



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**Facultad de Ingeniería**

**Un estudio sobre la optimización  
en la organización de las bases  
de datos**

**T E S I S**

Que para obtener el título de  
**INGENIERO**

presentan

Báez Reséndiz Nabor Fabián  
Mandujano Ortiz José Antonio  
Martínez Sánchez Juan Bosco  
Zarza Cordero Ignacio

**Director: Ing. José Arturo Origel Coutiño**

**México, D. F.**

**1984**



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.

# I N D I C E

PROLOGO .-----	1
INTRODUCCION .-----	4
Definición .-----	4
Objetivos .-----	6
Aplicaciones .-----	11
CAPITULO I. Características de la Base de datos. -----	16
Costo Mínimo. -----	18
Redundancia Mínima .-----	18
Integridad .-----	20
Seguridad .-----	20
Accesibilidad .-----	23
Migración de Datos .-----	24
Independencia de Datos .-----	24
CAPITULO II. Organización de la base de datos. -----	37
Introducción .-----	37
Organización Lógica. -----	40
Definición de términos .-----	40
Organización Lógica Global de los Datos. -----	41
Tres Tipos de Organización. -----	46
Organización Física .-----	48
Nivel Interno .-----	48
Redundancia .-----	50
Tipo de Procesamiento .-----	50

Razón de actividad del archivo .-----	51
Archivos Dinámicos .-----	52
Protección de los datos .-----	53
Medios de almacenamiento .-----	53
Formas de Acceso .-----	54
Registro Físico .-----	55
Celdas .-----	56
Paginación .-----	59
 CAPITULO III. Metodología .-----	 60
Introducción .-----	60
Esquemas .-----	61
Relaciones y Referencias Cruzadas .-----	61
Subesquemas .-----	63
Correspondencia de datos Simple y Compleja .-----	66
Como dibujar un Esquema .-----	66
Archivos planos .-----	68
Bases de datos Relacionales .-----	69
Normalización .-----	70
Tuples .-----	71
Claves .-----	71
Estructura de Datos .-----	73
Estructuras Estáticas .-----	73
Arreglos .-----	74
Registros .-----	74
Archivos .-----	77
Estructuras Dinámicas .-----	79

Apuntador .-----	80
Listas Ligadas .-----	81
Listas Circulares .-----	85
Listas Doblemente Ligadas .-----	86
Arboles .-----	86
Arboles Binarios .-----	90
Recorrido de Arboles Binarios .----	93
Arboles de Multicamino .-----	95
Técnicas de Direccionamiento .-----	98
Introducción .-----	98
Exploración o Barrido del Archivo .----	99
Búsqueda por bloque .-----	100
Búsqueda Binaria .-----	101
Archivos Secuenciales Indizados .-----	103
Archivos no Secuenciales Indizados .--	105
Direccionamiento Clave-igual-Direc ción .-----	105
Algoritmos para Conversión de Clave--	106
Desmenuzamiento (Hashing) .-----	107
Combinación de Técnicas .-----	110
Inserciones y Eliminaciones .-----	110
Consideraciones sobre el Hardware .-----	110
Desmenuzamiento (Hashing) .-----	112
Técnicas de Indización .-----	122
Manejo de Stacks (Pilas) .-----	130
Introducción .-----	130
Que es un mecanismo de Pila .-----	130

Implementación de Pilas en las Com putadoras. -----	132
CAPITULO IV. Trabajo Práctico. -----	134
Introducción. -----	134
Diseño de la Base de Datos. -----	136
Estudio Practicado de Técnicas de Acce- so. -----	141
Secuencial. -----	142
Secuencial Indexado. -----	153
No Secuencial Indexado. -----	154
Desmenuzamiento (Hashing). -----	157
Resultado de las Pruebas. -----	159
Tablas de Tiempos. -----	161
Procesador. -----	161
Entrada/Salida. -----	162
Gráfica de Tiempos Totales. -----	180
Descripción del sistema de acceso. -----	184
Descripción del algoritmo de acceso secu <u>en</u> cial en archivos secuenciales. -----	186
Descripción del algoritmo de acceso secu <u>en</u> cial en archivos INDEX SECUENCIAL. -----	193
Descripción del algoritmo de acceso no se- cuencial indexado. -----	194
Descripción del algoritmo de acceso Hashing	200
CAPITULO V.- Conclusiones. -----	201
BIBLIOGRAFIA. -----	205

## P R O L O G O

Debido al gran crecimiento económico y de población, ha aumentado en forma considerable la cantidad de datos que se requieren en los sistemas de información de las empresas; de tal manera que es difícil imaginar de que forma se pueden satisfacer -- las necesidades de información, sin el uso de computadoras. Esto se puede ver fácilmente en el caso de un sistema de reservaciones de una línea aérea; donde ante la necesidad de reservar una plaza para un cierto vuelo, se requiere conocer el número de lugares disponibles, en el momento en el cual se hace la reservación, tomando en cuenta que la compañía puede tener muchas oficinas, muchos vuelos y que la respuesta tiene que ser inmediata; además de que -- los datos pueden ser utilizados para otras aplicaciones como son la programación de vuelos.

Ante esta necesidad, los sistemas de información han ido evolucionando y adquiriendo ciertas características hasta llegar al concepto de base de datos que definimos en la introducción; para posteriormente en el capítulo primero hacer un análisis de las características que debe cumplir dicha base de datos ya que no es solo un receptáculo en el --- cual se almacenan datos, sino que es toda una metodología desarrollada para satisfacer las necesida--

des de información en el mínimo tiempo, a la persona indicada, en el lugar correcto y en el menor costo posible.

Para suministrar la información en forma eficiente, es importante que esta sirva al fin para la cual fue creada, por lo que hay que conocer las necesidades de la empresa que los requiere, para poder definir la organización que va a tener la información en la base de datos. Este análisis se realiza de acuerdo a los conceptos que se ven en el capítulo II, para cumplir con las características que se vieron en el primer capítulo, tomando en cuenta tanto las necesidades de los usuarios, como las propiedades de los medios de almacenamiento que se van a utilizar, por lo que el capítulo II tiene dos grandes divisiones. La primera que llamaremos "Organización Lógica", donde se estructuran los datos tomando en cuenta las necesidades de información actuales y a futuro, para que los cambios que puedan haber en dichas necesidades, afecten al mínimo a los usuarios que la estén utilizando. Una vez estructurados los datos se hace necesario un análisis para ver de que manera serán almacenados los datos en la computadora a lo que llamamos "Organización Física" que es la segunda división del capítulo II.

Posteriormente en el capítulo III se estudia la metodología que permite la implementación de los



datos tanto lógica como física. En este capítulo se hace un estudio de las técnicas de direccionamiento, las cuales nos permiten un almacenamiento de los datos en los dispositivos físicos de la computadora, también se realiza un estudio de las técnicas de acceso para la obtención de los datos tanto en línea como en lote.

De lo anterior pasamos al capítulo IV que es la implementación del ejemplo práctico, donde tomando en cuenta lo visto en los capítulos anteriores y la finalidad de este trabajo, realizamos un estudio de tiempos para las principales técnicas de acceso, ya que es donde se refleja con mayor fuerza la optimización de la base de datos.

Finalmente y basados en los resultados de las pruebas prácticas realizadas en el capítulo anterior obtenemos una serie de conclusiones que quedan a juicio de los lectores para su utilización.

## INTRODUCCION

### DEFINICION.

La idea de Base de Datos ha estado por todos - lados desde el principio de los 60's. Al igual que - otro Tópico llamado Manejo. El propósito es el de - proveer a una determinada Organización que requiera del manejo de un gran volumen de información, con - la habilidad de un mejor manejo de un conjunto de - datos, basado en las características de cada uno de los entes que la integran, así como los nuevos ele - mentos que la experiencia va introduciendo a la mis - ma, si nos ponemos a pensar en cuanto le cuesta a - una Organización el archivar, recabar, actualizar y eliminar datos; cuál sería el costo si esos datos - se perdieran o fueran saboteados; en la importancia que tienen en la toma de decisiones de la Organiza - ción y la adecuada presentación o actualización de los datos, veriamos la necesidad de conjuntar tales datos en un receptáculo, en donde la información es - té disponible para la programación de funciones efi - cientemente, esto es para la creación de un sistema que nos permita satisfacer la necesidad de informa - ción.

Este gran almacén puede estar concentrado en - una localidad determinada o distribuida en varias, todas ellas posiblemente interconectadas mediante - un sistema de Telecomunicaciones el cual llamaremos

## BASE DE DATOS.

La Base de Datos en resumen puede definirse como:

" UNA COLECCION DE DATOS INTERRELACIONADOS EN CON--  
JUNTO SIN REDUNDANCIAS PERJUDICIALES O INNECESARIAS,  
CUYA FINALIDAD ES LA DE SERVIR A UNA APLICACION O -  
MAS DE LA MEJOR MANERA POSIBLE, LOS DATOS SE ALMACE--  
NAN DE MODO QUE RESULTEN INDEPENDIENTES DE LAS APLI--  
CACIONES QUE SE VAN A DAR Y DEBEN TENER LA CAPACI--  
DAD PARA INCLUIR DATOS NUEVOS, MODIFICAR, EXTRAER -  
Y/O ELIMINAR LOS DATOS ALMACENADOS".

Toda base de datos compleja tiene que ser esta--  
blecida por etapas y en realidad, las bases actua--  
les solo sirven a un conjunto limitado de aplicacio--  
nes. Generalmente una computadora dispone de varias  
bases de datos y se van fundiendo hasta alcanzar la  
integración total que conduce a un aumento en el --  
rendimiento o la utilidad del sistema total.

## OBJETIVOS

Partiendo de la definición, el banco de datos debe ser diseñado de acuerdo a la serie de aplicaciones que se van a dar para el sistema de información, a través de programas de aplicaciones.

La base de datos no implica únicamente que una Organización deba adquirir un sistema de manejo de base de datos de un vendedor y entregarlo al grupo de usuarios dentro de la Organización. El elemento clave de dicho sistema son los objetivos, los métodos y procedimientos para llegar a ellos, sin estos métodos y procedimientos no puede funcionar eficazmente para el logro de sus objetivos.

Las bases de datos pueden ser diseñados para procesamiento por lotes, en tiempo real o en línea, muchas bases de datos sirven a una combinación de estos métodos de procesamiento y en ello se emplean por ejemplo: terminales de tiempo real al mismo tiempo que el procesamiento por lote.

El banco de datos diseñado para ser usado por operadoras de terminal debe asegurar un tiempo de respuesta adecuado para el diálogo entre el hombre y la terminal. Además, el sistema de base de datos debe tener capacidad para manejar un adecuado caudal de transacciones. En los sistemas en el que el

volumen de tráfico es reducido, el caudal de transacciones no tiene porque imponer muchas restricciones al diseño del banco de datos. En cambio en los sistemas de alto volumen de tráfico, por ejemplo -- los de reservaciones de plazas en aerolíneas, el -- caudal de transacciones tiene gran influencia sobre la organización en el almacenamiento de los datos.

En los sistemas diseñados solo para el procesamiento por lotes, el tiempo de respuesta no tiene un gran significado y el método de organización se elegirá teniendo a la vista la mayor eficiencia con este tipo de procesamiento.

El tiempo de respuesta adecuado en el caso de un sistema de terminales depende de la naturaleza del dialogo entre hombre y terminal. Para ciertas categorías de diálogo se requiere un tiempo de respuesta del orden de los 2 segundos (dos segundos desde el momento en que el operador completa su mensaje de entrada en la terminal y el instante en que aparece en ésta el primer carácter de respuesta).

En el caso de averiguaciones simples puede tolerarse en general tiempos de respuesta mucho mayores.

El tiempo de respuesta depende en mucho de la organización que tenga el banco de datos.

El valor de la colección de datos es escaso, -

no obstante, a menos que se utilice para comprender el mundo real y tomar las acciones adecuadas para mejorarlo. Simples volúmenes de hechos y cifras des organizados no sirven para mucho en la Industria, en el gobierno o en cualquier otra actividad. El -- problema central del procesamiento de datos es ex-- traer de una montaña de hechos, lo que es de cierto valor para el usuario humano, para lo cual se debe tener en cuenta la escala siguiente:

- Completamente impredecible
- En gran parte predecible
- No repetitivo, parcialmente predecible
- Bastante repetitivo, parcialmente indefinido.
- Muy repetitivo, definido con precisión.

En el fondo de la escala la información se requiere diariamente y a menudo, como ocurre en las - agencias y sucursales de banco, en este caso se co- noce perfectamente la naturaleza de la información que se pide, antes de hacer el pedido.

A medida que se asciende la escala los requerimientos de información son cada vez menos repetiti- vos y existe cierta variación en los datos que se - piden, de modo que la computadora se ve en la nece- sidad de "componer" la respuesta adecuada.

En el tope superior de la escala, la informa--

ción buscada debe ser extremadamente valiosa, y que cuanto menos frecuentes sean los pedidos de información, tanto más valiosa debe ser la información para justificar el sistema.

Para resolver este tipo de pedido se debe tener una organización muy bien elaborada, con los programas adecuados para su localización.

Un ejemplo que ilustra perfectamente el problema de la localización de los datos, son los volúmenes de una biblioteca, el catálogo está organizado de tal manera que se puede localizar un libro determinado por su título o por el autor, pero si alguien quiere saber cuántos y que libros incluyen la palabra Gerencia en su título, el catálogo no da ya la respuesta directa, por lo que será necesario explorar el catálogo para revisar los libros y como la palabra gerencia no siempre estará como primera palabra de un título se tendrá entonces, que revisar todas las entradas del catálogo, ésta búsqueda se podría abreviar si se contara con otro tipo de catálogo y esto es muy similar para los archivos de computadora, si se contara con índices especiales podríamos decir que en el diseño del sistema, se han anticipado esos tipos de preguntas aunque no su naturaleza exacta, ahora nos encontramos en la mitad de la escala.

El problema estriba básicamente, en que la información adopta una variedad casi infinita de formas, de tal manera que sería imposible construir -- una computadora para bibliotecas capaz de satisfacer cada averiguación, que los lectores puedan requerir. De la misma manera sería imposible construir o diseñar un sistema capaz de responder todas las preguntas de todos los gerentes de una organización, ya que un buen gerente se caracteriza, por las variaciones en sus preguntas, con el fin de obtener la respuesta más útil para sus fines. Sin embargo la manera de extraer información, de los sistemas de información futuros consistiría probablemente, en interrogarlos por medio de terminales suficientemente rápidas, presentándoles una variedad de preguntas hasta obtener la respuesta con la información necesaria, de ahí la necesidad de contar con un sistema que posea la capacidad adecuada para realizar una búsqueda rápida con múltiples atributos, a veces en múltiples registros o relaciones.



