambiar el orden de las columnas: esto se logra dando al sistema el nombre de las columnas en el orden adecuado y conservando la misma llave de la tabla original.

La utilidad de la proyección de una tabla, se verá en la etapa de consulta cuando se describan algunos ejemplos.

Si el usuario elige la opción PD (proyección-despliegue), la estructura y la información de la nueva tabla aparecen en la pantalla.

#### 2.4.4) Clasificar Tabla.

Con esta operación, los renglones de una tabla se reorganizan de acuerdo a un conjunto de columnas. El sistema debe conocer el nombre de la tabla, el nombre de las columnas que servirán para hacer la clasificación y el orden de la clasificación.

Un ejemplo para ilustrar el modo en que el sistema obtiene esta información se muestra:

```

+           CLASIFICACIÓN DE TABLA
+
+   NOMBRE DE LA TABLA QUE SE VA A CLASIFICAR?
DESPENSA
+   COLUMNAS DE:  DESPENSA
+                 %PRODUCTO          PRECIO          EMPAQUE
+                 %MARCA             CANTIDAD
+   NOMBRE DE LAS COLUMNAS POR LAS QUE SE VA A CLASIFICAR?
PRECIO, EMPAQUE
+   ORDEN ASCENDENTE O DESCENDENTE? (A , D)
A
+   TABLA DESPENSA CLASIFICADA POR LAS COLUMNAS PRECIO , EMPAQU
+   EN FORMA ASCENDENTE

```

Las cuatro operaciones anteriores, se efectúan sobre una sola tabla; en este sentido son operaciones unarias. En seguida se presentan tres operaciones al menos binarias: multiplicar, unir y diferencia de tablas; que dan como resultado una tercera.

#### 2.4.5) Multiplicar Tablas.

Con esta operación también llamada unión (join) ó "producto cruz", es posible crear la estructura de una nueva tabla como combinación de otras dos. Las dos tablas originales deben tener una columna-llave en común, es decir una columna con el mismo nombre, tipo y longitud, que servirá de eje para efectuar la multiplicación. Ejemplo:

VENDEDOR

ZNOMBRE	ZMERCAN
saul	silla
juan	mesa
tomas	silla
saul	mesa
juan	silla
saul	estufa

X

PRODUCTO

ZMERCAN	COSTO	PRECIO
silla	50	80
mesa	400	400
estufa	600	900

VENDPROD

ZNOMBRE	ZMERCAN	COSTO	PRECIO
saul	silla	50	80
juan	mesa	400	400
tomas	silla	50	80
saul	mesa	400	400
juan	silla	50	80
saul	estufa	600	900

El resultado de la multiplicación es una tabla cuyo número de columnas es la suma del número de columnas de las tablas

originales menos 1 y cuya columna-llave es el conjunto formado por las columnas-llave de las tablas originales. A continuación se muestra un ejemplo del diálogo necesario para multiplicar dos tablas:

```

+           MULTIPLICACIÓN DE TABLAS
+
+           NOBRE DE LAS TAUAS QUE SE VAN A MULTIPLICAR?
VENDEDOR, PRODUCTO
+           NOBRE DE LA COLUMNA-EJE?
MERCAN
+           NOBRE DE LA TABLA RESULTANTE?
VENDPROD
+
+           TABLA NUEVA: VENDPROD , CREADA MULTIPLICANDO
+           VENDEDOR Y PRODUCTO SOBRE MERCAN
+
+           VENDPROD TIENE 6 RENGLONES

```

#### 2.4.6) Unir Tablas.

Esta es una operación muy sencilla que sin embargo resulta de mucha utilidad en la etapa de consulta. Con esta operación es posible "juntar" dos tablas o más, una debajo de la otra; para que la unión pueda efectuarse, las tablas deben tener la misma estructura, esto es el mismo número de columnas y el nombre, longitud y tipo de cada columna también debe ser igual. El resultado de la operación es una tabla con estructura idéntica a las otras dos, que contiene la información de todas las tablas originales. Los renglones duplicados se han eliminado de la siguiente forma: A+B contendrá todos los renglones de A y los renglones de B que no sean duplicados; así pues unir tablas es una operación

no conmutativa. A continuación se muestra un ejemplo del diálogo:

```

+          UNION DE TABLAS
+
+  NOMBRE DE LAS TABLAS QUE SE VAN A UNIR?
DESPENS1, DESPENS2
+  NOMBRE DE LA TABLA RESULTANTE?
DESPENS3
+
+  TABLA NUEVA: DESPENS3 , CREADA A PARTIR DE LA UNION
+                DE DESPENSI Y DESPENSI2
+
+  DESPENS3 TIENE 17 RENGLONES

```

#### 2.4.7) Diferencia de Tablas.

Mediante esta opción el usuario puede "restar" dos tablas que tengan la misma estructura. Se muestra un ejemplo y el diálogo correspondiente:

LIBROS

ZAUTOR	ZTITULO
cervantes	quijote
azueta	los de abajo
paz	laberinto
marquez	cien años
hess	lobo

LIBPREST

ZAUTOR	ZTITULO
azueta	los de abajo
paz	laberinto

LIBCASA

ZAUTOR	ZTITULO
cervantes	quijote
marquez	cien años
hess	lobo

```

+          DIFERENCIA DE TABLAS
+
+  NOMBRE DE LA TABLA "MINUENDO"?
LIBROS
+  NOMBRE DE LA TABLA "SUSTRAENDO"?
LIBPREST
+  NOMBRE DE LA TABLA RESULTANTE?
LIBCASA

```

- +
  - +
  - +
  - +
  - +
- TABLA NUEVA: LIBCASA , CREADA A PARTIR DE LA  
DIFERENCIA ENTRE LIBROS Y LIBPREST
- LIBCASA CONTIENE 3 RENGLONES

El sistema se encarga de verificar que todos los datos proporcionados por el usuario son correctos y que las primeras dos tablas tengan la misma estructura.

## 2.5) Operaciones con Renglones.

Esta opción del menú principal, permite al usuario operar ya no con toda la tabla, sino con los renglones. Se pueden insertar, eliminar, modificar o copiar renglones de una tabla a otra. Gráficamente tenemos:

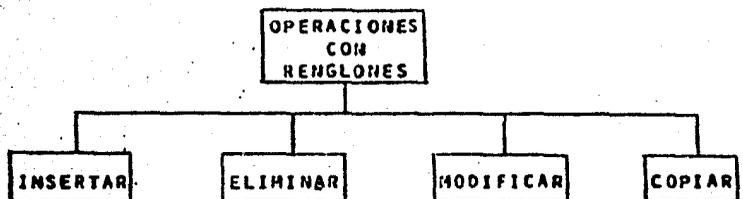


Fig.14 Menú de Operaciones con Renglones.

El menú para operar con renglones es el siguiente:

```

+           MENU PARA OPERAR CON RENGLONES
+
+ I INSERTAR      E ELIMINAR      M MODIFICAR
+ C COPIAR DE U/TARLA A OTRA      F FIN
  
```

### 2.5.1) Insertar Renglones.

Con esta opción el usuario puede insertar renglones al final de la tabla. Además de validar que la tabla donde se desea añadir renglones exista, el sistema verifica lo siguiente:

- Duplicidad de renglones
- Estructura del renglón: cada campo del renglón debe ser del mismo tipo que la columna correspondiente y la longitud no debe sobrepasar la longitud de la columna.

Si cualquiera de estas dos verificaciones falla, el sistema manda un mensaje de error e ignora todo el renglón.

Antes de iniciar la introducción de renglones, el sistema proporciona al usuario la estructura de la tabla; de este modo el usuario no tiene que memorizar la longitud y tipo de las columnas. Esta es una de las muchas características que hacen a MI1 un sistema fácil de operar, totalmente orientado al usuario.

Existen tres reglas sintácticas que se deben observar cuando se introducen renglones:

- 1) Los campos van separados por comas y opcionalmente por blancos. El blanco en este sistema es un caracter especial: solo se toma en cuenta si aparece en medio de un campo; es compactable; cualquier numero de blancos se transforman en uno. Cualquier campo, excepto las columnas-llave puede estar vacío; el sistema lo interpreta como el campo " ".
- 2) Cada renglón se inicia en una nueva línea.
- 3) La introducción de renglones se termina con una línea en blanco.

Se muestra a continuación ejemplo de un diálogo para introducir renglones.

```

+          INSERCIÓN DE RENGLONES
+
+          NOMBRE DE LA TABLA?
PROFHATE
+          COLUMNAS DE: PROFHATE
+          COLUMNA      NOMBRE      LONGITUD  TIPO
+             1          ZPROFESOR    15         A
+             2          MATERIA      20         A

```

```

+           3           SALON           4           N
+           4           HORA            2           N
+
+   PROFMATE CONTIENE 10 RENGLONES
+
+   1) SEPARA LOS CAMPOS CON COMAS
+   2) PARA TERMINAR MANDA UNA LINEA EN BLANCO
+
+   RENGLÓN 1?
MARTINEZ, CALCULO ACTUARIAL, 1324, 17
+   RENGLÓN 2?
PERALTA, ALGEBRA LINEAL II, 1739, 10
+   RENGLÓN 3?
+
+   SE INSERTARON 2 RENGLONES A LA TABLA PROFMATE

```

### 2.5.2) Eliminar Renglones.

Esta función sirve para que el usuario borre algunos renglones de una determinada tabla.