## APLICACIONES

Para quienes no comprendan su estructura detallada, La Base de Datos no es más que un gran depósito de elementos de datos que los usuarios pescan. Situándonos en la realidad, la mayoría de los datos en una base de datos típica, no son tan accesibles, como la base de datos ideal lo sugiere. Un ejemplo es que un usuario no puede extraer información de la nómina al azar, ya que este archivo por sus características de procesamiento e impresión de cheques, está organizado secuencialmente. Se pueden organizar los datos de tal manera que se puedan obtener éstos al azar, pero se necesitaría tener una organización física de los archivos completamente diferente y mucho más costosa, lo cual no sería suficientemente adecuada para la ejecución eficiente de este tipo de tareas, como el de preparar la nómina.

El analista de sistemas debe decidir por lo tanto que estructura de datos se utilizarán para:

- 1.- La eficiente ejecución de tareas repetitivas (es decir de procesamiento por lotes).
- 2.- Las averiguaciones predefinidas y directas (ej. muestrame el estado de la cuenta del señor X).
- 3.- La pesca al azar con el fin de atender las averiguaciones imprevistas y mal estructu-

radas (que proporción de pedidos para el modelo YXZ provienen de firmas grandes?).

Si los datos han de usarse para los casos 1 y 2, la misma estructura física será usada para ambos, pero si se tiene que contemplar también el caso 3, es preferible debido a restricciones de software -- agruparlos en una base de datos aparte.

Para distinguir éstos dos tipos de sistemas se les llamará para el caso 1 y 2, "SISTEMA DE OPERACIONES", debido a que el procesamiento de los datos es rutinario y preestablecido, para el caso 3 se -- le denominará "SISTEMA DE INFORMACION", y la naturaleza de las averiguaciones que atenderá no será definido de antemano como para el anterior.

En un sistema de información, debido a que los tipos de interrogantes no han sido previstos, se necesitan a menudo prolongadas búsquedas en los archivos para obtener una respuesta.

En los sistemas de operaciones esas búsquedas se soslayan por lo general porque la información está almacenada de la manera como se le necesita.

Daremos a continuación algunos ejemplos de los dos tipos de sistemas.

#### Sistemas de Operaciones:

Un sistema bancario con terminales para --

los cajeros.

Un sistema para planta manufacturera.

Un sistema para reserva de plazas en una aerolínea.

Un sistema de entrada de ordenes y averiguaciones en un departamento de ventas.

Un sistema de tarjetas de crédito.

Un sistema de control de tráfico aéreo.

Un sistema policial.

#### Sistemas de Información.

Un sistema para búsquedas bibliográficas en una biblioteca.

Un sistema de información para la gerencia de comercialización.

Un sistema de búsquedas en una base de datos judicial.

Un sistema de búsqueda en una base de datos de personal.

En algunos casos se tiene que un sistema de operaciones tiene los mismos campos de datos que un sistema de información, pero existen razones para que las bases de datos estén organizadas de manera diferente como ya lo hemos visto, pero se tienen que tomar en cuenta también los siguientes puntos:

- 1.- Las estructuras de datos utilizadas en los sistemas de información son tales, que resulta difícil y costoso mantener los datos al día. Los sistemas de operaciones, en cambio, emplean estructuras más sencillas, diseñadas para su fácil y rápida actualización.
- 2.- Muy importante, es muy difícil insertar -- nuevos datos y eliminar los viejos de la base de datos de un sistema de información, excepto mediante una prolongada operación fuera de línea. Las bases de datos de los sistemas de operaciones están organizados de tal modo que es fácil insertar o eliminar datos.
- 3.- Los sistemas de operaciones deben incluir las últimas transacciones. En cambio, carece de trascendencia el hecho de que los datos almacenados en una base de datos para el sistema de información, tengan una antigüedad de 24 horas o más.
- 4.- Los sistemas de operaciones tienen capacidad para manipular un gran volumen de ---- transacciones, de modo que es indispensable contar con una estructura de archivos de acceso rápido. Los sistemas de informa-

ción que contienen los mismos datos atienden normalmente un número bastante reducido de averiguaciones.

- 5.- Los sistemas de información pueden tener - informaciones resumidas o abreviadas sin - la totalidad de los detalles que son indis - pensables en el caso de los sistemas de -- operaciones.

Estas son las razones por las cuales un sistema de operaciones y un sistema de información que operan prácticamente con el mismo tipo de datos, al macenan esos datos en bases diferentes. Los siste-- mas de operaciones se actualizan en tiempo real. El sistema de información se actualiza fuera de línea, posiblemente por la noche a partir de los archivos preparados por el sistema de operaciones.

## C A P I T U L O I

CARACTERISTICAS DE LAS BASES DE DATOS

Para crear una base de datos hay que almacenar para cada entidad los datos de interes, no solo los que se necesitan para una determinada aplicación.

La idea final es el poder diseñar una base de datos no unicamente para poder usarla en trabajos - previstos, sino también el poder plantear al sistema interrogantes no previstos.

No obstante lo anterior, la implementación de la base de datos basada en ese concepto demuestra - ser una tarea larga y compleja; además, con el hardware actualmente disponible resulta muy costoso diseñarla de manera que permita ser inspeccionada con una rapidez adecuada como para proporcionar respuestas en tiempo real a los interrogantes no previstos de antemano.

Una base de datos debe tener entonces ciertas características básicas y otras más dependientes de la aplicación que se le vaya a dar.

Sin embargo, como esas características están - ligadas al software de la máquina que usemos, ninguna de ellas es aplicable tal y como idealmente debería ser, esto es debido a que ningún paquete de --- software satisface todavía las características de -

la base de datos ideal, por lo que el analista de sistemas se ve obligado a procurar la armonización de cualidades deseables pero a menudo contrarias.

Una de las características más importantes de la mayoría de las bases de datos es la de mantenerse en plena crisis de cambio y crecimiento. La base de datos debe prestarse a una fácil reestructuración siempre que haya que agregarle nuevos tipos de datos o utilizarla para las demás aplicaciones. Esta reestructuración no debe originar la necesidad de volver a escribir los programas de aplicación y en general, no debe ser fuente de trastornos. La facilidad con que pueda modificarse la base de datos tendrá siempre un efecto directo sobre la capacidad para desarrollar nuevas aplicaciones del procesamiento de datos dentro del organismo que la explota.

Podemos empezar a mencionar algunas de las características básicas de las bases de datos. Esta cuestión ha sido estudiada en detalle por muchos organismos, por empresas que utilizan computadoras, por comisiones del gobierno y por grupos para tareas confidenciales de los fabricantes de computadoras, de todas esas características hemos tomado para nuestro estudio las que consideramos más importantes:

## Costo Mínimo

Con el fin de mantener bajo el costo hay que elegir técnicas que minimicen las necesidades totales de almacenamiento. Aplicando éstas técnicas, la representación física de los datos en el almacén, será muy distinta de la que usa el programador. La conversión entre ambas representaciones se hace por Software.

Se debe tomar en cuenta que el costo por bit de almacenamiento está disminuyendo rápidamente, -- gracias al progreso tecnológico, y mientras no ocurra lo mismo con el costo de la programación, existirá una necesidad creciente de mantener sencilla -- la programación de aplicación y las organizaciones lógicas de los datos deben de estar diseñadas con -- este objetivo.

## Redundancia Mínima.

En un almacenamiento de tipo tradicional mu---chos datos se encuentran almacenados en varios volu---menes con distintas finalidades, y también con dis---tintas fechas de actualización, antes de la apari---ción de las técnicas de bases de datos, existía un altísimo nivel de redundancia en los sistemas de -- procesamiento de datos. En la mayoría de las biblio---tecas de cintas se tienen muchos datos redundantes.



Debido a que hemos definido la Base de Datos como "Un conjunto de datos sin redundancias perjudiciales o innecesarias", se tratará de eliminar esta redundancia utilizando ciertas técnicas diseñadas para este efecto, a pesar de ello a medida que crecientes volúmenes de información se combinan para formar bases de datos integradas, se crea una mayor posibilidad de redundancia, aunque también en algunos casos de bases de datos se admite cierta redundancia con el objeto de reducir los tiempos de acceso o simplificar los métodos de direccionamiento, a este tipo de redundancia le llamaremos redundancia controlada.

Como ya lo hemos hecho notar, la redundancia es honerosa porque ocupa más espacio de almacen que el necesario y requiere múltiples operaciones de actualización. Debido a que diferentes copias de la misma información suelen hallarse en diferentes etapas de actualización, la redundancia da a menudo origen a respuestas incoherentes, aunque hay que tomar en cuenta que algunos registros es conveniente duplicarlos para facilitar la reconstrucción en caso de daño accidental. De modo que en lugar de hablar de no redundancia, es mejor hablar de redundancia controlada o mínima como características de diseño.

## Integridad.

A menudo las bases de datos contienen información que es utilizada por muchos usuarios, entonces es necesario que estos datos no puedan ser destruidos o cambiados así como también las relaciones entre los distintos items. El almacenamiento de los datos así como los procedimientos de actualización deben asegurar que el sistema pueda recuperarse de contingencias tales como fallas de hardware y otros accidentes que puedan suceder.

Deben incluirse también procedimientos de chequeo que aseguren que los valores de los datos se ajusten a ciertas reglas preescritas de antemano.

En resumen si el sistema funciona adecuadamente deben existir mecanismos adecuados para asegurar que la base de datos en su aspecto físico está de acuerdo a su definición.

## Seguridad

La información de la base de datos es a menudo de gran valor para la Organización, de ahí que se deben establecer ciertos mecanismos de privacidad y protección para garantizar la seguridad de la base de datos.

Cuanto más vital es la información almacenada, tanto más importante es protegerla contra las fa---

llas de hardware y del software, a pesar de que los nuevos dispositivos contemplan mecanismos de protección de la información contra fallas imprevistas y de que los nuevos sistemas operativos contemplan ya sistemas de protección contra los imprevistos, también se debe proteger a la base de datos contra las catástrofes y contra todo tipo de personas, ya sean vándalos o incompetentes que pretendan darle un uso indebido.

La seguridad de los datos se refiere a la protección de estos contra el acceso accidental o intencional por parte de personas no autorizadas y -- contra su indebida destrucción o alteración.

La reserva se refiere al derecho de los individuos y organismos para determinar por sí mismo cuándo, cómo y en qué medida se permitirá la transmisión a terceros de la información que les concierne.

La reserva es una cuestión que traspasa los límites del centro de computación por lo que no detallaremos más sobre esto.

La seguridad es un tema muy complejo debido a sus variados aspectos. El analista de sistemas que se hace responsable de la seguridad debe de estar familiarizado con todas las particularidades del sistema porque éste puede ser atacado con fines ilícitos desde muchos ángulos.

Existen ciertos requisitos que se consideran básicos para la seguridad de la base de datos, y -- son los siguientes:

- a).- La base de datos debe de estar protegida contra el fuego, el robo y otras formas de destrucción.
- b).- Los datos deben de ser reconstruibles, -- porque por muchas precauciones que se tomen, siempre ocurren accidentes.
- c).- Los datos deben de ser sometidos a procesos de auditoría. La falta de auditoría -- en los sistemas de computación ha permitido la comisión de grandes delitos.
- d).- El sistema debe diseñarse a prueba de intrusiones. Los programadores, por ingeniosos que sean, no deben de pasar por -- alto los controles.
- e).- Ningún sistema puede evitar de manera absoluta las intrusiones malintencionadas, pero es posible hacer que resulte -- muy difícil eludir los controles. Los --- usuarios de la base de datos deben de ser sometidos a un proceso de identificación positiva antes de tener acceso a ella.
- f).- El sistema debe de tener capacidad para -- verificar que sus acciones han sido auto-

rizadas.

- g).- Las acciones de los usuarios deben de ser supervisadas, de modo tal que pueda descubrirse cualquier acción indebida o errónea.

### Accesibilidad

Algunos datos se usan con mucha frecuencia, -- mientras que otros raramente. Es deseable almacenar los datos de uso frecuente de manera que resulte fácil y rápido accederlos. Los datos de uso ocasional se almacenarán en cambio de manera más económica.

Para ser más objetivos, pondremos el ejemplo de una oficina; la información que se usa diariamente en ella se encuentra en los archivos de las secretarías, y la información que solo se usa o consulta ocasionalmente se quedará probablemente en el sótano, de modo que no estorbe y su almacenamiento no cueste mucho. El equivalente del sótano en una computadora, podría ser la cinta magnética, mientras que los datos de uso frecuente se encontrarán en disco o pack, de tal manera que se les pueda leer en fracciones de segundo. Toda base de datos más o menos compleja tendrá múltiples niveles de facilidad de acceso.

## Migración de Datos.

Este tópicó está muy relacionado con el anterior, debido a que en cierto momento se requiere tener más accesibilidad a un cierto conjunto de datos que no se requería, como lo es por ejemplo la música POP, de una popularidad efímera. En la bolsa de valores por ejemplo, suele surgir repentinamente un gran interés por obtener información acerca de acciones que el mes anterior no tenían prácticamente movimiento alguno.

A medida que cambia la popularidad de un conjunto de datos, será conveniente mudarlos dentro -- del almacén a posiciones más o menos accesibles de acuerdo con su actividad. En algunos casos no se mudan los datos, pero sí se modifican los índices que se utilizan para direccionarlos. En algunos casos - ésto se hace automáticamente, en otros, la operación está a cargo de los programadores de sistemas del Administrador de datos.

## Independencia de Datos.

La característica más importante acerca de las bases de datos es la Independencia de Datos por lo que la hemos dejado para su estudio hasta el final del presente capítulo, y antes de entrar de lleno - en él haremos un poco de historia.

Antes de que aparecieran las computadoras de la tercera generación hacia 1965, los archivos se organizaban como se muestra en la figura I-1.

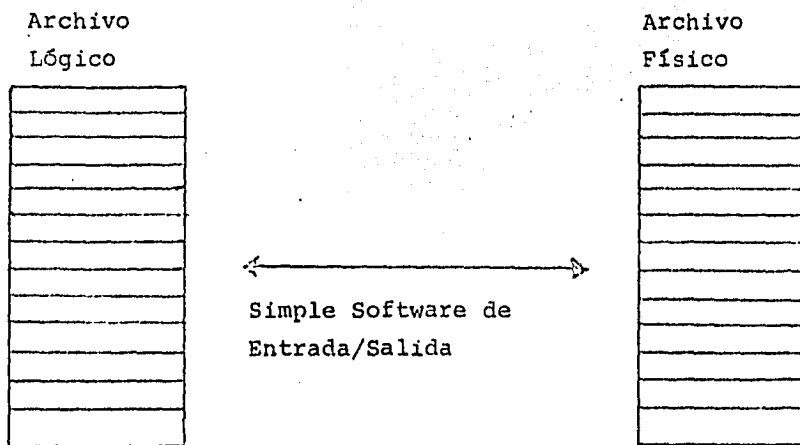


FIG. I-1 ORGANIZACION DE ARCHIVOS ANTES DE LAS COMPUTADORAS DE LA TERCERA GENERACION.