Cuando una tabla se queda sin renglones se dice que esta vacía. Su estructura permanece y en cualquier momento se pueden volver a introducir renglones a ella. La estructura de una tabla se elimina desde operaciones con tablas.

De la información proporcionada por el usuario, el sistema verifica: que la tabla exista, y que los renglones también existan. En caso contrario el sistema manda un mensaje de error y repite la pregunta. Un ejemplo del diálogo es:

```

+           ELIMINACIÓN DE RENGLONES
+
+   NOMBRE DE LA TABLA?
PROFMATE
+   COLUMNAS-LLAVE DE PROFMATE: PROFESOR
+
+   1) INTRODUCE LA LLAVE DE LOS RENGLONES
+   2) PARA TERMINAR MANDA UNA LINEA EN BLANCO
+
+   RENGLON 1?

```

PERALTA

+ RENGLO 2?

JUAREZ

+ RENGLO 3?

+

+ SE ELIMINARON 2 RENGLONES DE LA TABLA PROFMATE

### 2.5.3) Modificar Renglones.

En esta opción, el usuario puede cambiar el contenido de uno o varios campos de un renglon. Existen dos tipos de modificación:

- por llave: el usuario puede modificar uno o varios campos de un renglon si conoce la llave.
- global por columna: todos los campos bajo una cierta columna se modifican. Se vera la utilidad de esta operacion en la seccion 2.9.

Al seleccionar la opción M MODIFICAR, el sistema despliega el siguiente menú:

```

+           MENU PARA MODIFICAR RENGLONES
+
+   L(D) LLAVE: MODIFICACION POR           F FIN
+           C(D) COLUMNA: GLOBAL POR

```

- Modificación por llave: el sistema verifica que la tabla, el renglón y el campo que se van a modificar existan y además que la nueva información sea consistente con la estructura de la tabla.

A continuación se muestra un ejemplo:

```

+           MODIFICACION DE RENGLONES POR LLAVE
+
+   NOMBRE DE LA TABLA?
PROFMATE

```

```

+
+ COLUMNAS DE: PROFMATE
+ COLUMNA      NOMBRE          LONGITUD      TIPO
+
+ 1            ZPROFESOR       15            A
+ 2            MATERIA        20            A
+ 3            SALON          4             N
+ 4            HORA           2             N
+
+ 1) INTRODUCE LA LLAVE DEL RENGLON,
+    EL NOMBRE O NUMERO DE LA COLUMNA Y
+    LA NUEVA INFORMACION
+
+ 2) PARA TERMINAR MANDA UNA LINEA EN BLANCO
+
+ LLAVE      ,      COLUMNA      ,      NUEVA INFORMACION
+
+ MODIFICACION 1?
+ ARROYO, SALON, 1794
+ MODIFICACION 2?
+ PERALTA, 4, 19
+ MODIFICACION 3?
+
+ SE HICIERON 2 MODIFICACIONES A LA TABLA: PROFMATE

```

Esta forma de hacer el cambio tiene la gran ventaja de que aún el campo-llave puede modificarse; en otros sistemas es necesario primero eliminar el renglón y después insertar otro renglón con la nueva información en los campos-llave. Dicho de otro modo, modificar el campo-llave en otros sistemas conceptualmente toma el doble de tiempo que en MI1.

Este método tiene la desventaja -sin embargo- de que se debe introducir una línea por cada modificación que se desea efectuar.

- Global por columna: un ejemplo del diálogo para modificar los renglones de una tabla en forma global por columna es este:

```

+ MODIFICACION DE RENGLONES GLOBAL POR COLUMNA
+

```

```

+      NOMBRE DE LA TABLA?
DEPARTAM
+
+      COLUMNAS DE: DEPARTAM
+      COLUMNA      NOMBRE      LONGITUD      TIPO
+      1            XLLAVE      3            N
+      2            PRECIO      5            N
+      3            NURECAM     1            N
+      4            COLONIA     10           A
+
+      NOMBRE DE LA COLUMNA?
PRECIO
+      NUEVA INFORMACION?
12345
+
+      EN LA TABLA DEPARTAM SE MODIFICO LA COLUMNA PRECIO

```

En ambas opciones el usuario puede ver el resultado de las modificaciones introduciendo LD o CD según corresponda.

#### 2.5.4) Copiar Renglones.

Por medio de esta función, es posible copiar renglones de una tabla a otra. Los elementos necesarios para efectuar esta función son: el nombre de la tabla fuente, el nombre de la tabla receptora y las llaves de los renglones que van a copiarse.

El sistema verifica que la estructura de la tabla fuente sea igual a la estructura de la tabla receptora y que los renglones existan. Los renglones copiados no se eliminan de la tabla fuente. Un ejemplo del diálogo se muestra a continuación:

```

+      COPIA DE RENGLONES
+
+      NOMBRE DE LA TABLA FUENTE?
PROFMATE
+      NOMBRE DE LA TABLA RECEPTORA?
PROFHODI

```

+  
+ COLUMNAS-LLAVE DE PROFMATE: PROFESOR  
+  
+ 1) INTRODUCE LA LLAVE DE LOS RENGLONES  
+ 2) PARA TERMINAR MANDA UNA LINEA EN BLANCO  
+  
+ RENGLO 1?  
MARTINEZ  
+ RENGLO 2?  
FERNANDEZ  
+ RENGLO 3?  
  
+ SE COPIARON 2 RENGLONES DE LA TABLA:  
+ PROFMATE A PROFMODI

## 2.6) Consulta.

La etapa de consulta de MI1, es lo que en algunos otros sistemas se ha denominado "query language" (ó lenguaje de consulta). Esta función es el corazón del sistema. En esta etapa se recupera y explota la información almacenada en las tablas.

Una consulta consiste en seleccionar ciertos renglones de una tabla. Los renglones seleccionados forman otra tabla, que puede destruirse ó almacenarse en forma definitiva. Si se almacena, puede llevarse a cabo otra consulta sobre ella y así sucesivamente hasta llegar a la información específica que el usuario requiere, tal como se indica en la sección 2.9.

La consulta se efectúa sobre una sola tabla. Existen tres funciones para seleccionar los renglones almacenados en ella: consulta por rango, por llave y con condición. A continuación se muestra el menú de consulta:

```

+           MENU DE CONSULTA
+
+   R(L)  POR RANGO           L(L)  POR LLAVE
+   C(L)  CON CONDICION      F     FIN

```

### 2.6.1) Consulta por rango.

Con esta función se puede ver un renglón o conjunto de renglones si se conoce su posición en la tabla. Un ejemplo:

```

+           CONSULTA POR RANGO
+

```

```

+ NOMBRE DE LA TABLA?
DEPARTAM
+ RANGO DE RENGLONES? (MIN 1, MAX 50)
1-3, 7, 9
+
+ REGLON      XLLAVE      PRECIO      NURECAM      COLONIA
+
+ 1           347         6000        2            jardin
+ 2           351         12000       1            juarez
+ 3           386         26000       2            copilco
+ 7           529         3000        1            educacion
+ 9           536         20000       2            san jeronimo
+
+ SE ALMACENA ESTA TABLA DEFINITIVAMENTE? (S , N)
SI
+ NOMBRE DE LA TABLA?
DEPARTAZ
+ TABLA NUEVA : DEPARTAZ

```

### 2.6.2) Consulta por Llave.

Con esta opción es posible localizar un conjunto de renglones cuando se conoce su llave. Para localizar un rango de renglones es necesario que la tabla este ordenada en forma ascendente. A continuación se muestra un ejemplo:

```

+ CONSULTA POR LLAVE
+
+ NOMBRE DE LA TABLA?
DIBUJOS
+
+ COLUMNAS-LLAVE DE DIBUJOS: NOMBRE
+
+ 1) INTRODUCE LA LLAVE DE LOS RENGLONES
+ UNA POR UNA O UN RANGO DE LLAVES
+ 2) PARA TERMINAR MANDA UNA LINEA EN BLANCO
+
+ REGLON 1?
DRAGON
+ REGLON 2?
PANTERA - QUIQUE
+ REGLON 3?

+ XNOMBRE      TAMANO      MATERIAL      FECHA
+
+ DRAGON       50x40      ACRILICO      NOV 79
+ PANTERA      20x30      TINTA         ENE 81
+ PINGUIN      90x10      ACUARELA      JUN 82

```

```

+ QUIQUE          90x90          MADERA          MAR 78
+
+ SE ALMACENA ESTA TABLA DEFINITIVAMENTE? (S , N)
S
+ NOMBRE DE LA TABLA?
DIBPREF
+ TABLA NUEVA: DIBPREF

```

Si alguno de los renglones que el usuario desea consultar no se encuentran en la tabla, el sistema manda un mensaje de error.