Estos modos de almacenamiento contaban con las siguientes características:

- a).- Archivos organizados de modo secuencial o simple.
- b).- Estructura física de los datos esencialmente igual a la estructura de los archivos lógicos.
- c).- Procesamiento en lotes, sin acceso en tiempo real.
- d).- Del mismo archivo existen varias copias porque se guardan las generaciones anteriores de datos.
- e).- El software se ocupa solo de las operaciones de entrada salida.
- f).-El programador de aplicaciones diseña la distribución física de los datos y la incorpora a los programas de aplicación.
- g).- Si se cambia la estructura de los datos o los dispositivos de almacenamiento, los programas de aplicación deben volver a escribirse, recompilarse y probarse.
- h).- Los datos se diseñan y optimizan, por lo general para una única aplicación.

- i).- De ahí que los mismos datos difícilmente se intercambian entre aplicaciones.
- j).- Alto nivel de redundancia entre los archivos de datos.

Como se nota a partir de las características, estos modos de almacenamiento no contaban con una Independencia de datos, es decir, si se modificaba la organización de los datos o se cambiaban los dispositivos de almacenamiento, el programador estaba obligado a volver a escribir los programas y repetir desde luego los procesos de compilación y depuración. Para actualizar un archivo, se escribía de nuevo. Se conservaba el original, el padre y también la generación anterior, el abuelo, y a menudo antepasados más remotos además de que la mayoría de los archivos servían para una sola aplicación, muchas veces los datos servían para muchas aplicaciones pero organizados de diferente manera lo que provocaba un alto nivel de redundancia.

A finales del decenio 1960-1970, se reconoció la naturaleza cambiante de los archivos y de los dispositivos de almacenamiento y se intentó proteger al programador contra los efectos de los cambios que se introducían en el hardware. El software hizo posible modificar la distribución física de los datos, sin que por ello se altere su estructura



lógica, siempre que no se introdujesen cambios en los contenidos de los registros ni en la estructura fundamental de los archivos, resultando una organización como se muestra en la figura: I-2.

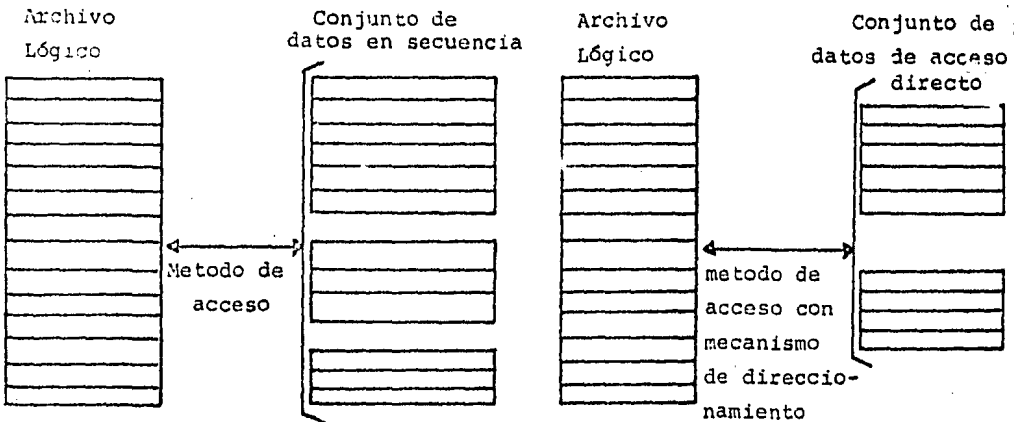


FIG. I-2 ORGANIZACION DE ARCHIVOS, PREDOMINANTE AL FINAL DEL DECENIO 1960-1970

Este método de acceso tenía las siguientes características:

- a).- Es posible el acceso secuencial o el acceso directo -- (al azar) a los registros (no a los campos).
- b).- El procesamiento se hace por lotes, en línea, o en tiempo real.
- c).- Se distingue la organización lógica de la organización física, pero las relaciones entre ellas son bastante sencillas.
- d).- Pueden cambiarse las unidades de almacenamiento sin necesidad de modificar los programas de aplicación.
- e).- Las estructuras de datos son por lo general de los tipos secuencial, secuencial indizado, o de acceso directo simple.

- f).- Por lo general no hay recuperación por -- clave múltiple.
- g).- Se admiten ciertos recursos de seguridad, pero no muy fiables.
- h).- Hay todavía tendencia al diseño y optimización de los datos principalmente para una aplicación.
- i).- Existe todavía mucha redundancia en los - datos.
- j).- Cuando se usan estructuras jerárquicas, - el programador tiene que construir, por - lo común, las relaciones de padre a hijo.
- k).- El software provee "métodos de acceso", - pero no "administración de datos".

En esta época los archivos estaban diseñados - también como los de la primera, para una aplicación determinada o para aplicaciones muy similares. Si - se diseñaba un archivo por ejemplo para el departamento de compras de una empresa se incluían en él - los elementos de datos necesarios para la aplica--- ción compras, pero si dentro de la misma empresa la aplicación obligaciones a pagar necesitaba práctica mente la misma información, se tenía no obstante -- que diseñar los archivos del caso, ya que el otro - archivo estaba estructurado según las conveniencias del departamento de compras y no permitía hacer ave riguaciones necesarias para la aplicación obliga---

ciones a pagar. Es evidente que si las dos aplicaciones hubieran podido aprovechar el mismo conjunto de archivos se habría economizado tiempo de procesamiento y espacio de almacén.

Con la evolución del procesamiento de los datos comerciales se comprendió que era conveniente - independizar los programas de aplicación no solo de los cambios en el hardware de almacenamiento y de las consecuencias del aumento del volumen de los archivos, sino también respecto a las eventuales adiciones hechas a los datos ya almacenados, tales como nuevos campos y nuevas relaciones. Este criterio se ha venido desarrollando desde principios de la década 1970-1980 con las siguientes características:

- a).- De los mismos datos físicos se derivan múltiples bases de datos lógicos.
- b).- Se puede tener acceso a los mismos datos de diferentes maneras, según los requisitos de la aplicación.
- c).- El software provee los medios para disminuir la redundancia.
- d).- Las distintas aplicaciones comparten los mismos elementos de datos.
- e).- La ausencia de redundancia facilita la conservación de la integridad de los datos.

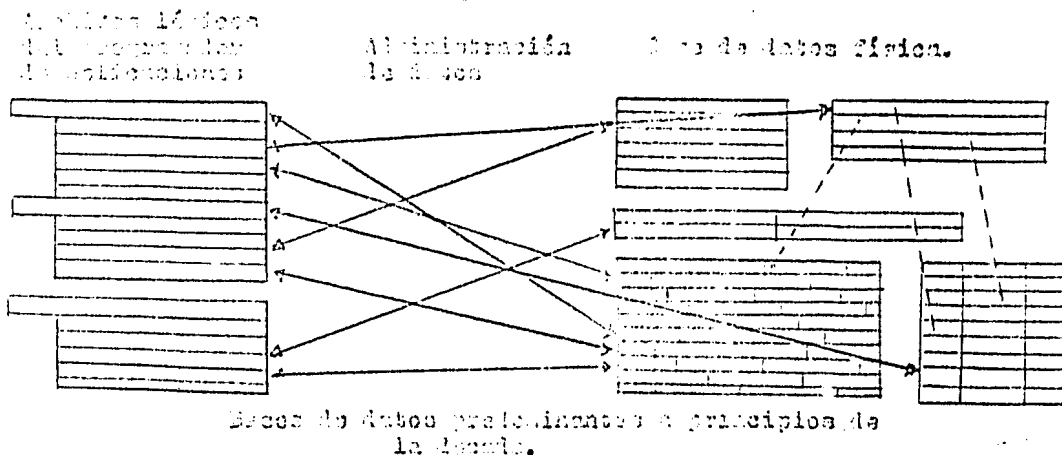
- f).- La organización del almacenamiento físico es independiente de los programas de aplicación. Pueden cambiarse a menudo para mejorar el desempeño de la base de datos sin modificación de los programas de aplicación.
- g).- Los datos son direccionables en los niveles de campo y de grupo.
- h).- Es posible la recuperación por claves múltiples
- i).- Se utilizan formas de organización de datos -- muy complejos sin que ello se refleje en los programas de aplicación.

La estructura de la base de datos tendrá que ser eventualmente modificada para mejorar su rendimiento o para permitir otros tipos de averiguación. Cambiarán asimismo las necesidades de los usuarios y los tipos de averiguación que le interesan.

No se puede imponer a una empresa una estructura de datos inmutable así como tampoco un grupo de programas de aplicación no modificables ya que tendrían como consecuencias en el primer caso tener -- que dedicar todos los recursos de software a modificar los programas de aplicación existentes, y en el segundo caso, el software se debe referir a los datos a nivel de elemento (ítem de datos o campo) más bien que a nivel de registro.

Estas técnicas se van venido perfeccionando y-

se ha pasado de la organización que se tenía a principios de la década (como se observa en la figura 1-3 a la que se tiene actualmente como se verá un poco más adelante.



La medida en que se exige al programador de aplicaciones de tener que conocer los detalles de los datos físicos varía de un sistema a otro. Un ejemplo muy fácil para ver la dependencia es el tomar un coche de alquiler para ir a ver una película determinada, posiblemente con algunos conductores usted solo tenga que dar el nombre de la película, con otros el nombre del cine, con otros tal vez tenga que dar la dirección y con los más despistados tenga que hacerla hasta de guía, del mismo modo el programador de aplicaciones puede simplemente tener que dar el nombre de aplicación del elemento de datos o registro que necesita. Con otro software, tendría que dar, posiblemente la identificación del ítem y también el nombre del conjunto de datos en que se halla. En el peor de los casos tendría que saber como está organizado el conjunto de datos y como determinar la dirección de máquina del ítem y finalmente en el caso de que no halla independencia de datos, el programador tendrá que conocer el formato físico exacto del registro y en cierto modo actuar como guía.

Al proceso de convertir la referencia que el programador de aplicación hace a un registro lógico y sus ítems de datos, en una referencia de máquina a un registro físico y sus ítems de datos se le llama vinculación (Binding). Una vez producida la vinculación el programa no es ya independiente de los datos físicos.

Cuando los primeros sistemas de bases de datos hubieran estado en uso durante cierto tiempo, se patentiza la necesidad de un mayor grado de independencia de datos. La estructura lógica general de datos se hizo más compleja en muchos casos y, al crecer la base de datos en tamaño, se hizo inevitable el cambio de la estructura lógica general. Resultó importante que esa estructura general pudiese cambiar sin forzar el cambio de los muchos programas de aplicación que la utilizaban. En muchos sistemas, los cambios de la estructura lógica general de los datos son: La evolución permanente. Por esta razón se necesitan dos niveles de independencia de datos: La independencia lógica y la Independencia física.

Por independencia lógica de los datos se entiende la modificación de la estructura lógica general sin afectar a los programas de aplicación (el cambio desde luego no debe eliminar ninguno de los datos que el programa necesita).

Por Independencia física de los datos se entiende que pueden modificarse la distribución y la organización física de los datos sin afectar ni la estructura lógica general ni los programas de aplicación.

El concepto de Independencia de datos en sus dos niveles se ilustra en la figura: III-2.

El bloque que aparece en el centro de la ilustración representa la estructura lógica global de los datos llamada a menudo vista lógica global de los datos. Esta vista puede ser enteramente distinta de la que ofrece la estructura física y de la propia de los programas de aplicación. El software de la base de datos se encargará, en efecto de convertir la vista que el programador de aplicaciones tiene de los datos en la vista lógica global y transformará luego, esta vista lógica global en la representación física.

Las características de una base de datos actual se pueden resumir como sigue:

- a).- El software procura la independencia lógica y física de los datos, permitiendo que exista una vista lógica global independientemente de ciertos cambios en las vistas de los programas de aplicación o de la distribución física de los datos.
- b).- Los datos pueden evolucionar sin que se incurra en costos de mantenimiento excesivos.

- c).- Se proveen medios para que un administrador de datos actúe como Controlador y custodio de los datos y asegure que la organización de estos - es siempre la mejor para los usuarios en general.
- d).- Se proveen procedimientos eficaces para controlar el secreto, la seguridad y la integridad - de los datos.
- e).- En algunos sistemas se utilizan archivos para permitir una rápida exploración de la base.
- f).- Las bases de datos se diseñan de modo que provean respuestas a tipos de averiguación no previstos por el diseñador.
- g).- Se facilita la migración de datos.
- h).- El software provee un lenguaje para la descripción de datos para el administrador de datos, un lenguaje de ordenes para el programador de aplicaciones, y a veces un lenguaje de interrogación para el usuario.

El propósito fundamental de la estructura que vemos en la figura de las bases de datos actuales - es el de permitir la máxima libertad para cambiar - las estructuras de los datos sin tener que rehacer mucho de lo ya hecho en la base de datos, ésto se logra si la base de datos cuenta con las caracterís



características deseables que se muestran en la figura de abajo, esta figura también nos indica si es posible hacer esos cambios sin reestructurar la organización física del almacén. Las cruces que aparecen en la figura representan objetivos para el diseño contemporáneo del software de base de datos.

- Se agrega un nuevo programa de aplicación, el que utiliza nuevos tipos de datos
- un programa de aplicación exige una representación modificada de los datos existentes (por ejemplo punto flotante en lugar de punto fijo).
- Se agrega un nuevo programa de aplicación, que utiliza los tipos de datos existentes.
- Se insertan nuevas ocurrencias de registro o se eliminan las viejas.
- Se mejora la descripción lógica global de los datos, o se crean nuevas relaciones entre los datos.
- Se consolidan dos bases de datos.
- Se mejora la organización física de los datos posiblemente se usan diferentes representaciones.
- Se modifican los métodos de direccionamiento.
- Los datos se mudan a un tipo diferente de volumen
- Se modifica el software.
- Se modifica el hardware.

	No hay cambios en los otros programas de aplicación	No hay cambio en la descripción lógica global de los datos.	No hay cambio en la organización del almacenamiento físico de datos
Se agrega un nuevo programa de aplicación, el que utiliza nuevos tipos de datos	X	X	X
un programa de aplicación exige una representación modificada de los datos existentes (por ejemplo punto flotante en lugar de punto fijo).	X	X	X
Se agrega un nuevo programa de aplicación, que utiliza los tipos de datos existentes.	X		
Se insertan nuevas ocurrencias de registro o se eliminan las viejas.	X	X	X
Se mejora la descripción lógica global de los datos, o se crean nuevas relaciones entre los datos.	X		
Se consolidan dos bases de datos.	X		
Se mejora la organización física de los datos posiblemente se usan diferentes representaciones.	X	X	
Se modifican los métodos de direccionamiento.	X	X	
Los datos se mudan a un tipo diferente de volumen	X	X	
Se modifica el software.	X	X	
Se modifica el hardware.	X	X	

FIG. I-4 CARACTERÍSTICAS DESEABLES EN UNA BASE DE DATOS.