### 2.6.3) Consulta con Condición.

En esta opción se despliegan todos los renglones que cumplen con alguna característica especial o condición. Algunos ejemplos son: la columna COLONIA es igual a TLALPAN, la columna PRECIO es menor que 6000, etc. Una condición esta formada por: el nombre de una columna, una relación y una constante.

La columna puede ser alguna de las incluídas en la tabla que se esta consultando. La relación es cualquiera de las siguientes: IGUAL, DIFERENTE, MENOR QUE y MAYOR QUE. La constante es cualquier cadena de caracteres del mismo tipo que la columna. Un ejemplo:

```

+ CONSULTA CON CONDICION
+
+ NOMBRE DE LA TABLA?
TERRENOS
+
+ COLUMNAS DE: TERRENOS
+ COLONIA          PRECET          XCALLE
+ METROS           NOMBRE          XNUMERO
+
+ CONDICION?
+ COLUMNA?

```

## COLONIA

+ CARACTERISTICAS DE LA COLUMNA: COLONIA  
 + LONGITUD: 15  
 + TIPO: A

+ RELACION? (IGUAL, DIF, MENOR, MAYOR)

## IGUAL

+ CONSTANTE?

## DEL VALLE

+ EXISTEN 2 RENGLONES QUE CUMPLEN LA CONDICION.  
 + QUIERES VERLOS? (S, N)

S

COLONIA	METROS	PRECMET	NOMBRE	%CALLE	XNUMERO
DEL VALLE	824	3500	LOPEZ	P.SAENZ	247
DEL VALLE	328	4900	DIAZ	MAGDALENA	104

+ SE ALMACENA ESTA TABLA DEFINITIVAMENTE? (S, N)

N

## 2.7) Fin de Sesión.

Para terminar una sesión, se requiere que el sistema despliegue el menú principal. Esto se logra terminando todos los menús anteriores por medio de una f. Al terminar la sesión el sistema manda el siguiente mensaje:

+ TERMINA SESION CON M11

y algunos datos estadísticos.

## 2.8) Entrada Adelantada de Respuestas para Usuarios Experimentados.

En la sección 1.6 se menciona la desventaja de la operación a base de menús, que consiste en que si un usuario cuenta con alguna experiencia con el sistema y ya conoce las opciones a los menús y las respuestas que debe teclear, el

despliegue de menús y preguntas puede ser una operación tediosa que consume tiempo innecesario.

Para eliminar esta desventaja, MI1 cuenta entre sus características con entrada adelantada de respuestas. Esto significa que el usuario puede -si así lo desea- responder al sistema antes de que este le formule la pregunta y puede adelantar tantas respuestas como desee. Cada respuesta debe separarse de la siguiente por medio de un punto y coma (;).

Como un ejemplo, en el menú principal el usuario podría introducir la siguiente cadena:

V; N; G; ALUMNOS; F; T; C; ARCHIVO

con esto indica al sistema lo siguiente:

- deseo ver el menú de vision global (V)
- quiero ver el nombre de las tablas almacenadas (N)
- quiero ver las generalidades (G) de la tabla ALUMNOS
- deseo terminar con el menú de vision global (F)
- quiero entrar al menú de operaciones con tablas (T)
- deseo crear (C) una tabla cuyo nombre es ARCHIVO

Con esta serie de comandos, el usuario ha evitado que el sistema despliegue varias preguntas y menús que en este caso no eran necesarios. Ahora bien, si alguna de las respuestas del usuario no es correcta, las respuestas siguientes se ignoran. Para terminar veamos otro ejemplo:

Supongamos que el sistema tiene desplegado el menú

principal. El usuario introduce la siguiente cadena:

T; DD; LIBROS; LIBPREST; LIBCASA; F; F

que significa:

- operar con tablas (T)
- efectuar una diferencia de tablas y desplegar el resultado (DD)
- la tabla "minuendo" es LIBROS
- la tabla "substraendo" es LIBPREST
- la tabla resultante es LIBCASA
- (en este momento aparece en pantalla la tabla LIBCASA)
- termina operacion con tablas (F)
- termina sesion con el sistema (F)

## 2.9) Combinaciones Inter-menús.

Tal como se mencionó en la sección 2.1.2, uno de los objetivos de MI1 es el proporcionar un conjunto de operaciones básicas, sencillas y fáciles de usar, que se puedan combinar en muchas formas, para llevar a cabo operaciones mas complejas.

En esta sección se daran varios ejemplos de cómo se pueden combinar estas operaciones básicas.

a) Objetivo: debido a factores económicos, todos los profesores que trabajan en el departamento de contabilidad(clave 4) van a ganar 15000. Para cumplir con

este objetivo, en la tabla PROFESOR (XNOMBRE, DEPARTAM, ESTUDIO, SUELDO, POSICION) el SUELDO del personal del departamento 4 debe cambiarse a 15000.

Secuencia de operaciones básicas:

1) Por medio de consulta con condición, se obtienen los renglones cuya columna DEPARTAM valga 4. Esta tabla se almacena con el nombre PROF4.

PROF4

XNOMBRE	DEPARTAM	ESTUDIO	SUELDO	POSICION
urrutia	4	mc	17000	asociado
lens	4	lic	12000	ayudante

2) Usando modificación global por columna, se actualiza la columna sueldo de la tabla PROF4 que ahora contiene 15000.

3) Unir tablas PROF4 y PROFESO, la tabla resultante es PROFESO. Los renglones duplicados se eliminan, dando preferencia a los renglones de la primera tabla.

4) Eliminar tabla PROF4.

Con estas cuatro operaciones básicas, hemos llevado a cabo un proceso complejo. Es verdad que en algunos lenguajes de consulta esta operación podría efectuarse directamente en un solo paso; sin embargo, agregar operaciones adicionales a las básicas, tiene el inconveniente de que el usuario tiene que invertir una cantidad considerable de tiempo en aprender a operar el sistema, cosa que no sucede en MI1 pues como ya se ha visto, el sistema guía al usuario en todo momento.

b) Objetivo: deseo ver impresos todos los TERRENOS (COLONIA, METROS, PRECMET, NOMBRE, XCALLE, XNUMERO, COSTO) que cumplan con la siguiente característica:

- costo menor ó igual a 1 500 000 y que se localicen en Tepepan ó Copilco.

Secuencia de operaciones básicas:

1) En consulta con condición, obtener una tabla TER1500 que contiene los renglones de la tabla TERRENOS cuyo COSTO es menor a 1 500 001.

2) Sobre la tabla TER1500 efectuar una consulta con condición pidiendo que la columna COLONIA sea igual a Tepepan, se almacena como TTEPEP.

3) Repetir paso anterior con COLONIA igual a Copilco, se almacena en TCOPIIL.

4) Unir tablas TTEPEP y TCOPIIL en la tabla RESULTAD.

5) En consulta con rango, se listan todos los renglones de la tabla RESULTAD.

6) Se eliminan tablas TER1500, TTEPEP y TCOPIIL.

c) Objetivo: anteriormente no se necesitaba toda la información; únicamente el nombre de la calle, y el número donde están ubicados los terrenos. Además, se desea primero el número sobre la calle, la calle y la colonia.

Secuencia de operaciones básicas:

1) Proyectar la tabla RESULTAD sobre las columnas XNUMERO, XCALLE y COLONIA y dejarla en OTRATABL.

2) Consultar por rango la OTRATABL listando todos los renglones.

d) Objetivo: se necesita una lista de los granos que estan escasos y los telefonos de los proveedores para poder hacer los pedidos. (Producto escaso es igual a menos de 100 unidades en existencia).

Tablas originales:

GRANOS ( XPRODUCTO, XMARCA, COSTO, EXISTEN)

PROVGRA ( XMARCA, TELEFONO)

Secuencia de operaciones básicas:

- 1) En consulta con condición, obtener una tabla con los productos que estan escasos (EXISTEN menor a 100), guardarla como ESCASOS.
- 2) Multiplicar tablas ESCASOS y PROVGRAM sobre la columna XMARCA; poner el resultado en MULTIPLI.

MULTIPLI

XPRODUCTO	XMARCA	COSTO	EXISTEN	TELEFONO
frijol b	la luz	70	50	6861747
arroz	marikita	150	0	5285744
frijol b	marikita	90	90	5285744
trigo	umum	300	33	3731290

- 3) Proyectar MULTIPLI sobre las columnas XPRODUCTO y TELEFONO dejando como llave ambas columnas; la tabla resultado es PRODTELE.
- 4) consultar con listado y por rango la tabla PRODTELE.
- 5) Eliminar las tablas ESCASOS, MULTIPLI y PRODTELE.

e) Objetivo: de la tabla PELICULAS (XTITULO, DIRECTOR, PRODUCTOR, PAIS, AÑO) eliminar todos los registros en donde el director esté en el rango Fernandez - Mendoza.

Secuencia de operaciones básicas:

- 1) Clasificar tabla PELICULAS en orden ascendente por DIRECTOR.
- 2) En consulta por llave formar una tabla con todos los registros cuyo director este en el rango Fernandez - Mendoza. La tabla se llama PELIC-ELIM.
- 3) Restar tablas PELICULAS y PELIC-ELIM y dejar el resultado en PELICULAS.

### Capítulo 3: Diseño de la Implementación.

Anteriormente hemos presentado la parte externa del sistema manejador de base de datos MI1; esto es, la interfase del usuario que es la columna vertebral de cualquier sistema de cómputo.

En este capítulo, expondremos la parte interna, que consiste en reconocer y diseñar las estructuras adecuadas para manipular la información almacenada en la computadora y los algoritmos necesarios para llevar a cabo todas las operaciones que el usuario efectúa.

#### 3.1) Estructuras de Datos.

La información que se almacena en la computadora es un conjunto selecto de datos sobre el mundo real. Los datos representan una abstracción de la realidad y es necesario elegir un modelo para representarla.

Al conjunto de modelos que tenemos para representar objetos en computadora se le llama estructuras de datos.

Nuestro sistema tendrá dos tipos de estructuras: las internas y las externas.

Las internas se refieren a la información que estará almacenada en la memoria de la computadora y las estructuras externas están relacionadas con él o los modelos usados para

almacenar información en dispositivos secundarios, específicamente floppy disks.

### 3.1.1) Externas.

Una de las características más importantes de las estructuras de datos externas es que permanecen almacenadas aún después de terminada una sesión con MI1. La información que se desea conservar entre sesión y sesión tendrá pues que guardarse de esta forma.

Las estructuras de datos externas se encuentran almacenadas en los siguientes archivos:

- El archivo DIREC-TABLAS es un directorio de tablas, conteniendo el nombre y el número de renglones de cada una de las tablas almacenadas por el usuario.

El primer registro da información sobre el número total de tablas en el sistema y el porcentaje de espacio disponible en disco. Este archivo está organizado en orden cronológico (último registro-última tabla almacenada).

primer registro:

NUM-TAB	2	(número de tablas)
PORC-DISC	3	(porcentaje de disco) (disponible)

otros registros:

NOM-TAB	k	(nombre de la tabla)
NUM-REN	2	(número de renglones)
NUM-COL	2	(número de columnas)

El usuario podrá definir el valor de k, es decir el número máximo de caracteres que contiene el nombre de la tabla y

columnas.

- Un archivo por cada tabla creada.

El primer registro contiene información describiendo la tabla:

NUM-COL	2	(número de columnas)
CRTR-COL		(característica de la columna)
	k	nombre
	2	longitud
	1	tipo
	1	orden de llaves

Este registro es de longitud variable. El número de ocurrencias del campo CRTR-COL esta determinado por el valor de NUM-COL.

Los siguientes registros contienen renglones de información almacenados por el usuario, mas un caracter extra CLAVE para marcar los renglones eliminados.

**Aspectos físicos de los discos:** las estructuras de datos externas presuponen que la computadora permite la escritura a disco de registros de longitud variable. Por ejemplo, el primer registro del archivo donde se almacena cada tabla mide  $2+n(k+4)$  caracteres, donde n es el número de columnas de la tabla; este número puede ser distinto para cada tabla. Otros registros del archivo tienen una longitud tambien variable dependiendo de la tabla.

Ciertas computadoras personales sólo permiten escritura a disco de registros de longitud fija, típicamente 128 ó 256 caracteres. Para estas computadoras, se escribiría una rutina

de bloqueo con objeto de formar los registros de longitud variable a partir de registros de longitud fija. De esta manera, los aspectos físicos de los discos quedan ocultos en la rutina de bloqueo, sin afectar el resto de la implementación. Esta rutina podría optimizar la localización de los registros lógicos en un registro físico; por ejemplo, si los registros lógicos miden 120 caracteres y el registro físico 128, la rutina podría automáticamente "empacar" el registro lógico con 8 caracteres extras de relleno. Nuevamente, ni el usuario ni el resto de la implementación necesitan tomar en cuenta este detalle de eficiencia.

### 3.1.2) Internas.

Las estructuras de datos almacenadas en la memoria de la computadora son:

- DESC-TAB : arreglo de elementos.

primer elemento

k nombre de la tabla  
2 numero de columnas

siguientes elementos

k nombre de la columna  
2 longitud de la columna  
1 tipo de la columna  
1 orden de llaves

- R-NOM-CLM : Un arreglo de elementos conteniendo el nombre de las columnas.

- R-NOM-TAB : nombre de tablas.

- R-OPCION: opción seleccionada en los menús.

- R-LIST-DES: indica si se desea listar o desplegar la

información.

- R-ASC-DES: la clasificación se hará en forma ascendente o descendente.
  - R-SI-NO: booleana, guarda respuestas del tipo (SI,NO).
  - R-RELACION: indica IGUAL, DIFERENTE, MAYOR o MENOR; se usa en consulta.
  - R-CONSTANTE: especifica la constante en consulta.
  - R-REGLON: un renglón de información.
  - R-LLAVE: llave del renglón, usada para eliminar, modificar o copiar renglones.
  - R-RANGO: rango de renglones que se desea consultar.
  - R-NVA-INFO: nueva información al modificar renglones.
  - R-DESPL: booleana, indicando si se despliega el resultado de operaciones.
  - NUM-REN : número de renglones.
  - NUM-CLM: número de columnas.
  - R-CLM-EJE: columna eje al efectuar la multiplicación.
- Un máximo de tres tablas podrán ser usadas simultáneamente.

### 3.2) Diagramas de Warnier-Orr.

Iniciado en 1976 en Francia por Warnier y extendido en los Estados Unidos por Orr, el concepto de especificar programas por medio de conjuntos, ha resultado ser una herramienta muy útil y poderosa.

Los diagramas de Warnier-Orr están basados en el concepto de

que un proceso complejo puede descomponerse fácilmente en varios subprocesos mas sencillos, que a su vez pueden descomponerse, llegando finalmente a procesos que resultan manejables.

Esta idea no es nueva y se conoce por varios nombres: programación de arriba-abajo, ó por refinamiento sucesivo; - programación modular, etc. Los diagramas Warnier-Orr proporcionan una forma gráfica y clara de mostrar ésta descomposición, particularmente útil para mostrar la estructura jerárquica de los sistemas.

El manejo de excepciones en todo sistema, es un aspecto que requiere mucho trabajo y detalle y que en general no proporciona información adicional sobre el sistema. Por esta razón, el manejo de errores no se describirá en los diagramas, asumiendo que los datos de entrada son correctos.

El sistema se ha dividido en 5 módulos principales como lo indica la figura 15. La subdivisión de los cuatro primeros módulos se muestra en las figuras 16-19.

Los diagramas de Warnier-Orr se presentan a continuación.

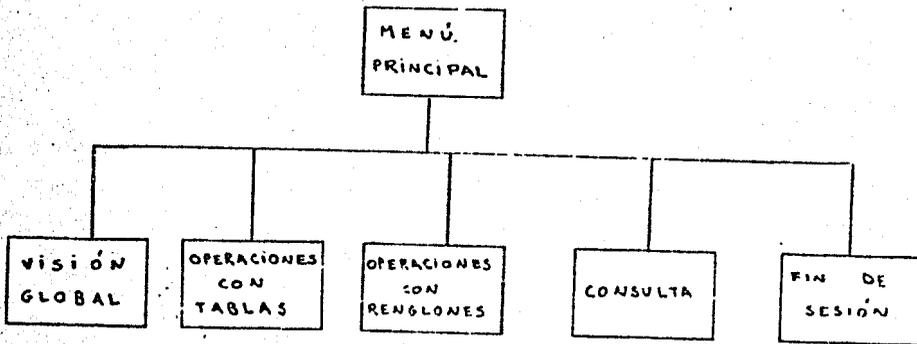


Fig. 15 Diagrama General del Sistema.

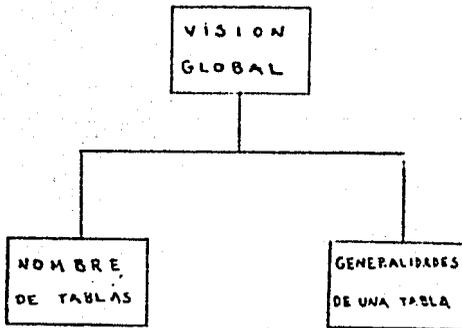


Fig. 16 Diagrama de Visión Global.