En realidad así como en la práctica los datos son pocas veces redundantes, así también son pocas veces completamente independientes.

Cuando un determinado conjunto de datos sirve a una variedad de programas de aplicación, cada uno de estos percibe, en general diferentes relaciones entre aquellos. En gran medida, es preocupación principal en la organización de la base de datos la representación de las relaciones que existen entre -- ítems de datos y registros, así como también el como y en donde se almacenan. En las bases de datos - previstas para aplicaciones diversas pueden existir múltiples interconexiones entre los datos.

## C A P I T U L O   I I .

### ORGANIZACION DE LA BASE DE DATOS

#### INTRODUCCION.

De acuerdo a la definición de Base de Datos dada, dentro del capítulo anterior, nos encontramos ante la necesidad de disponer de un medio donde --- ubicar todos los datos significativos del organismo al que es aplicable el sistema de información.

Desde el punto de vista físico, el desarrollo actual de las computadoras electrónicas nos permite disponer de múltiples dispositivos (hardware) que cumplen satisfactoriamente con este requisito. Pero, emplear los medios y mecanismos que proporciona una computadora hace indispensable la selección y aplicación de una metodología organizativa encaminada a aprovechar eficientemente los recursos del sistema computacional, selección relacionada con la optimización de los tiempos de procesamiento y acceso del sistema, con el consecuente mejor empleo de los recursos económicos disponibles.

Resumiendo un poco, podemos decir que la organización física de los datos dentro de los dispositivos de memoria nos hace aprovechar de la mejor manera los recursos físicos de la computadora.

El desarrollo de una Base de Datos no solo contempla el aspecto físico del almacenamiento; con anterioridad a esta organización, el programador descubre y encuadra dentro de un sistema lógico a los datos, de acuerdo a las cualidades y relaciones importantes que existen en ellos, esto es: los datos muestran características notables y relaciones fundamentales, las que determinan que el analista establezca sistemáticamente la relación u organización lógica de la Base de Datos.

De lo anterior encontramos que es importante tener en cuenta y diferenciar en una Base de Datos un tipo de organización física y una organización lógica; la primera desarrollada por la necesidad de aprovechar eficientemente los recursos de la computadora y por la misma organización lógica y, ésta última, proyectada, generada y manejada por el programador como la estructura básica que agrupa y da sentido a la Base de Datos.

En adelante utilizaremos las palabras física y lógica para distinguir las dos estructuras que adoptan los datos y las relaciones que existen entre ellos. La descripción física de los datos se ocupa de como se les registra en el hardware, la descripción lógica se refiere a la forma en que los datos se presentan al programador de aplicaciones. Asimismo emplearemos los términos enunciados para cali

ficar diversos aspectos de los datos; relaciones lógicas, estructura lógica y descripción lógica cuando consideramos los datos desde el punto de vista - del programador, en cambio se le llamarán relaciones físicas, estructura física y descripción física si detallan la manera de almacenar y relacionar físicamente los datos en los dispositivos de almacenamiento.

Las estructuras de datos y sus afines pueden ser diferentes desde el punto de vista del programador y desde el punto de vista de la organización física, hay posiblemente muchas diferencias entre la organización lógica y la organización física, debidas a las limitaciones de los equipos (la capacidad, forma de almacenaje, etc.), así como la arquitectura propia del sistema de computo, no obstante, es ineludible un vínculo entre dichas estructuras. La forma en que concibe la estructura el programador y como la estructura se encuentra físicamente se vincula y se transforma por medio del software. El software presenta al programador los registros lógicos ordenados convenientemente a partir de la estructura física de los datos, análogamente, el mismo software introduce a la organización física los registros del sistema lógico, ésto último lo podemos observar en la figura II-3, que da un esquema de la diferencia que existe entre la estructura ló-

gica y la estructura física, así como la relación de transformación cumplida por el software.

### ORGANIZACION LOGICA

En este punto iniciaremos el desarrollo de la Organización Lógica sin dejar de hacer referencia a la estructura física, a los programas de aplicación y a las relaciones correspondientes.

### DEFINICION DE TERMINOS.

Antes de enunciar las características de la Organización Lógica definiremos algunos términos empleados durante su estudio, dichos términos se ilustran mediante la figura II-1.

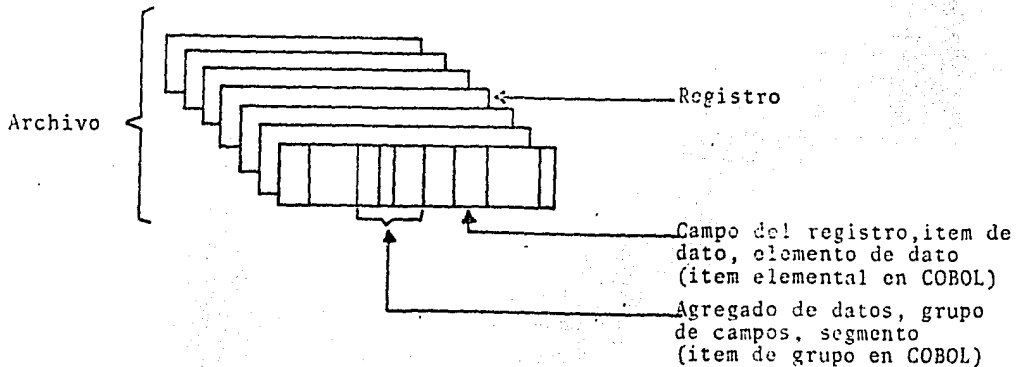


FIG. II-1 TERMINOLOGIA PARA DESCRIBIR LOS DATOS DESDE EL PUNTO DE VISTA DEL PROGRAMADOR DE APLICACIONES.

**BYTE.-** El byte es el grupo de bits más pequeño con dirección propia, convencionalmente el byte comprende ocho bits. Más adelante en programas de aplicación identificaremos el byte con un carácter de una cadena de información.

**CAMPO DEL REGISTRO.-** El campo es el grupo de datos nominado más pequeño. Puede formarse por cualquier número de bytes o caracteres.

AGREGADO DE DATOS.- Es una colección de campos, dentro de un registro al que se concibe como un todo. Por ejemplo: Nombre, puede ser el agregado de datos compuesto por Primer apellido, Segundo apellido y Nombre de pila.

REGISTRO.- El registro es una colección nomina da de campos, no necesariamente campos íntimamente relacionados.

SEGMENTO.- Formado por uno o más campos de datos, es la cantidad básica de datos que se permuta entre el programa de aplicación y la base de datos bajo el control del software. A menudo se identifica al Segmento con el Agregado de datos, debido a que éste último tiene el significado de un todo.

ARCHIVO.- Archivo es la colección nominada de todas las ocurrencias de un mismo tipo de registro lógico.

BASE DE DATOS.- Colección de ocurrencias de -- múltiples tipos de registros, pero incluye además - las relaciones que existen entre registros, entre - agregados y entre campos de datos.

#### ORGANIZACION LOGICA GLOBAL DE LOS DATOS.

En párrafos anteriores afirmamos que la organización lógica es concebida, desarrollada y manejada como la estructura conceptual básica que agrupa y -

da sentido a la base de datos. Además hemos definido algunos términos referidos a los datos dentro de un sistema o estructura lógica.

Resumiendo lo anterior encontramos que: son -- los datos los elementos fundamentales de la estructura, agrupados en campos, agregados, registros y - archivos vinculados orgánicamente. Asimismo, la organización lógica intrínsecamente relacionada con - la organización física del almacenamiento y los programas de aplicación, todo esto conformando la Base de Datos.

La Organización lógica global de los datos es pues la estructura lógica general que determina toda la organización de archivos lógicos de aplica---ción. Es una visión más amplia, sencilla y completa que permite la independencia tanto estructural como operativamente, de lo que hemos definido como Organización Lógica de la base de datos.

La Organización lógica global prevee una es---trutura general que contenga los datos necesarios para todos los archivos de aplicación, y sus características estructurales satisfacen las necesidades de Independencia, expansión, cambio y actualización de los tres distintos tipos de organización de las bases de datos.

Es importante destacar algunas características



significativas de las bases de datos que nos servirán para una mejor comprensión de lo que llamamos ORGANIZACION LOGICA GLOBAL DE LOS DATOS.

a).- Cambio y crecimiento continuo.- Una de -- las características más importantes de la mayoría -- de las bases de datos es la de mantenerse en plena crisis de cambio y crecimiento. Las bases de datos deben prestarse a una fácil reestructuración siempre que haya que agregarle nuevos tipos de datos o utilizarlas para nuevas aplicaciones.

La idea básica en la implantación de una base de datos es la de que los mismos datos deben de ser aprovechados para tantas aplicaciones como sea posible. Es posible que el analista proyecte la base de datos, teniendo a la vista una aplicación determinada, con carácter definitivo en cuanto a su contenido y utilización. Por consecuencia, vincula sus datos a una organización física que considera eficiente para esa estructura en particular. Las necesidades varían de manera imprevista. Las estructuras -- tienen que ser correspondientemente modificadas, -- con la consecuencia de que habrá que volver a escribir y depurar muchos de los programas de aplicación que se vienen usando.

b).- Desarrollo a nivel ítem o campo.- Mencionamos que la base de datos tiende a ser eventualmente modificada para permitir nuevos tipos de aplica-

ción. Se presenta la necesidad de agregar nuevos tipos de registro y en los registros existentes incluir nuevos ítems de datos.

Si se pretende que sea posible agregar nuevos elementos de datos a los registros sin que ello trastorne los programas de aplicación, es preciso que el software se refiera a los datos a nivel de ítem más bien que a nivel registro.

c).- Según los programas de aplicación, los analistas crean archivos lógicos a partir de la base de datos.- En el punto A afirmamos que los datos deben ser aprovechados en múltiples aplicaciones; asimismo, en el punto B decíamos que el software debe referirse a los datos a nivel ítem. Estas afirmaciones son complementarias: cada aplicación requiere de un específico archivo lógico y el programador puede desarrollar tal archivo si el software maneja ítems de registro.

Otro punto que debemos destacar ahora es que hemos introducido un nuevo concepto: el Archivo Lógico de Aplicación. La base de datos permite diversas aplicaciones; el programador emplea para cada aplicación un específico archivo lógico. Independientemente de como se organicen los datos, el programador debe ver el archivo como una estructura relativamente simple, planeada de acuerdo con las ne-

cesidades de su aplicación.

d).- Independencia lógica y física de los datos.- En el punto A apuntamos que el cambio y crecimiento de la base de datos implica un cambio en la estructura lógica. Por independencia lógica de los datos se entiende que la modificación de la estructura lógica general no afecta a los programas de aplicación.

Por independencia física de los datos se entiende que pueden modificarse la distribución y la organización física de los datos sin afectar ni la estructura lógica general ni los programas de aplicación.

La figura II-2 ilustra las características enunciadas así como los conceptos de Independencia lógica y física. Podemos apreciar que los archivos lógicos de aplicación y la organización de almacenamiento. Mantienen una independencia entre sí, pero se encuentran ampliamente relacionados por la Organización lógica global de los datos; la cual se esquematiza mediante el bloque que aparece en el centro de la ilustración.

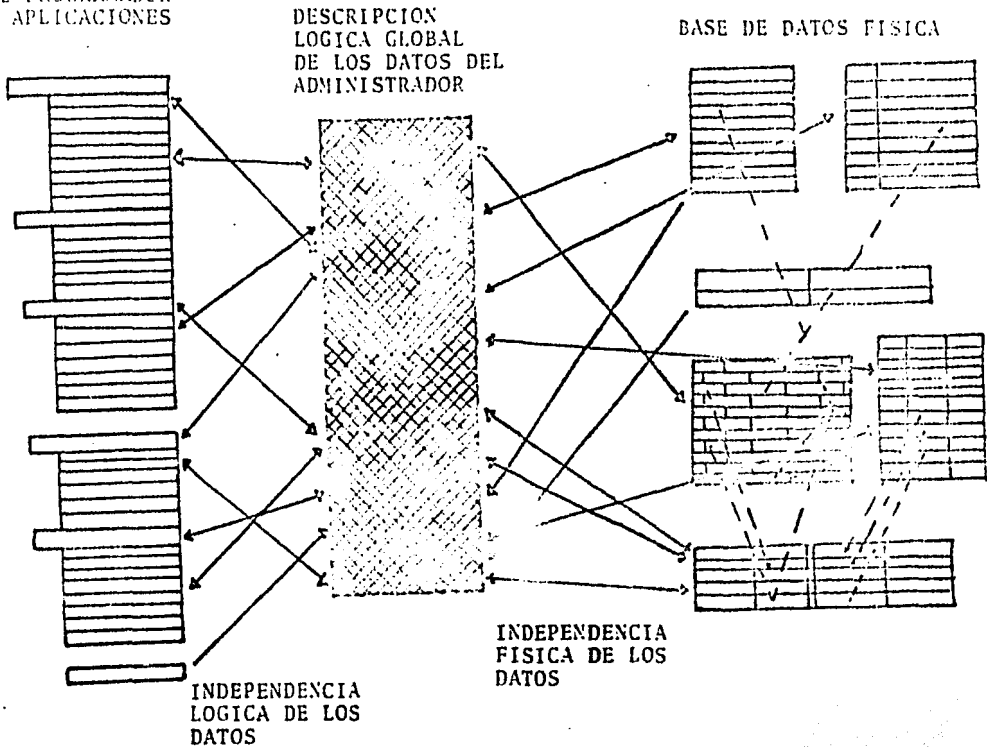


FIG. II-2 SISTEMA DE BASE DE DATOS CON  
INDEPENDENCIA LOGICA Y FISICA.

TRES TIPOS DE ORGANIZACION.

Existen tres aspectos en la organización de los datos derivados del concepto de Independencia de datos; y son los que se observan en la figura II-2. Nos referimos a ellos como:

- Organización de Archivos
- Organización Lógica Global de los Datos
- Organización Física del Almacenamiento.

La organización de archivos se preocupa de la vista de los datos tal como los percibe el programador de aplicaciones.

La organización lógica global se preocupa por la organización general de la base de datos de la cual pueden derivarse múltiples organizaciones de archivos.

La organización física del almacenamiento tiene por objeto la representación y distribución física de los datos y la organización de éstos en las unidades de almacenamiento.