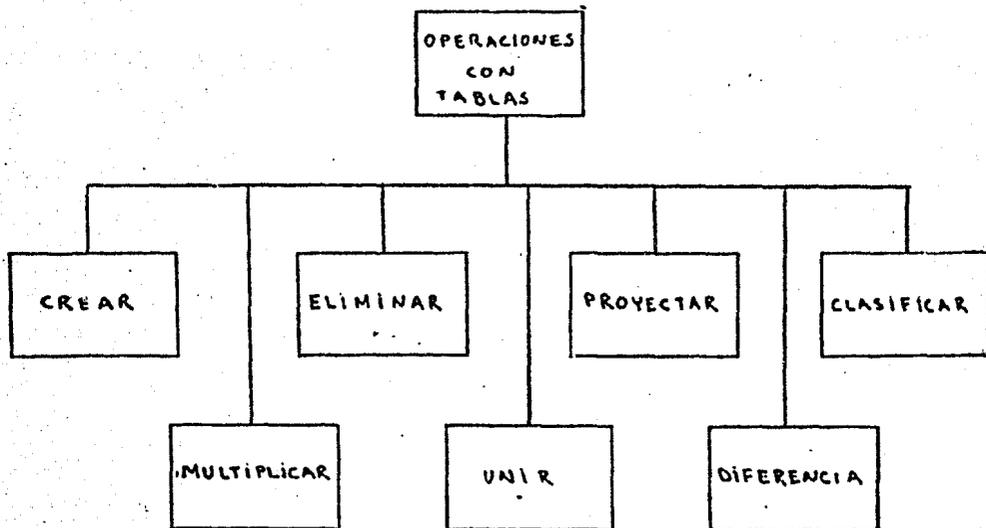


Fig.17 Diagrama de Operaciones con Tablas.

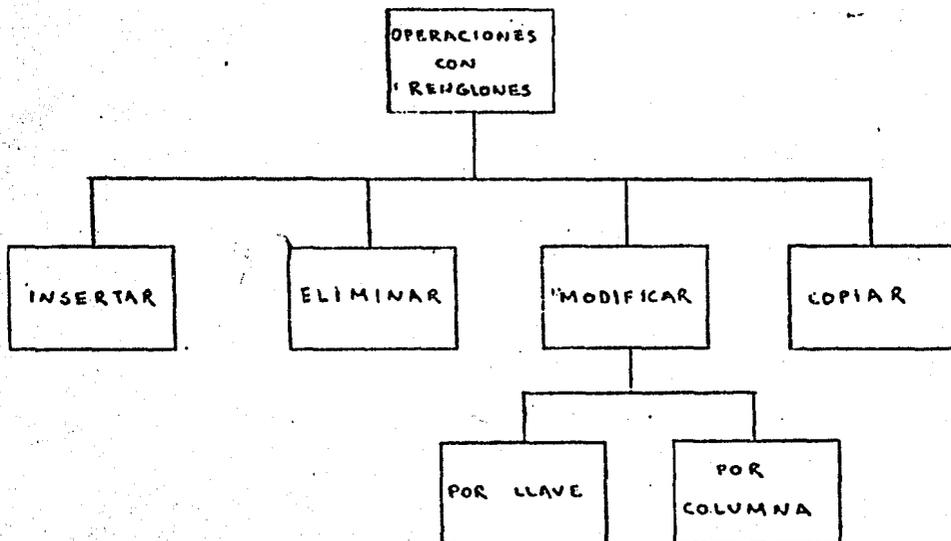


Fig.18 Diagrama de Operaciones con Renglones.

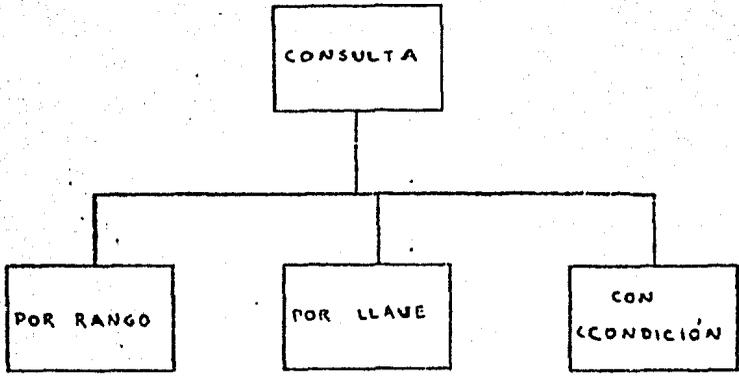
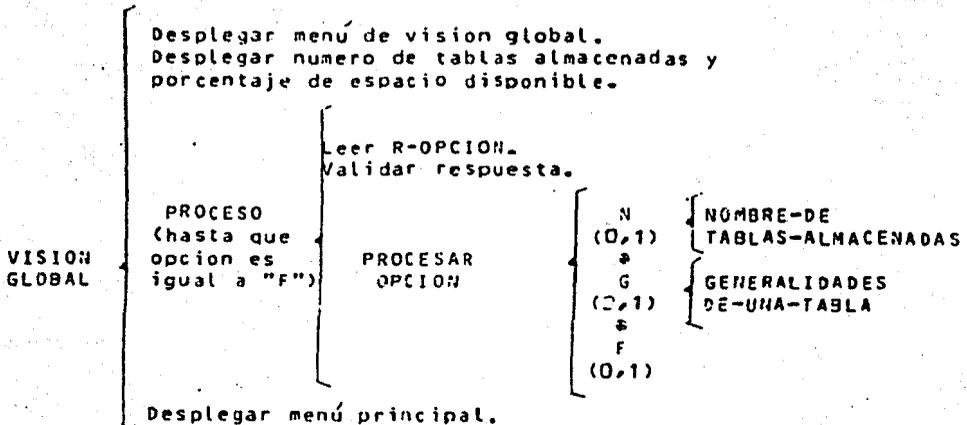
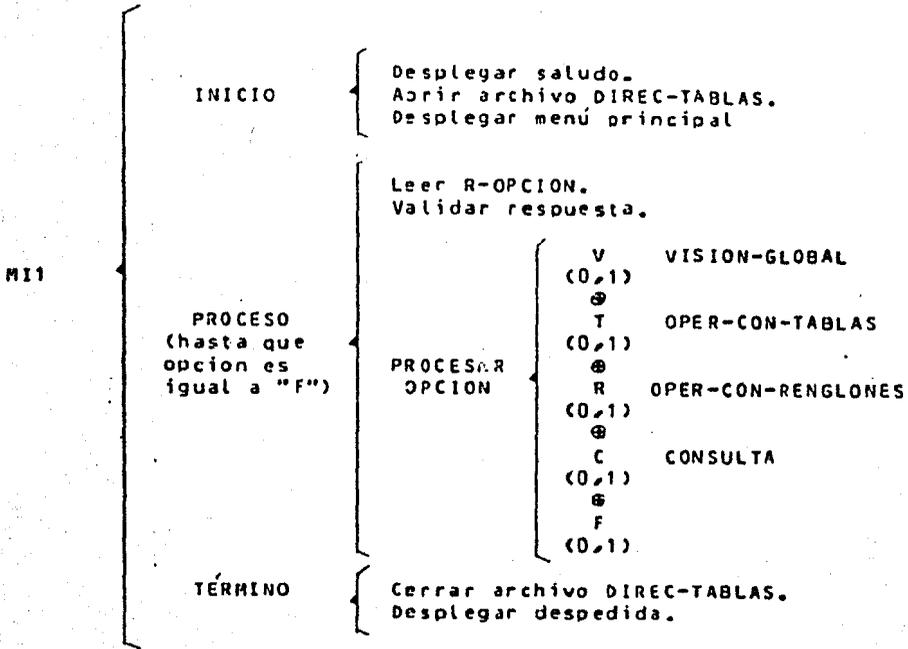


Fig.12 Diagrama de Consulta.



TABLAS  
 ALMACENADAS

- Abrir archivo DIREC-TABLAS.  
 Escribir "Nombre de tablas almacenadas",  
     "Orden...Nombre de Tabla...No.Renglon"
- LEER-ESC (1,NO-TABLAS)
  - Leer un registro de DIREC-TABLAS.
  - Escribe INDICE NOM-TAB NUM-REN.
- Cerrar archivo DIREC-TABLAS.

GENERALIDADES  
 DE-UNA-TABLA

- Escribir "Generalidades de una Tabla".  
     " Nombre de la tabla?".  
 Leer R-NOM-TAB1.  
 Validar respuesta.  
 Abrir archivo R-NOM-TAB1.  
 Escribir "Generalidades de la tabla:" R-NOM-TAB1.  
     "Columna...Nombre...Longitud...Tipo".  
 Leer primer registro del archivo R-NOM-TAB1.
- ESCRIBIR (1,NUM-COL)
  - Escribir INDICE LLAVE NOMBRE  
     LONGITUD TIPO
- Cerrar archivo R-NOM-TAB1.

OPERACIONES  
CON-TABLAS

Desplegar menú de tablas.