El software de la base de datos se encargará - de convertir la vista que el programador de aplicaciones tiene de los datos en la vista lógica global, y transformará luego, esta vista lógica global en la representación física.

Si la función de la base de datos fuese meramente la de almacenar datos, su organización resultaría por demás simple. En efecto, la mayoría de -- sus complejidades surgen por el hecho de que tam--- bién tienen que indicar las relaciones que existen entre los diversos ítems de datos que almacena.

Hay una diversidad de maneras de representar - lógicamente las relaciones entre datos. Algunas son buenas y otras resultan confusas y conducen a errores de interpretación, las hay con demasiadas limitaciones y que no permiten la representación de algunas de las relaciones que existen en la realidad, algunas imponen limitaciones a la modificación o a la actualización de la base, aquí vamos a tratar al gunas de estas maneras.

La vista del usuario debería al menos en teo-- ría formularse sin preocupación alguna por la repre-- sentación física ya que esto es tarea del Adminis-- trador de Datos como se muestra en la Fig. II-2.

El software debiera hacer tan impenetrables -- las barreras de la figura de modo que cuando un pro

grama de aplicación cambie los datos que utiliza, - no se vean afectados los demás programas de aplicación y no sea necesario rehacer los programas de -- aplicación cuando se reestructure la descripción lógica global de los datos o en el otro caso, cuando la organización física pudiera ser afinada o cambia da totalmente sin necesidad de rehacer ni la des--- cripción lógica global ni los programas de aplica-- ción.

En la práctica, la industria de la computación no ha llegado aún a desarrollar un software de administración perfecto, de modo que siempre tendrá que llegarse a soluciones de compromiso entre las des-- cripciones lógicas y físicas como veremos.

### ORGANIZACION FISICA.

En la organización física de los datos no nos preocuparemos de como ve el usuario los datos, sino como se organizan y distribuyen éstos en los medios de almacenamiento.

### NIVEL INTERNO.

Podemos dividir la organización física en dos niveles: Un nivel que ve la forma en que los datos son almacenados, y el otro el cómo son almacenados estos datos.

En el primer nivel se encuentra la representa-

ción de toda la base de acuerdo a las múltiples ocurrencias de los diferentes tipos de registros de almacenamiento. En realidad este nivel se encuentra fuera de la forma física de almacenamiento ya que solo describe la manera en que los datos son almacenados. El segundo nivel se refiere a el cómo, esto es, se refiere a los registros físicos, bloques, y a su distribución en los dispositivos. Tampoco se refiere a los datos desde el punto de vista de sus aplicaciones en sí.

El primer nivel se refiere específicamente a los elementos necesarios que debe conocer el manejador de la base de datos, es decir, las relaciones que guardan entre sí los datos, como son representados los mismos, la existencia de índices, tablas de conversión, si el acceso es de tipo secuencial o no, las técnicas de compactación, en fin todos los elementos que son necesarios para definir una estructura de almacenamiento que permita una mejor búsqueda y localización de los datos, minimización de tiempo en estas funciones, el buen rendimiento operativo, etc.

Para la realización de una estructura que cumpla los objetivos mencionados arriba, existen varias características que resulta conveniente conocer. Estas son:

## REDUNDANCIA

Hemos visto que uno de los objetivos de un banco de datos es la eliminación de la redundancia de datos. Si bien en esto la organización lógica tiene mucho que ver, la estructura de almacenamiento es la que más debe ayudar a la eliminación de la redundancia sin tomar en cuenta que tan compleja pueda resultar esta estructura, debido a que ésta nunca será vista por el usuario.

## TIPO DE PROCESAMIENTO.

Es necesario conocer que tipo de procesamiento es el que se va a realizar, de acuerdo a las necesidades de información que tenga la organización que hará uso de la base, por ejemplo, hay trabajos como la preparación de nóminas en la cual la relación se realiza en forma secuencial, esto es, se recorre el archivo leyendo registro por registro de acuerdo a un plan determinado, y realizando los recibos; a este tipo de trabajo se le llamará proceso secuencial.

En cambio hay trabajos como la reservación de hoteles, actualización de inventarios, en los que se trabajará mejor con un proceso de tipo aleatorio, esto significa, que se presentará cuando el movimiento del archivo en inserciones o supresiones de datos es muy frecuente y el registro se verifica y/o actualiza al recibirse cada orden. Para esto es



necesario un acceso rápido a los datos que se necesitan, pues no se cuenta con tiempo suficiente para explorar todo el archivo; a este tipo de trabajo se le llamará procesamiento al azar.

La organización física debe realizarse en función del tipo de procesamiento. Los registros que siempre se usan secuencialmente y en el mismo orden se podrán almacenar fácilmente y sin requerir de una búsqueda más elaborada. En cambio cuando es en forma aleatoria los medios de localización deberán ser más elaborados y es entonces posible que los medios de direccionamiento resulten factor dominante para la distribución física de los registros.

#### RAZON DE LA ACTIVIDAD DEL ARCHIVO.

Es de todo punto necesario conocer que cantidad de datos se utilizan en un archivo secuencial que ha sido leído. Por ejemplo, en un proceso de facturación, explorado en forma secuencial, se lee todo el archivo o un bloque de datos del mismo, pero sólo se necesitarán algunos registros del total leído. Por esta razón es conveniente conocer las relaciones que guardan los datos leídos en un pasado acceso al archivo, con respecto a los datos utilizados.

Para poder determinar que tipo de almacenamiento usar (disco, cinta, etc.), ya que de esta manera

sabremos si la búsqueda efectuada es más de forma -  
secuencial que aleatoria (índice de la relación al-  
to), o viceversa (índice de la relación bajo). A es-  
ta relación se le llama razón de actividad del ar-  
chivo ( R A T L O ) por sus siglas en inglés y se de-  
fine como:

$$R A T L O = \frac{\text{Número de registros leídos}}{\text{Número de registros explorados}}$$

También es necesario conocer si el procesamien-  
to es en línea o en lote. Si es en línea los tiem-  
pos de respuesta deberán ser cuidados así como la -  
disponibilidad de los datos, esto es, que si exis-  
ten fallas en el sistema éstas no afecten el acceso  
a la información; esto se puede lograr duplicando -  
los datos en otro dispositivo o bien ubicando los -  
datos en unidades desmontables que se pueden trans-  
ferir a otra máquina y así asegurar una transmisión  
de datos en línea en forma constante.

### ARCHIVOS DINAMICOS

Existen archivos en los cuales la eliminación  
e inserción de los datos se efectúa en forma fre-  
cuente, entonces la organización física debe estar  
preparada para estas inserciones o eliminaciones, -  
ya que esto puede afectar la óptima distribución fí-  
sica de los datos.

La mayoría de los archivos que trabajan con estas características requieren de una reorganización periódica para poder mantener una organización óptima, y es en este tipo de archivos en los que la facilidad o duración del proceso de organización (mantenimiento de la base) llega a ser un factor importante para la estructura de almacenamiento de los datos.

Otro aspecto que se debe tomar en cuenta es -- que puede haber casos en los cuales las inserciones pueden ser más abundantes que las eliminaciones, -- con lo que el archivo crece. Este crecimiento requiere por lo tanto, de organizaciones que acepten la expansión y que no estén limitadas por el tamaño de las celdas, bloques de índices y otras restricciones.

#### PROTECCION DE LOS DATOS.

Es necesario que existan medios dentro de la organización física que permiten reconstruir los datos que se consideran vitales. Esto se puede efectuar por medio de copias en medios de almacenamiento perfectamente actualizados, o mediante una estructura que facilite esta recuperación.

#### MEDIOS DE ALMACENAMIENTO

La mayoría de los datos si no todos, deberán ser almacenados en los dispositivos de almacenamien

to secundario (memoria auxiliar), pero como la unidad de procesamiento solo trabaja con los datos que se encuentran en la memoria principal, entonces necesitamos transferir los datos que se encuentran en la memoria auxiliar a la principal para poder trabajar con ellos; esto es, tenemos que ejecutar una --operación de entrada salida (I/O), lo cual ocasiona una pérdida de tiempo, ya que tenemos que localizar el registro en la unidad (tiempo de acceso) y después leerlo (tiempo de lectura); estos dos tiempos sumados forman lo que se llama tiempo de respuesta.

Es por esto que cuando hablamos de organiza---ción física, hablamos de dos niveles: el primero --que ya vimos, habla de la estructura de almacena---miento, y el segundo, se preocupa de la manera de -distribuir esa estructura en los medios de almacena---miento, economizando el mayor espacio posible y jun---to con los métodos de acceso, minimizando el tiempo de respuesta.

#### FORMAS DE ACCESO.

Los diferentes tipos de unidades de almacena---miento se clasifican en dos tipos básicamente: los dispositivos de acceso directo como son el disco fijo, paquetes de discos (disk pack) y los de acceso secuencial, que principalmente son las cintas magné---ticas; esta clasificación se debe a que los prime--

ros tienen la facilidad de acceder los registros en cualquier orden, mientras que los segundos lo hacen prácticamente en orden secuencial; esto significa - que si se quiere leer un registro que se encuentra en medio de la cinta, se tendrá que pasar por los - registros anteriores antes de poder leerlo, lo que trae una pérdida de tiempo en la lectura de los datos.

Teniendo en cuenta que el tiempo de acceso es mucho mayor que el tiempo de lectura, si queremos - minimizar el tiempo de respuesta, tendremos que ver la manera de disminuir el de acceso.

Si en lugar de transferir en cada operación de entrada salida, un registro de almacenamiento por - vez, transferimos un bloque completo de ellos, el - tiempo de acceso se divide entre todos los regis---tros de almacenamiento que sean transferidos en el bloque, en lugar de dividirse en un solo registro,- con lo cual disminuirémos en forma considerable el tiempo de acceso.

### REGISTRO FISICO

En los discos que son los dispositivos que más se usan, existe espacio en cada pista, delimitados por fronteras naturales de acceso llamados regis---tros físicos; de tal manera que cuando accedamos a disco, podemos extraer un registro físico o bloque.

Estos registros físicos son de longitud fija y se pueden contener varios registros de almacenamiento; esta forma de acceso nos permite poder disminuir el tiempo de respuesta, por lo que cuando diseñamos la organización física, es conveniente ver la mejor manera posible de agrupar los registros de almacenamiento para - por un lado utilizar el mayor número posible en cada transferencia ( R A T L O ).

Para aprovechar más eficientemente el espacio de almacenamiento y lograr una mayor transferencia de datos, los registros físicos son mucho más grandes que los del almacenamiento, que -- normalmente son pequeños. Esto se hace debido a que la cobertura del registro físico siempre resulta incompleto, quedando un espacio vacío en cada registro físico (fig II-3), este espacio es cubierto con un relleno de caracteres no significativos, el cual disminuye conforme es mayor la longitud del registro físico con respecto al de almacenamiento.

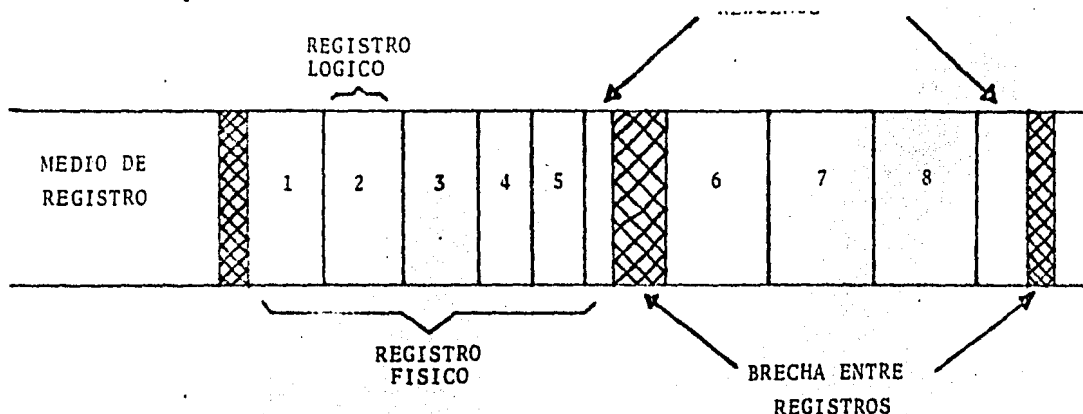


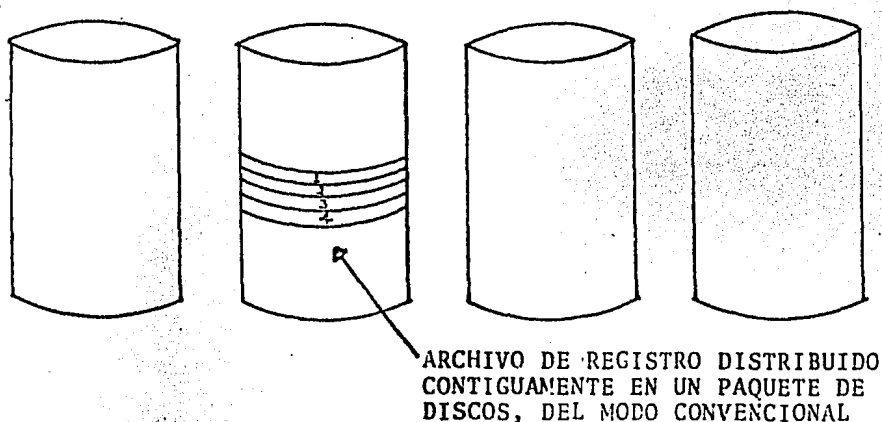
FIG. II-3 ORGANIZACION EN LA CUAL LOS REGISTROS FISICOS SON DE LONGITUD FIJA Y LOS LOGICOS DE LONGITUD VARIABLE.

#### CELDAS

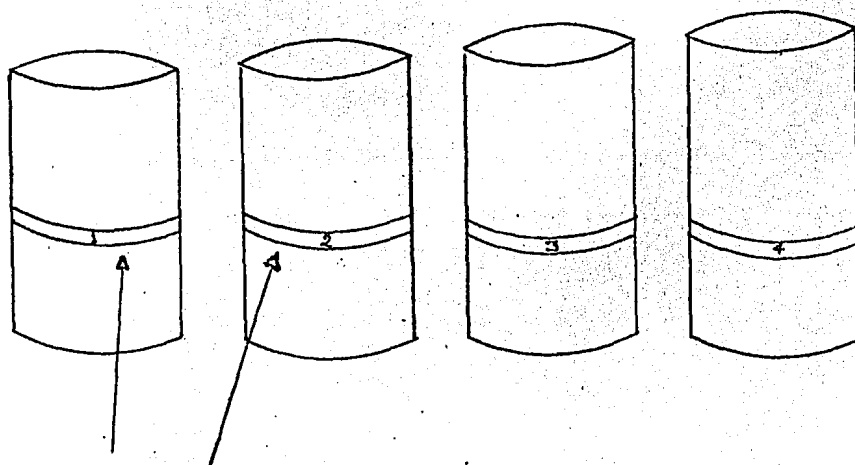
Además de los registros físicos existen otras zonas delimitadas por fronteras naturales del hardware, como son: los pistas, los cilindros, los módulos, etc., que llamaremos celdas. La importancia de las celdas estriba en que cuando pasamos de una celda a

otra en una búsqueda, el tiempo de acceso a un dato aumenta según una función escalón; si distribuimos nuestros datos en las celdas de tal manera que para un trabajo, dos o más de ellas puedan ser leídas simultáneamente decimos que esas celdas tienen -- una organización celular y acceso en paralelo, por el contrario si las celdas son leídas en forma secuencial se considerará una organización convencional en serie (fig. II-4).

### 1.- ORGANIZACION CONVENCIONAL EN SERIE



### 2.- ORGANIZACION CELULAR EN PARALELO



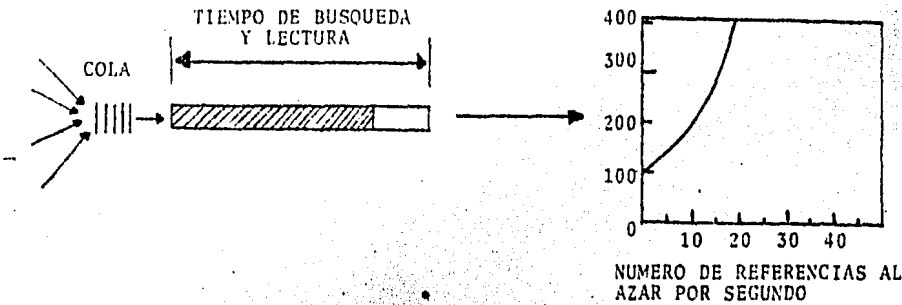
EL MISMO ARCHIVO DISTRIBUIDO A TRAVES DE LOS DISPOSITIVOS DE ALMACENAMIENTO DISPONIBLES CON EL FIN DE ABREVIAR EL TIEMPO DE RESPUESTA EN TIEMPO REAL, O CONSEGUIR UNA MAYOR PRODUCCION.

FIG. II-4

Debido a que en muchos sistemas los pedidos de registros se suceden al azar, se forman colas de datos en espera de ser servidos; esto origina que al tiempo de respuesta se deba no solo al tiempo de lectura y acceso; sino también al tiempo de espera en la cola.

Si observamos que pasa con un trabajo cualquiera cuando está organizado en forma secuencial y lo comparamos con uno organizado en forma paralela (fig. II-5), vemos que en la organización en paralelo, la búsqueda puede ser realizada simultáneamente, -- pero no así las lecturas. En este caso existen dos colas: una para la operación de búsqueda y otra para la de lectura, pero como la búsqueda es simultánea el tiempo de cola es menor y como consecuencia el tiempo de respuesta baja. Por lo tanto para minimizar el tiempo de respuesta en estos casos, es importante tomar en cuenta la organización en paralelo.

1. COLAS CON LA ORGANIZACION EN SERIE



2. COLAS CON LA ORGANIZACION CELULAR EN PARALELO

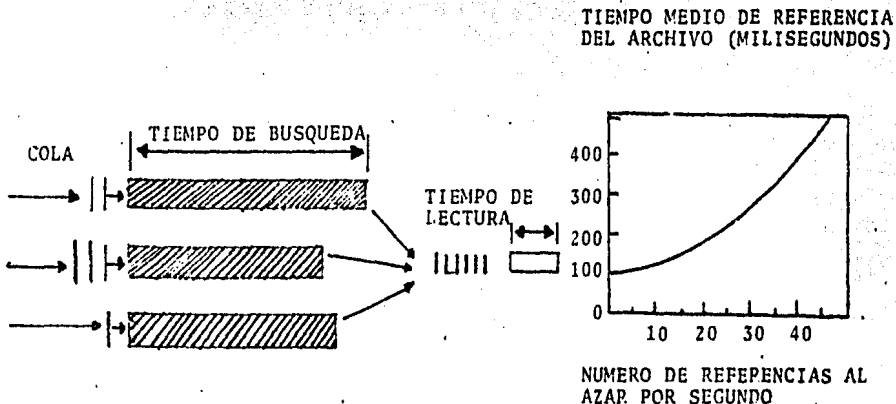


FIG.- II-5 SI LA MAYORIA DE LOS ACCESOS TIENEN POR OBJETO EL ARCHIVO DE LA FIGURA II-4 LOS TIEMPOS DE ESPERA EN COLA PARA EL ACCESO A ESTOS REGISTROS SERAN MAYORES CON LA ORGANIZACION CONVENCIONAL EN SERIE QUE CON LA ORGANIZACION CELULAR EN PARALELO.



PAGINACION.

Debido a la cantidad de datos a almacenar y la necesidad de minimizar los costos de almacenamiento, los sistemas tienen además de los dispositivos de acceso directo, dispositivos de acceso secuencial. Esto lleva a distribuir los datos según su frecuencia de uso, no solo dentro de un mismo dispositivo; sino en los diferentes medios de almacenamiento.

El traspaso de datos entre los diferentes niveles de almacenamiento se realiza por medio de páginas de longitud fijas; el tamaño de la página es un parámetro del sistema y no está determinado por la longitud del registro elegida por los programadores, por lo que los registros que van a ser transferidos tienen que ser empaquetados en las páginas.

## C A P I T U L O   I I I

### METODOLOGIA

#### INTRODUCCION.

La información procesada de alguna forma representa una abstracción de la realidad. Esta información, que es manejada por la computadora, consiste básicamente de un conjunto de datos seleccionados del mundo real y son considerados como una abstracción de la realidad ya que ciertas cualidades y propiedades de los objetos son ignorados por ser irrelevantes.

Cuando hablamos de información nos referimos a tres niveles. El primer nivel es el del mundo real, éste es el de los objetos, que llamaremos ENTIDADES; las cuales tienen una serie de propiedades que las describen. Una entidad puede ser un objeto tangible, por ejemplo, una pieza o un artículo; o bien ser intangible, tal como un suceso, el nombre de una tarea, la cuenta de un cliente o un concepto abstracto.

El segundo se refiere a las ideas que se utilizan para describir la información que se obtiene del mundo real. Esto es, las propiedades de las entidades a las cuales llamaremos ATRIBUTOS. Este es el campo de las ideas y de la información existente.

en la mente de las personas.

Y por último el tercer nivel, en el cual utilizamos una serie de signos para representar la información.

Por lo tanto al hablar de la base de datos en una computadora, nos estamos refiriendo a un conjunto de datos almacenados de acuerdo a estos tres niveles de información.

### ESQUEMAS.

Llamaremos esquema a la descripción lógica de la base de datos que queda determinado por los siguientes puntos:

- Los tipos de entidades que la forman.
- Los datos elementales que ocurren en cada tipo de entidad.
- Las propiedades estructurales de cada dato elemental.
- Los conjuntos de tipos de entidades ligadas entre sí.

Una base de datos puede estar formada por uno o más esquemas.

### RELACIONES Y REFERENCIAS CRUZADAS.

Los esquemas se presentan a menudo a modo de diagramas de bloques como se muestra en la figura

III-1, donde las líneas llenas que unen ciertos bloques representan las relaciones entre los atributos y las líneas punteadas re presentan referencias cruzadas.

Las relaciones indicadas en el esquema agregan una información que no es inherente a la entidad con la que se está relacionando, por ejemplo, no hay nada en la entidad Orden de compra -- que diga que artículos y en que cantidad se compran. Esta información se completa solo cuando se unen las entidades Renglón de compra con orden de compra por medio de los atributos número de proveedor y Número de artículo.

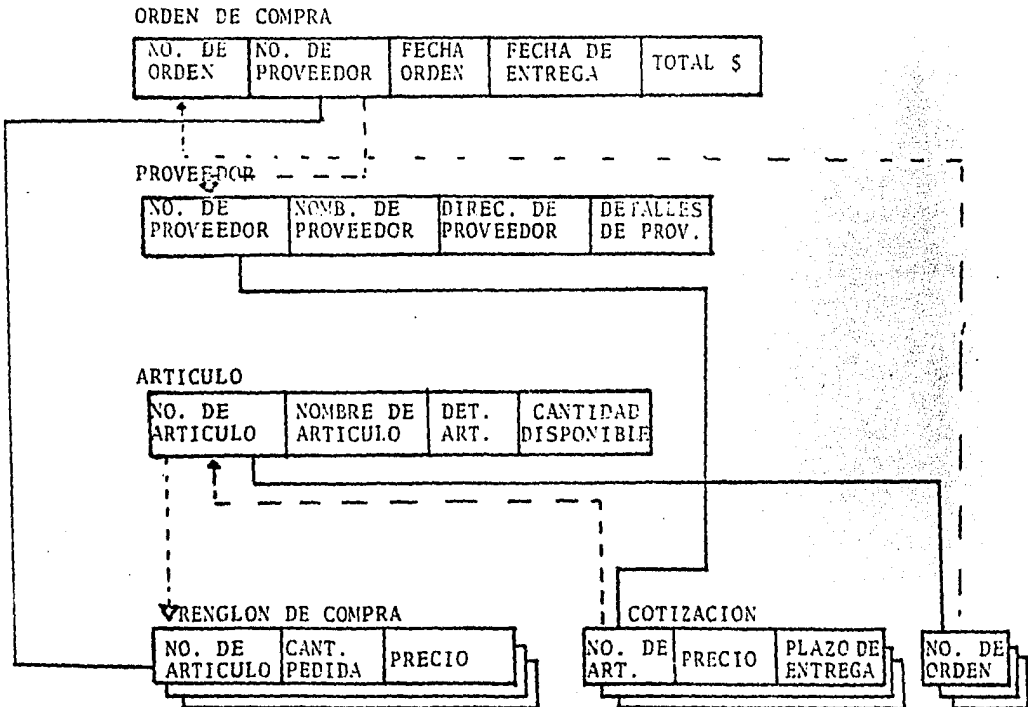


FIG. III-1 ESQUEMA. LAS LINEAS CONTINUAS ENTRE BLOQUES REPRESENTAN RELACIONES; LAS LINEAS DISCONTINUAS FIGURAN REFERENCIAS CRUZADAS.

Las referencias cruzadas no aportan nueva información. Si se suprimiera la línea discontinua entre la entidad proveedor y Orden de compra, no se perdería información en ninguna de ellas. Las líneas de referencia cruzada en el esquema indican uniones existentes que permiten localizar más fácilmente la información deseada.

### SUBESQUEMAS.

Esquema es el diagrama general de las entidades y atributos que forman nuestra base de datos. El subesquema se puede ver como los esquemas de la base de datos redefinidos de acuerdo a la visión de cada usuario, permitiendo una seguridad en la información ya que solo los registros tipo definidos en un subesquema pueden ser accedados y también ver la información desde muchos puntos de vista de acuerdo a las necesidades de los diferentes usuarios, sin que esto provoque redundancia en la información almacenada. En la figura III-2 se muestran los subesquemas correspondientes a dos usuarios.

SUBESQUEMA PARA EL PROGRAMADOR A

ARCHIVO MAESTRO DE ORDENES DE COMPRA.

NO. DE ORDEN	NO. DE PROV.	NOMB. DE PROVEEDOR	DIREC. DE PROVEEDOR	FECHA DE ENTREGA	FECHA ORDEN	TOTAL \$
--------------	--------------	--------------------	---------------------	------------------	-------------	----------

DETALLES ORDEN DE COMPRA.

NO. DE ARTICULO	NOMBRE DE ARTICULO	CANTIDAD PEDIDA	PRECIO
-----------------	--------------------	-----------------	--------

SUBESQUEMA PARA EL PROGRAMADOR B

ARTICULOS EN EXISTENCIA.

NO. DE ARTICULO	NOMBRE DE ARTICULO	CANTIDAD DISPONIBLE
-----------------	--------------------	---------------------

DETALLE PEDIDOS PENDIENTES

NO. DE ORDEN	NOMBRE DE PROVEEDOR	CANTIDAD PEDIDA	FECHA DE ORDEN	FECHA DE ENTREGA
--------------	---------------------	-----------------	----------------	------------------

FIG. III-2 DOS DIFERENTES PROGRAMADORES NECESITAN DIFERENTES VISIONES DE LA BASE. AQUI SE ILUSTRAN SUS RESPECTIVOS SUBESQUEMAS.

Los programadores de aplicaciones no tienen porque conocer la totalidad del esquema. En cambio, el administrador de datos debe asegurarse de que los subesquemas que los usuarios utilizan son derivables del esquema.

## CORRESPONDENCIA DE DATOS, SIMPLE Y COMPLEJA.

La relación que existe entre dos tipos de datos puede ser simple o compleja. Por simple queremos decir correspondencia biunívoca o unívoca, es decir de uno a uno y por compleja tenemos una relación de uno a muchos, como en el caso de un departamento con sus empleados, donde por cada departamento existen muchos empleados, las relaciones las representamos con la siguiente notación:

Número de empleado  $\longleftrightarrow$  Número de Depto.  
Correspondencia simple

Número de empleado  $\longleftrightarrow$  Número de Depto.  
Correspondencia compleja.

En la notación indicada la flecha de una sola punta indica relación única, con dos puntas indicamos relación múltiple como en el segundo caso.

### COMO DIBUJAR UN ESQUEMA.

Un esquema mal dibujado, mas que aclarar las cosas las complica, por lo que se han desarrollado un conjunto de reglas para elaborar un esquema y -- son las que damos a continuación:

- 1.- El diagrama debe distinguir claramente los nombres de las entidades y atributos.

- 2.- Debe quedar clara la unión de datos que forman un atributo.
- 3.- Debe aclararse la distinción entre atributos y entidades.
- 4.- El diagrama debe indicar claramente que relaciones son simples y cuales complejas.
- 5.- Las relaciones deben diferenciarse de las referencias cruzadas.
- 6.- No deben utilizarse nombres duplicados.

Por medio de un software de administración adecuado, la vista de los datos que se presenta al usuario se mantendrá independiente de la representación física, de modo que, como hemos dicho tantas veces, ésta y el hardware podrán alterarse sin afectar la descripción lógica de los datos que interesan al usuario, lo cual dicho de otra manera es que debemos procurar una manera de describir los datos que:

- 1.- Pueda ser entendida fácilmente por los usuarios que no tienen preparación previa como programadores.
- 2.- Que haga posible ampliar la Base de datos sin modificación de la estructura lógica existente, y por lo tanto sin modificación de los programas de aplicación.
- 3.- Que permita la máxima flexibilidad en la formulación de interrogantes de forma no



prevista, o espontánea, en las terminales.