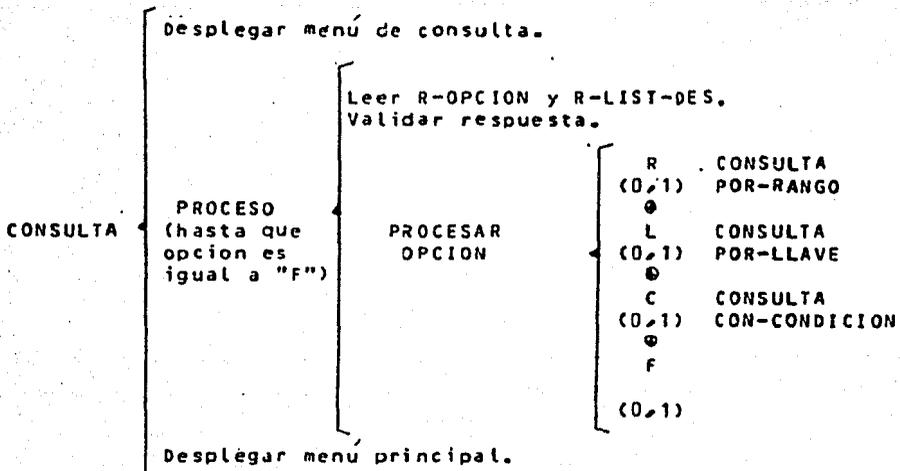
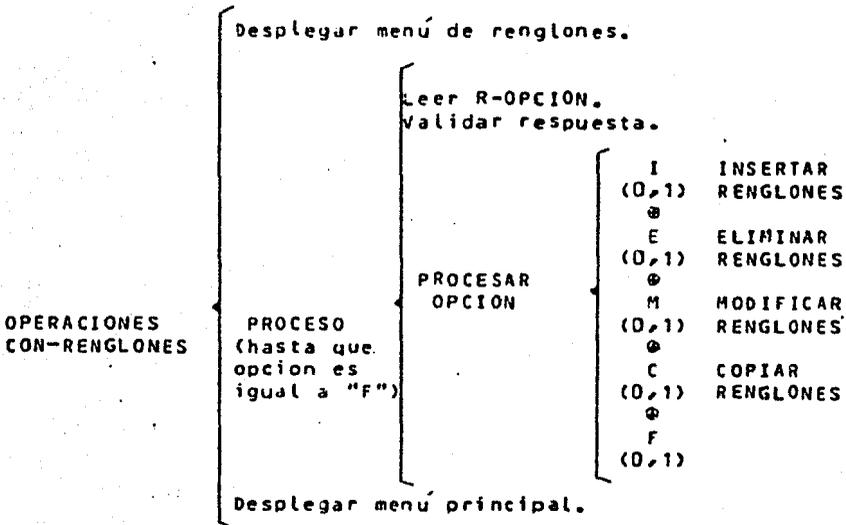
Leer R-OPCION y R-DESPL.  
Validar respuesta.

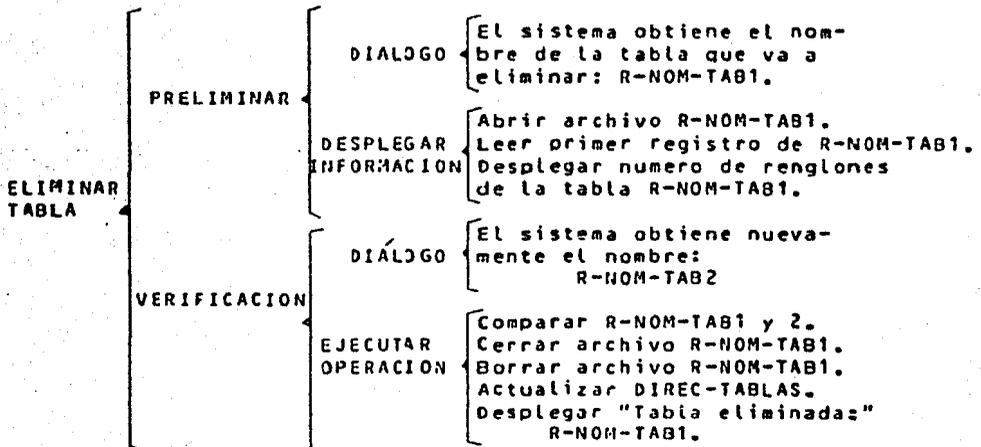
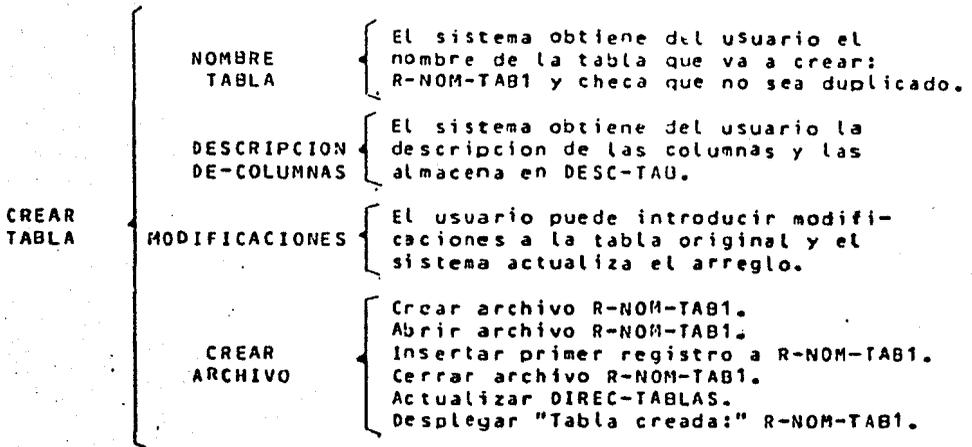
PROCESAR  
OPCION

PROCESO  
(hasta que  
opcion es  
igual a "F")

- C            CREAR-TABLA  
(0,1)
- ⊕
- E            ELIMINAR-TABLA  
(0,1)
- ⊕
- P            PROYECTAR-TABLA  
(0,1)
- ⊕
- S            CLASIFICAR-TABLA  
(0,1)
- ⊕
- M            MULTIPLICAR  
TABLAS  
(0,1)
- ⊕
- U            UNIR-TABLAS  
(0,1)
- ⊕
- D            DIFERENCIA  
DE-TABLAS  
(0,1)
- ⊕
- F  
(0,1)

Desplegar menú principal.

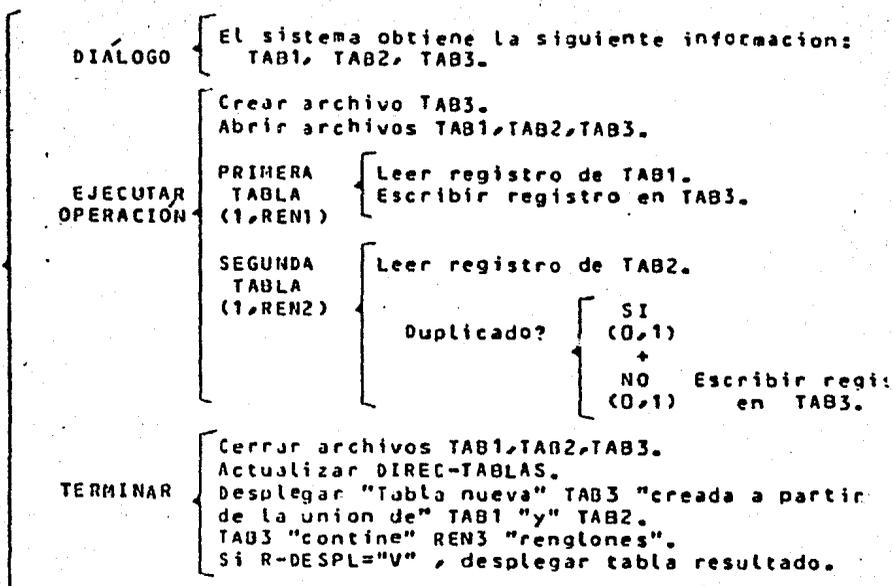


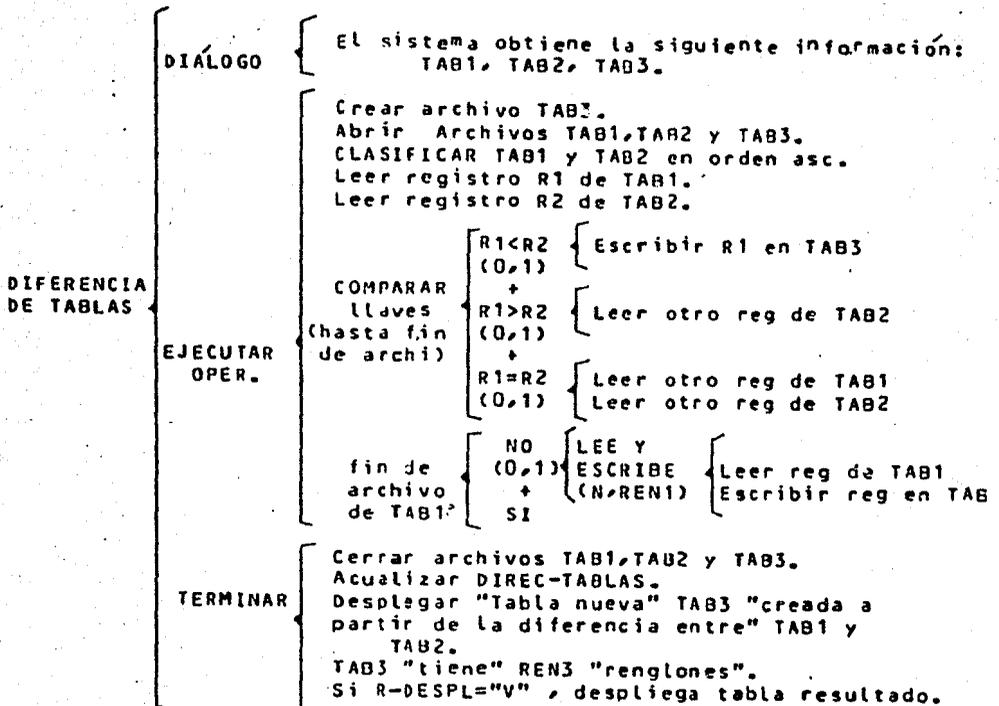




CLASIFICAR TABLA	DIALOGO	El sistema obtiene la siguiente informacion: R-NOM-TAB1, R-NOM-CLM(1, NUM-CLM), y R-ASC-DES.
	EJECUTAR OPERACION	Abrir archivo R-NOM-TAB1. Para ordenar el archivo en orden R-ASC-DES se usa el metodo de heap-sort [Knuth73] tomando como llaves las columnas R-NOM-CLM(1, NUM-CLM).
	TERMINAR	Cerrar archivo R-NOM-TAB1. Escribir "Tabla" R-NOM-TAB1 "clasificada" por R-NOM-CLM(1, NUM-CLM) "en forma" R-ASC-DES. Si R-DESPL="V" desplegar tabla resultante.

MULTIPL. TABLAS	DIALOGO	El sistema obtiene la siguiente informacion: TAB1, TAB2, TAB3, R-CLM-EJE.		
	EJECUTAR OPERACION	<p>Crear archivo TAB3. Abrir archivos TAB1, TAB2, TAB3. Clasificar TAB1 y TAB2 en orden ascendente tomando como llave R-CLM-EJE. Leer registro R1 de TAB1. Leer registro R2 de TAB2.</p> <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 10px;">COMPARAR colums-eje (hasta fin de archivo)</td> <td style="border-left: 1px solid black; padding-left: 10px;">           = crear reg y            (0,1) escribir en TAB3.            ⊕            R1&lt;R2 Leer otro reg de TAB1.            (0,1)            ⊕            R1&gt;R2 Leer otro reg de TAB2.         </td> </tr> </table>	COMPARAR colums-eje (hasta fin de archivo)	= crear reg y (0,1) escribir en TAB3. ⊕ R1<R2 Leer otro reg de TAB1. (0,1) ⊕ R1>R2 Leer otro reg de TAB2.
	COMPARAR colums-eje (hasta fin de archivo)	= crear reg y (0,1) escribir en TAB3. ⊕ R1<R2 Leer otro reg de TAB1. (0,1) ⊕ R1>R2 Leer otro reg de TAB2.		
TERMINAR	Cerrar archivos TAB1, TAB2, TAB3. Actualizar DIREC-TABLAS. Escribir "Tabla nueva" TAB3 "creada multipl." TAB1 "y" TAB2 "sobre" R-CLM-EJE. TAB3 "contiene" REN3 " renglones". Si R-DESPL="V", desplegar tabla resultante.			

UNIR  
TABLAS



INSERTAR RENGLONES	DIALOGO	El sistema obtiene el nombre de la tabla R-NOM-TAB1. Abrir archivo R-NOM-TAB1. Leer registro de R-NOM-TAB1. Desplegar nombre, longitud, tipo y si es llave de las columnas de R-NOM-TAB1.
	EJECUTAR OPERACION (hasta que renglon="")	Desplegar " Renglon" I "?". Leer R-REGLON. Validar respuesta. Checar por duplicado. Escribir R-REGLON en R-NOM-TAB1.
	TERMINA	Cerrar archivo R-NOM-TAB1. Actualizar DIREC-TABLAS (parte de renglones). Desplegar "Se insertaron" NUM-REN "a la tabla" R-NOM-TAB1.

ELIMIN. RENGL.	DIALOGO	El sistema obtiene el nombre de la tabla TAB1. Abrir archivo TAB1. Leer primer registro de TAB1. Desplegar nombre de columnas-llave.			
	OBTENER INFORMACION (hasta que R-LLAVE="")	Desplegar " Renglon" I "?". Leer R-LLAVE. Validar respuesta. Marcar el registro R-LLAVE como eliminado en el campo CLAVE. Crear archivo TAB1-BIS Abrir archivo.			
	ELIMINAR REGISTROS	<table border="0" style="margin-left: 20px;"> <tr> <td style="vertical-align: top;">LEER-ESC (1, NUM-REN)</td> <td style="vertical-align: top;">           Leer registro de TAB1.            MARCA            ELIMINADO         </td> <td style="vertical-align: middle;">           { SI            NO         </td> <td style="vertical-align: middle;">           Escribir registro en TAB1-BIS.         </td> </tr> </table> Cerrar archivo TAB1 y BIS. Borrar archivo TAB1. Cambiar nombre a TAB1-BIS por TAB1. Actualizar DIREC-TABLAS (parte de renglones). Desplegar "Se eliminaron" I "renglones" de la tabla R-NOM-TAB1.	LEER-ESC (1, NUM-REN)	Leer registro de TAB1. MARCA ELIMINADO	{ SI NO
LEER-ESC (1, NUM-REN)	Leer registro de TAB1. MARCA ELIMINADO	{ SI NO	Escribir registro en TAB1-BIS.		

MODIFICAR  
RENGLONES  
(hasta que  
opcion="F")

Desplegar menú para modificar renglones.  
Leer R-OPCION y R-DESPL.  
Validar respuesta.

R-OPCION? { L      POR-LLAVE  
                  (O,1)  
                  +  
                  C      POR-COLUMNA  
                  (O,1)  
                  +  
                  F  
                  (O,1)

COPIAR  
RENGLONES

DIALOGO

El sistema obtiene la siguiente informacion:  
R-NOM-TAB1 - tabla fuente  
R-NOM-TAB2 - tabla receptora  
Validar que ambas tablas tengan  
la misma estructura.  
Abrir archivos R-NOM-TAB1 y 2.  
Desplegar las columnas llave de R-NOM-TAB1.

COPIA  
(hasta que  
llave=" ")

Desplegar "Renglon" I "?".  
Leer R-LLAVE.  
Validar respuesta.  
Leer registro R-LLAVE de R-NOM-TAB1.  
Escribir registro R-LLAVE en R-NOM-TAB2.

TERMINAR

Cerrar archivos R-NOM-TAB1 y 2.  
Actualizar DIREC-TABLAS (parte de renglones).  
Desplegar "Se copiaron" I "renglones de la  
tabla" R-NOM-TAB1 "a" R-NOM-TAB2.

### 3.3. Estimados de Tiempo y Espacio.

El proposito de la presente sección es presentar estimados de tiempo de ejecución y espacio requerido tanto en memoria como en disco magnético. Se incluyen fórmulas algebraicas generales para algunas operaciones en la base de datos, y despues se aplican estas fórmulas utilizando tablas-ejemplo y constantes típicas de computadoras personales en el mercado. Se podrá observar que los tiempos resultantes son sumamente razonables.

En todos los casos se asume que el tiempo total de una operación está dominado por el acceso y transmisión de datos entre la memoria y el disco magnético. El tiempo de UCP normalmente se desprecia; así como el tiempo empleado por funciones que se ejecutan una sola vez, tales como despliegue de menús. El tiempo de apertura y cierre de archivos sí se considera.

#### 3.3.1. Constantes Típicas.

Las siguientes constantes corresponden a la computadora personal de IBM. [IBM82]

- Capacidad del disco: 40 pistas  
8 sectores/pista  
512 bytes/sector

---

163,840 bytes/disco

- Tiempo de acceso (seek time): 25ms + 8 ms/pista

- Tiempo de acceso promedio (20 pistas):  $(25+20 \cdot 8) = 185\text{ms}$
- Velocidad de transferencia (VT):  $250 \text{ Kbits/seg} = 31 \text{ bytes/mseg}$
- Retraso por rotacion (RR):  $83 \text{ mseg.}$

### 3.3.2. Tiempo de Apertura y Cierre de Archivos.

Para abrir ó cerrar archivos se requiere un acceso al directorio del disco, leer un sector del directorio y un acceso al primer sector del archivo. El tiempo total es:

$$2(185+183) + 512/31 = 552\text{ms}$$

Este tiempo se denomina "TAC" en las formulas que siguen (Tiempo de Abrir ó Cerrar).