### ARCHIVOS PLANOS.

La manera más común de asociar un valor con un agregado de datos y asociar éstos con atributos de entidad, consiste en almacenar juntos los agregados de datos en una secuencia fija como se ve en la figura III-3.

NOMBRE DEL ATRIBUTO	CLAVE PRIMARIA		CLAVES SECUNDARIAS					
	NUMERO DE EMPLEADO	NOMBRE	SEXO	CALIFICACION	FECHA	DEPARTAMENTO	CODIGO DE CAPACIDAD	SALARIO
	53730	JAINES ALBERTO	1	03	100335	044	73	2000
	28719	LARRONDO JOSE	1	05	101019	179	43	1800
REGISTRO SEGMENTO O TUPLE	53550	GOMEZ EMILIA	0	07	090938	044	02	1100
	79632	GUTIERREZ PEDRO	1	11	011132	090	11	5000
	15971	JUAREZ FERNANDO	1	13	021242	172	43	1700
	51883	PEREZ JUAN	1	03	091130	044	02	2000
	36453	BARRAGAN JAIME	1	08	110941	044	02	1800
	41618	ROSAS PATRICIA	0	07	071235	172	07	2500
	61903	PEREA VICTOR	1	11	011030	172	21	3700
	72921	TELLEZ CAROLINA	0	03	020442	090	93	2100

↑ IDENTIFICADOR DE ENTIDAD

CONJUNTO DE VALORES DE UN ITEM DE DATOS

ALGUNOS ATRIBUTOS SON IDENTIFICADORES DE ENTIDAD EN OTROS ARCHIVOS.

FIGURA III-3

A esta manera de asociar entidades y atributos se le llama Matriz plana o archivo plano.

Obsérvese que algunos de los valores de los -- atributos pueden ser nombres o identificadores de -- entidad en otros archivos, como se ve en la figura III-1, donde Número de proveedor es un atributo de la entidad Orden de compra y en seguida tenemos la entidad Proveedor donde Número de proveedor es el -- identificador de la entidad.

De la misma figura podemos decir que cada grupo de datos que forman una fila horizontal puede -- constituirse en un registro de entidad como en la -- figura III-3.

Debe hacerse notar que el registro almacenado puede diferir del registro que percibe el programador de aplicaciones, tal vez contenga más campos y éstos estén ordenados de distinta manera. Así mismo cabe hacer notar que el registro almacenado puede -- diferir en mucho del registro físico, ya que el registro físico suele contener muchos registros lógicos almacenados y ser compactados por medio de ciertas técnicas.

### BASES DE DATOS RELACIONALES.

A una tabla como la de la figura III-3 se le llama relación y a la base de datos construida por medio de relaciones se le llama Base de datos Relacional o sea que está construida con matrices planas de agregados de datos.

## NORMALIZACION

Se entiende por normalización la conversión de las matrices de agregados de datos no planas a matrices planas o archivos planos.

La manera más fácil de normalizar un archivo es eliminar -- los grupos repetitivos y formar con él una tabla o archivo plano debidamente nominado, cambiando las claves de entrada como se ve en la figura III-4.

ORDEN DE COMPRA

NO. DE ORDEN	NO. DE PROV.	FECHA ORDEN	FECHA ENT.	NO. DE ARTICULO	PRECIO	CANT.	TOTAL \$
--------------	--------------	-------------	------------	-----------------	--------	-------	----------

ORDEN DE COMPRA

NO. DE ORDEN	NO. DE PROV.	FECHA ORDEN	FECHA ENTREGA	TOTAL \$
--------------	--------------	-------------	---------------	----------

CLAVE

RENGLON DE COMPRA

# DE ORDEN	# DE ARTIC.	PRECIO	CANT.
------------	-------------	--------	-------

CLAVE

FIG. III-4 ELIMINACION DE UN GRUPO REPETITIVO POR PARTICION DEL ARCHIVO EN DOS RELACIONES.

Aparentemente y de la figura anterior aumenta la redundancia, pero no necesariamente es así, recuerdese que la normalización atañe solamente a la estructura lógica y no a la forma en que se encuentran en la máquina (física).

Debe tenerse muy en cuenta que los grupos repetivos se encuentren separados en el diseño original de la Base de datos, de otra manera se tendrá que hacer al crecer dicha base, trayendo problemas de profundas modificaciones a los programas de aplicación que serán sometidos a nuevas pruebas, subiendo considerablemente los costos de mantenimiento, gasto que se obviará con la normalización y adecuada independencia de datos en el software.

### TUPLES

El agrupamiento de agregados de datos observado en la figura III-3 representa una relación entre éstos. Se define entonces el tuple como: "Conjunto de valores relacionados de ésta u otra manera".

### CLAVES

Todo tuple debe estar asociado con una clave que permita su identificación. A veces el tuple se identifica mediante un único atributo. En otras ocasiones, no obstante, hay que recurrir a más de un atributo para lograr la identificación inequívoca de un tuple. La clave entonces debe cumplir con los dos siguientes requisitos:

- 1.- Identificación unívoca.- En cada tuple de una relación, el valor de la clave debe identificar unívocamente ese tuple.

2.- No redundancia.- Ningún atributo de la clave podrá ser descartado sin destruir la propiedad de identificación unívoca.

En cada tuple puede existir más de un conjunto de atributos capaz de satisfacer estos dos requisitos, cuando existe esta opción, la clave se elegirá de modo que ninguno de los atributos tengan un valor indefinido y que el número de atributos sea el mínimo.

El proceso de normalización ya descrito elimina los dominios no simples (tomando como dominio todos los atributos contenidos en un tuple), convirtiendo los datos a la forma de tablas bidimensionales. Es ésta la que llamamos primera forma normal. El proceso de normalización subsiguiente examina las relaciones de la primera forma normal y puede partir alguna de ellas aún en relaciones más simples. Este proceso consta de dos pasos; en el primero, se reducen los datos a la segunda forma normal, mientras que el segundo nos lleva a la tercera forma normal. Las ideas básicas en las que se apoyan estas nuevas formas normales son simples, pero las ramificaciones de esta normalización son muchas y muy sutiles y varían según el tipo de uso de la base.

## ESTRUCTURA DE DATOS.

La manera de almacenar datos en una computadora es a través de las llamadas estructuras de datos, de las que existen dos grandes grupos que son; Estructuras Estáticas tales como Arreglos, Registros, etc. y las llamadas Estructuras Dinámicas como son los Arboles, Listas, etc.

En una estructura de datos estática se trabaja con agrupaciones de datos que no van a cambiar, esto significa que es posible predecir la cantidad de memoria que se va a usar y que ésta no va a cambiar durante el proceso de los datos en la computadora. Por otro lado en las estructuras dinámicas puede aumentar o disminuir la cantidad de memoria, así como, puede cambiar la estructura misma de acuerdo a los requerimientos y condiciones que se presenten en un momento dado.

### ESTRUCTURAS ESTATICAS.

A las estructuras estáticas también se les llama básicas o fundamentales y están junto con elementos no estructurados pueden usarse para crear o generar estructuras dinámicas; dichas estructuras estáticas son las que se mencionan a continuación:

- ARREGLOS
- REGISTROS

## - ARCHIVOS

La utilización en su mayoría, de estructuras - dinámicas o avanzadas, nace de la necesidad de dar una estructura sin conocer aún la cantidad de memoria a usarse.

### ARREGLOS

Un arreglo se define como "Una estructura homogénea, en la cual sus componentes son del mismo tipo, al cual se le llama Tipo base". El arreglo también es definido como una estructura de acceso aleatorio, esto es; todos los componentes pueden ser seleccionados en forma aleatoria y son igualmente direccionables.

La definición de un arreglo del Tipo T por lo tanto especifica tanto el tipo base T como un Índice del tipo I.

TIPO T = ARREGLO (I) de  $T_0$

### REGISTROS.

Como vimos anteriormente es por medio de los registros como vamos a almacenar las propiedades de una entidad e identificar a la entidad misma. Por ejemplo: en el almacenamiento de la entidad Alumno con los atributos nombre, dirección, número de cuenta, teléfono y carrera; se usará un registro con el nombre "alumno" y se dividirá en cinco campos para

almacenar las propiedades que interesan de la entidad alumno.

De aquí podemos definir el registro como un -- "conjunto de datos que representan los atributos -- reunidos para formar entidades. De forma general se podría decir que es la unión de elementos estructurados entre sí".

Es conveniente tener una notación para definir los registros, misma que se hará de la siguiente -- forma:

```
TIPO T = REGISTRO S1 : T1
                    S2 : T2
                    S3 : T3
                    S4 : T4
```

Donde T es el nombre de la entidad que representa el registro; S<sub>1</sub>, S<sub>2</sub>, etc. son los nombres de los campos que representan a los atributos, y T<sub>1</sub>, T<sub>2</sub>, T<sub>3</sub>, etc. especifican el tipo de datos que pueden tomar cada uno de los campos. Por ejemplo, el registro alumno se define como:

```
TIPO ALUMNO = REGISTRO NOMBRE: ALFA
                    No. DE CUENTA: REAL
                    DIRECCION: ALFANUMERICO
                    TELEFONO: REAL
```



## CARRERA: REAL

Aquí vemos que el registro representa a la entidad " Alumno " y los cinco campos a los atributos nombre, No. de cuenta, dirección, teléfono y carrera.

Los tipos de datos se dividen en: Los tipos de datos elementales como son los números reales, enteros, alfanuméricos, etc., es decir los datos que no necesitan ser definidos para conocerlos y los tipos de datos que necesitan ser definidos en función de las entidades y atributos que representan para poderlos identificar en un registro. Y por último los tipos de datos que son constantes dentro del registro.

Para entender esto volvamos al ejemplo del registro "alumno". Existe un campo llamado dirección con tipo de datos alfanuméricos, lo que significa - que está formado por tipo de datos elementales. Se podría hablar por ejemplo de tipo de datos constantes si hubiera un campo que se tuviera que llenar - con una palabra específica como por ejemplo el sexo que sería llenado con la palabra masculino o femenino, existen otros campos que se tienen que llenar - como si fueran registros por ejemplo si en el registro "alumno" se tuviera un campo con el nombre fecha de nacimiento, éste se tendría que llenar como si fuera otro registro por medio de tipos elementa-

les de datos o constantes:

TIPO FECHA = CAMPO DIA: (1....30)  
 MES: (1....12)  
 AÑO: (1...2000)

En el ejemplo anterior el tipo de dato "fecha" está formado por tres subcampos que son día, mes y año, a los que llamaremos "agregado de datos".

Para referirnos a un campo específico ( $S_i$ ) dentro de un registro T usaremos la notación siguiente:

$T.S_1$

De esta manera si queremos designar el nombre de Pedro al registro del ejemplo anterior lo representaremos de la siguiente manera:

ALUMNO . NOMBRE := PEDRO.

Cada registro es usado para cada entidad de -- tal manera que si en el ejemplo tenemos 50 alumnos necesitaremos 50 registros para almacenarlos. Las - entidades del mismo tipo se agrupan y forman un con- junto de entidades o registros.

### ARCHIVOS.

La forma más sencilla de almacenar un conjunto de registros es de forma secuencial, es decir, un - registro tras otro como queda representado en la -- figura III-5.

NOMBRE	No. DE CUENTA	DIRECCION	TELEFONO	CARRERA
REGISTRO	1			
REGISTRO	2			
REGISTRO	3			
REGISTRO	4			

FIG. III-5 ARCHIVO.

Un almacenamiento secuencial que comúnmente es llamado archivo puede definirse como un número infinito de registros, almacenados uno tras otro y en el que para cada secuencia es posible construir una más larga.

Análogamente a la notación empleada para un registro, un archivo se definirá como:

TIPO T = ARCHIVO DE  $T_0$

Esta expresión define un archivo llamado T, compuesto por 0 o más registro  $T_0$ .

Se considera que un archivo tiene dos estados de operación: el de grabación y el de lectura. Estas operaciones solo pueden ser ejecutadas en un or

den determinado y solo un registro del archivo puede ser incorporado en cada operación.

La ventaja de un acceso secuencial es particularmente vista si los archivos son guardados en la memoria auxiliar, esto es, si la transferencia entre diferentes medios es involucrada.

### ESTRUCTURAS DINAMICAS

Cuando hablamos de una base de datos, no solo nos referimos a una serie de datos agrupados de acuerdo a las entidades y atributos escogidos; también hablamos de estos mismos datos interrelacionados entre sí, por lo que es necesario indicar las relaciones estructurales existentes entre los componentes de la base de datos que nos permitan una descripción lógica para una fácil y rápida búsqueda y recuperación de los datos deseados.

La manera de localizar un dato en la memoria de la computadora es por medio de direcciones, entonces las relaciones entre los registros se indicarán por medio de las direcciones de los mismos. Pero cuando almacenamos los datos no siempre es posible definir el tamaño de la memoria que vamos a usar, ni las direcciones de los registros; para resolver este problema se generan las direcciones de los registros cuando se requieren y se van relacio-

nando estos por medio de apuntadores.

### APUNTADOR.

Básicamente un apuntador es un campo dentro de un registro que indica la dirección del registro o registros que están relacionados con él. Los apuntadores son representados por medio de flechas como se indica en la figura III-6

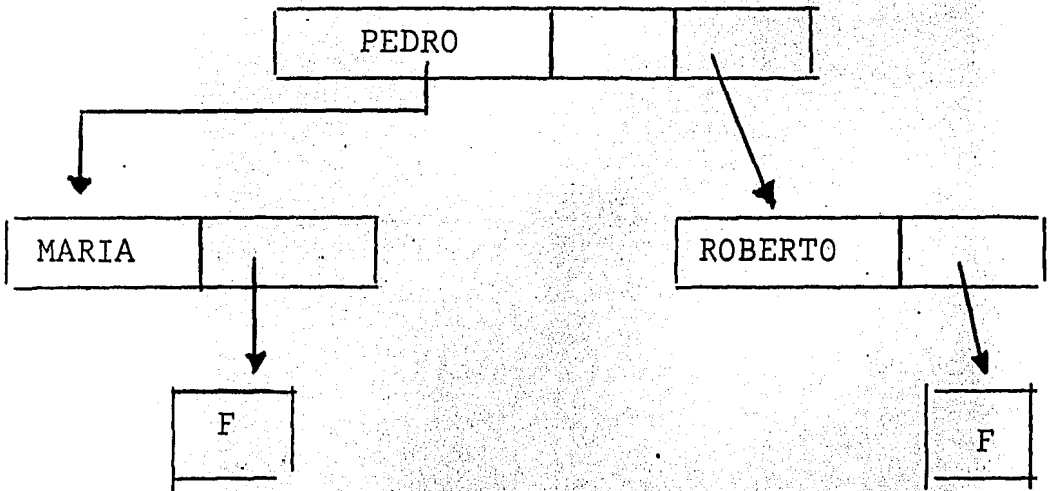


FIG. III-6 Relación de registros a través de apuntadores.