### 3.3.3. Nombre de Tablas Almacenadas.

Esta operacion involucra abrir y cerrar el directorio de tablas y leer el registro correspondiente a cada tabla. Los tiempos se pueden calcular como sigue:

abrir y cerrar	2TAC
retraso por rotacion	NUMTAB*RR
longitud de c/registro	k+4
tiempo de transferencia	$[(k+4)/VT]*NUMTAB$
tiempo total	$2TAC+NUMTAB(RR+(k+4)/VT)$

El espacio en disco es:  $NUMTAB (k+4) \text{ bytes}$

Para 40 tablas y k igual a 13 obtenemos:

Tiempo:

$$2 \cdot 552 + 40(83 + 14/31)\text{mseg} = 4442 \text{ mseg} = 4.5 \text{ seg}$$

Espacio:

$$40 \times 14 = 560 \text{ bytes}$$

### 3.3.4. Crear Tabla.

1) Abrir y cerrar el archivo correspondiente a la tabla, escribir su primer registro.

$$2TAC + RR + NUMCOL[(k+4)/VT]$$

Para una tabla de 10 columnas y k igual a 10, el tiempo es:

$$2 \times 552 + 83 + 10(14/31) = 1191 \text{ mseg} = 1 \text{ seg}$$

2) Actualizar el directorio de tablas. Involucra copiar el archivo:

$$4TAC + 2NUMTAB [RR + (k+4)/VT] \text{ mseg}$$

Para 40 tablas y k igual a 10, el tiempo estimado es 8.8 seg.

### 3.3.5. Proyectar Tabla.

$$4TAC + RR + NUMCOL [(k+4)/VT] + REN [RR + (LREN/VT)]$$

es el tiempo que toma leer una tabla completa de REN renglones donde la longitud de cada renglon es LREN. Para valores típicos NUMCOL = 5, k = 10, REN = 60 y LREN = 40, tenemos:

$$2 \times 552 + 83 + 5(14/31) + 60 [83 + (40/31)] = 6247 \text{ mseg} \\ = 6 \text{ seg.}$$

### 3.3.6. Multiplicar Tablas.

Esta operación presenta un caso en el que el tiempo de procesamiento de UCP podría no ser despreciable ya que involucra el ordenamiento interno de las dos tablas. El tiempo total puede calcularse como sigue:

#### 1) Ordenamiento.

##### a. Leer ambas tablas a memoria

$$2TAC + 2RR + REN1(LREN1/VT) + REN2(LREN2/VT)$$

Suponiendo de nuevo 60 renglones de longitud 40 en cada tabla, se obtiene un tiempo de 1424 mseg.

##### b. Heapsort interno.

Heapsort involucra aproximadamente  $7 \cdot N \cdot \log N$  comparaciones entre las llaves [Knuth73] donde  $N$  es el número de elementos a ordenarse. Utilizaremos una constante de 10 en vez de 7, para tomar en cuenta operaciones adicionales a las comparaciones tales como intercambios de elementos. El tiempo que toma cada operación es proporcional a la longitud de la columna-pivote.

$$10(REN1 \cdot \log REN1) + C + LCOL + 10(REN2 \cdot \log REN2) + C + LCOL$$

Para nuestro ejemplo, considerando una longitud de columna de 10, y una constante de proporcionalidad  $C = 10$  microsegundos, se obtiene un tiempo de: 960 mseg.

##### c. Escribir ambas tablas toma un tiempo de 712 mseg.

2) Ya ordenadas las dos tablas se requerirá el tiempo siguiente para leerlas y escribir la tabla producto:

$$6TAC + 6RR + (1/VT)[REN1 \cdot LREN1 + REN2 \cdot LREN2 + \\ REN3(LREN1 + LREN2 - LCOL)]$$

Donde REN3 es el número de renglones de la tabla producto.

En nuestro caso, consideraremos  $REN3 = REN1/2 = 30$ .

Obteniéndose un tiempo de 3832 mseg.

El tiempo total para multiplicar dos tablas en este ejemplo es 6828 mseg.

### Conclusión.

Hemos presentado el diseño de un sistema de base de datos relacional orientado al usuario. Las principales características de este sistema fueron derivadas a partir de un marco teórico que proporciona una estructura sólida a las decisiones de diseño.

El marco teórico nos llevó a considerar un sistema interactivo para computadoras personales basado en el modelo relacional, y operado en forma sencilla a base de menus. El sistema incluye una serie de funciones simples que se pueden combinar en varias formas para lograr funciones mas complejas.

La implementación se describió utilizando diagramas de Warnier-Orr, a partir de los cuales la programación del sistema en cualquier lenguaje de computadora es relativamente directa.

## BIBLIOGRAFÍA

[Ahl] Ahl, D.H. "Ten Rules for Writing User-Oriented Programs" en Personal Computer Proceedings de la NCC 79.

[Bartl] Bartling, J.R. "Strong Trend to DBMS Confirmed", en Computer Decisions, Julio 1980.

[Boehm] Boehm, B.W. "Software and its Impacts: A Quantitative Assessment", en Datamation, Mayo 1973.

[Brook] Brooks, Jr. F.P. The Mythical Man-Month, Addison-Wesley, 1975.

[BW1] "Missing Computer Software", en Business Week, Sept. 1980.

[BW2] "A Rush of New Companies to Mass-Produce Software", en Business Week, Sept. 1980.

[Canda] "CSA Group Continues Work Toward DBMS Standard", en Canadian Datasystems, Oct. spe 1980.

[CompS] Computer Surveys, Vol. 8 No. 1 Marzo 1976, ACM.

[Date] Date, C.J. An Introduction to Database Systems, Addison-Wesley, 1977.

[Elmo] Elmore, R.W. "An Information Retrieval System", en BYTE Oct. spe 80.

[Gray] Gray, P. "Analyzing the Future Impacts of Personal Computers", en Personal Computer Proceedings de la NCC 79.

[Hayes] Hayes, J.P. "Technology: Changes in Personal Computers", en Personal Computer Proceedings de la NCC 79.

[Higg] Higgins, David A. Program Design and Construction, Prentice-Hall Series in Personal Computing, USA 1979.

[Holm] Holmes, M.G. "Who is Using Personal Computers?", en Personal Computer Proceedings de la NCC 79.

[IBM82] IBM personal Computer Technical Reference Manual, #6025005, p.2-112

[Kern] Kerns, R. "Personal Computers vs. Terminal for Interactive Computing", en Personal Computer Proceedings de la NCC 79.

[Knuth73] Knuth, Donald E. "The Art of Computer Programming" Vol.3 Addison-Wesley 1973.

[Ledg] Ledgard, H. et al. "The Natural Language of Interactive Systems", en CACH, Oct. spe 1980.

[Lema] Lemaitre, C. et al. "Un Sistema de Consulta en Espanol de una Base de Datos", Comunicacion Interna No.2, 1979, Facultad de Ciencias UNAM.

[Luba] Lubar, D. "Christmas Buyer Guide", en Creative

Computing, Dic. 1980.

[Mart73] Martin, J.T. The Computerized Society, Pelican Books, England 1973.

[Mart77] Martin, J.T. Computer Data-Base Organization, Prentice Hall, N.J. 1977.

[Mead] Meadow. Man-Machine Communication, Wiley-Interscience.

[Meld] Meldman, M.J. et al. RISS: A Relational Data Base Management System for Minicomputers, Van Nostrand, 1978.

[Nilles] Nilles, J.M. "Personal Computers in the Future: An Overview", en Personal Computer Proceedings de la NCC 79.

[Rama] Ramamoorthy, C.V. "A Design Methodology for User Oriented Computer Systems", en Tutorial: Software Methodology, IEEE 78.

[Sigpl] Sigplan Notices, Vol.16 No.3, Marzo 1981, ACM.

[Slon] Slonim, J. "Mediator: An Integrated Approach to Information Retrieval", en SIGIR 78, International Conference on Information Storage and Retrieval Proceedings, ACM.

[Tsich] Tsichritzis, D.C y Lochovsky, F.H. Data Base Management Systems, Academic Press, New York 1977.

[Ullman] Ullman, J.D. Principles of Database Systems, Computer Science Press, 1980.

[Vandij] Vandijck "An Overview of Current Relational Database Query Languages", en Data Base the Next Five Years, Infotech State of the Art Conference, Dic. 1977.

[Wein] Weinberg, G.M. The Psychology of Computer Programming, Van Nostrand 1971.

[Wied] Wiederhold, G. Database Design, Mc Graw Hill, 1977.

[Willi] Williams, G. "Is this Really Necessary? A First Look at Design Techniques", en BYTE Marzo 1981.

[Willih] Williamson, H. et al. "The User Interface Design Process", en Second Annual Conference on Office Communications Products, Houston 1981.