Cuando operamos con apuntadores existen dos -- funciones posibles que hay que especificar:

- 1.- Cuando nos referimos al registro que es señalado por el apuntador  $P \nearrow : T$ ; y que se puede leer como: T registro apuntado por P.

2.- Cuando nos referimos al apuntador mismo:

P :     ↗     T; y que se puede leer como P -  
apuntador del registro T.

### LISTAS LIGADAS

El almacenar los registros en un archivo tiene la desventaja de que si queremos cambiar el orden - del mismo, tenemos que reubicar los registros dentro del archivo lo que da como resultado una estructura poco flexible.

Para lograr una mayor flexibilidad en el almacenamiento secuencial, un registro debe contener, - además de la información, una liga (apuntador) con el siguiente registro que se desea relacionar secuencialmente, como aparece en la figura III-7.

DIRECCION	DATOS	APUNTADOR
A	REGISTRO 1	B
B	REGISTRO 2	C
C	REGISTRO 3	E
D	REGISTRO 4	NIL
E	REGISTRO 5	D

FIG. III-7 Archivo ligado a través de apuntadores.

En el caso anterior vemos como los registros - son ligados, ya no de acuerdo al orden inicial de -

almacenamiento, sino de acuerdo a los apuntadores, que contienen la dirección del registro al que hay que ir durante la búsqueda.

En el ejemplo anterior en el registro 4 no existe dirección alguna, sino simplemente la palabra -- "NIL" que indica el fin de la lista.

De esta forma encontramos un conjunto de elementos independientes alineados en forma de una lista, ligados por medio de apuntadores, en la que un apuntador inicial P nos indica el inicio de la lista. (figura III-8).

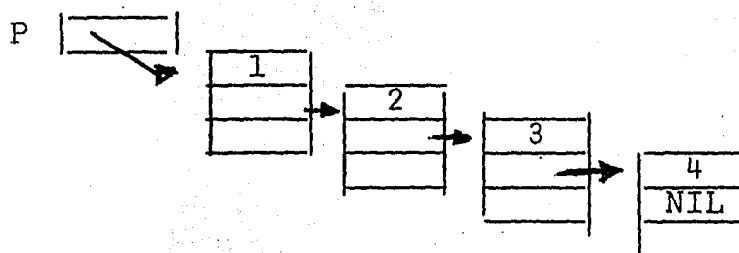


FIG. III-8 Archivo en forma de Lista Ligada.

Es así como tenemos la ventaja de reordenar la lista sin tener que reubicar cada uno de los registros. De esta manera con solo cambiar el valor de los apuntadores queda nuevamente ordenada la lista.

Otra ventaja más es que no existe el desperdicio de memoria como en el caso de los archivos ya que, por medio de las ligas, podemos aprovechar to-

dos los registros; además se puede tener varios ordenamientos de la lista de acuerdo a las necesidades de búsqueda con solo aumentar el número de apun-tadores en cada registro.

La definición de cada registro de la lista queda de la siguiente manera:

TIPO T = REGISTRO LLAVE ; ENTERA  
SIGUIENTE ; ↗ T.

Cada registro estará entonces formado por un campo, al que llamaremos "llave", cuyo propósito será almacenar la dirección que servirá para identificar al registro; otro campo para almacenar el apuntador y otra serie de campos para almacenar la información de los registros (que no están incluidos en ésta descripción).

Cuando queremos insertar un elemento al inicio de la lista tenemos primero que asignar un apuntador (q) al registro que queremos insertar y luego, simplemente, reasignaremos los apuntadores como sigue:

q ↗ . siguiente : = p ; p : = q.

q . siguiente : = p significa que el valor del apuntador "p", que apunta al primer registro de la lista es asignado al apuntador del registro que queremos insertar, quedando la lista anterior de la siguiente manera (figuras III-9 y III-10).



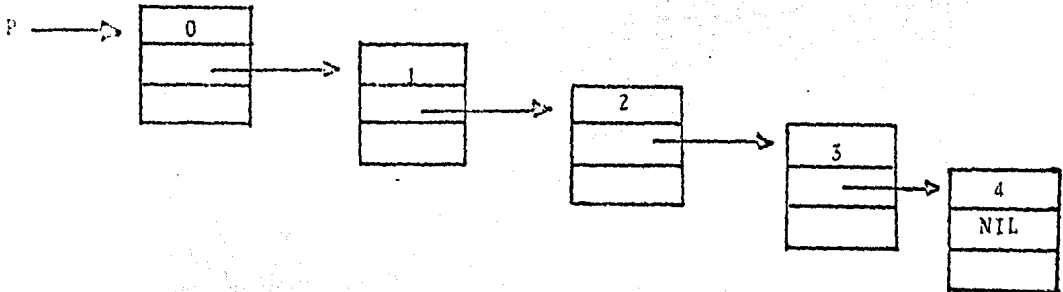


FIG. III-10

luego reasignaremos apunadores:  $p:=q$ . Quedando la inserción de la siguiente manera:

Para generar una lista, ejecutamos la operación de inserción de elementos al inicio de la lista descrita anteriormente, empezando cada lista vacía e insertando después cada uno de los elementos.

El proceso completo para una lista de "n" elementos es como sigue:

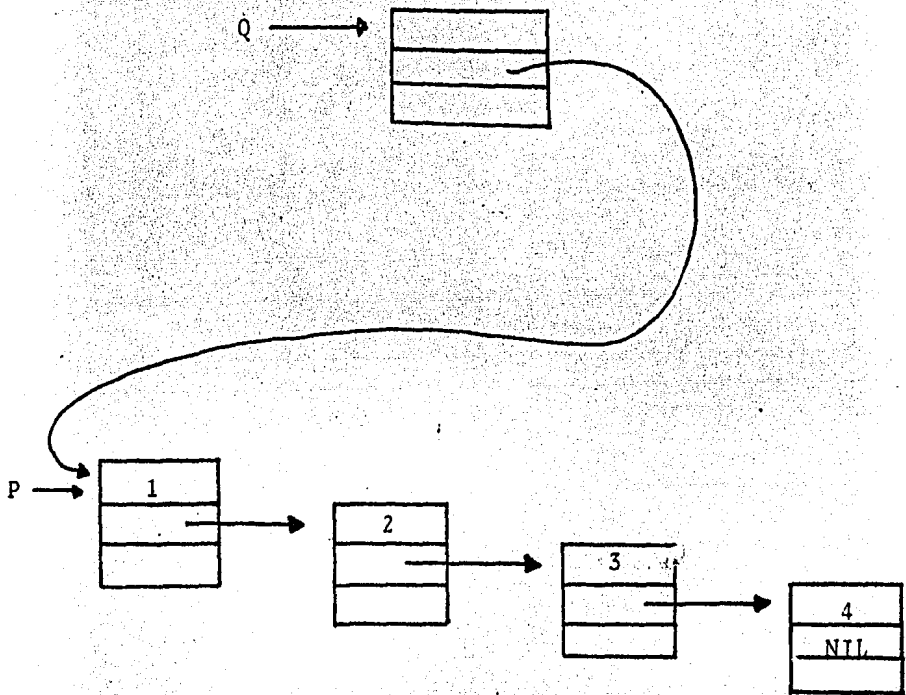
```

P = NIL      ( inicio de la lista )
SI N > 0
SI: Q ↗ . SIGUIENTE : = P ; P : = Q ;
   Q ↗ . LLAVE : = n ; n : = n-1
NO: FIN.
  
```

Cuando queremos insertar un elemento no al principio de la lista, sino después del registro apuntado por P, se hará de la siguiente manera: (figura III-11).

```

Q ↗ . SIGUIENTE : = P ↗ . SIGUIENTE
P . SIGUIENTE: =Q.
  
```



FIG, III-9

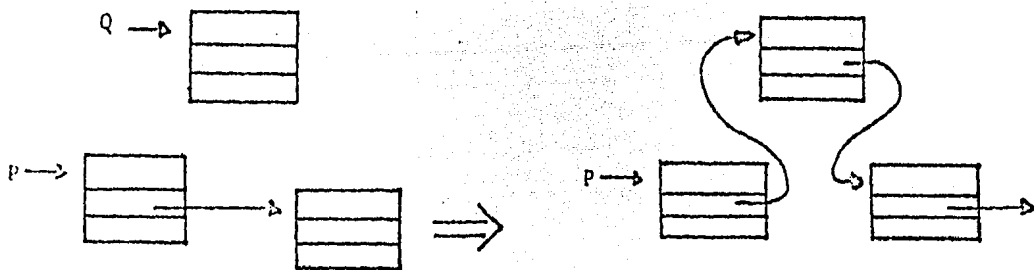


FIG. III-11

Si queremos eliminar el registro sucesor de P (elemento -- apuntado por P), se deja de la siguiente manera (figura III-12).

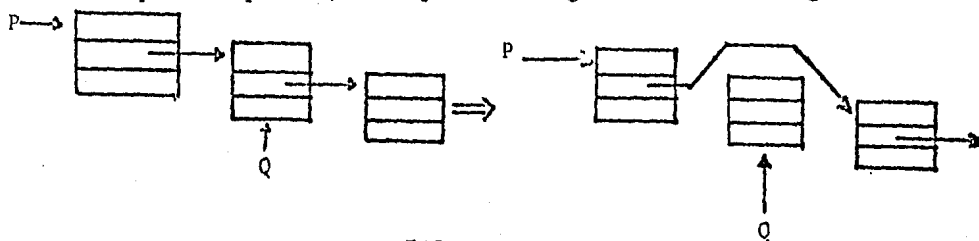


FIG. III-12

### LISTAS CIRCULARES.

Un ligero cambio en la manera de ligar los registros de una lista, es la lista circular. La lista circular tiene la característica de que su último registro en lugar de indicar el fin de la lista, es ligado al primer registro. Esto hace posible el empezar la búsqueda de la lista en cualquier punto y le da un mayor grado de simetría a la estructura. (Figura III-13).

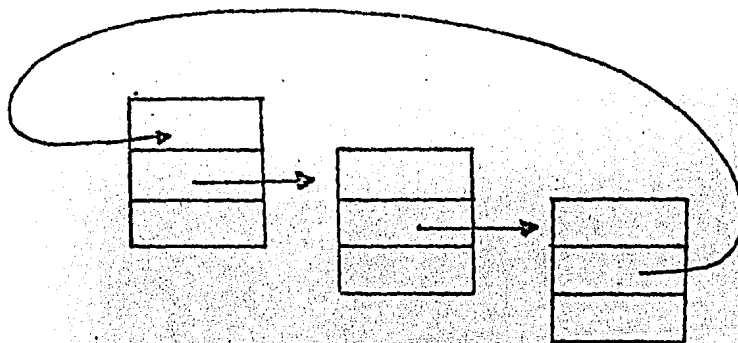


FIG. III-13 LISTA LIGADA CIRCULAR

### LISTAS DOBLEMENTE LIGADAS

Para una mayor flexibilidad en el manejo de las listas podemos incluir dos ligas en cada nudo, en lugar de una sola, que dando la lista de la siguiente manera: (Figura III-14).

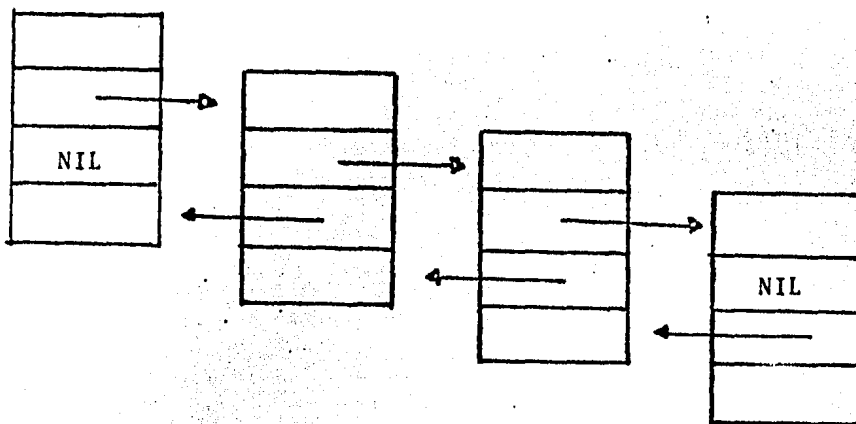


FIG. III-14 LISTA DOBLEMENTE LIGADA

En el caso de una lista doblemente ligada y circular existe una completa simetría en la búsqueda de un registro, ya que no importa en que punto de la lista se entre, siempre se podrá elegir un camino más corto.

### ARBOLES

Otra de las estructuras dinámicas que es utilizada en la organización de una base de datos es la estructura de árbol.

Un árbol es una serie de elementos llamados nudos, relacionados en forma jerárquica, que forman una serie de niveles y, en el que, el nivel más alto tiene un solo nudo al cual se le denomina "Raíz".

Definiendo formalmente un árbol podemos decir -- que:

"Es un conjunto finito, llamado  $T$ , de uno o más nudos tal que:

- a) Existe un nodo particular, llamado raíz del árbol, del que se extienden dos o más segmentos (en general).
- b) Los nudos restantes entran en una partición de " $m$ " conjuntos, disjuntos,  $T_1 \dots T_m$ ; -- tal que: cada uno de los  $T_i$  es a su vez un árbol. Los árboles  $T_1 \dots T_m$  se llaman subárboles de la raíz. Es decir, un subárbol es cualquier árbol que se puede obtener del árbol original mediante una "poda" cortando algunas de sus ramas.

De esta definición se pueden desprender las siguientes características:

- a) Con excepción de la raíz, todo nudo está -- vinculado a otro nudo de nivel superior al que llamaremos "Padre".
- b) Todo nudo de un árbol es a su vez raíz de -- un subárbol del árbol original, formado por uno o más elementos relacionados entre sí, -- a los que denominaremos "Hijos".
- c) Se define como grado de un nudo al número -- de subárboles que contenga.

- d) Se llamará "Hoja" a un nudo de grado cero, esto es, al nudo que no tenga hijos.
- e) Se llamará nudo "Ramal" a un nudo de grado mayor que cero.
- f) Se define el nivel de un nudo como: el nivel de la raíz igual a cero y el nivel de cualquier otro nudo igual al nivel de su antecesor directo (padre) más uno.
- g) Se define un "Bosque" como un conjunto de - cero o más árboles disjuntos.
- h) En algunos casos el orden de los subárboles  $T_1, T_2, \dots, T_m$  es importante, en cuyo caso T será un árbol ordenado.
- i) La altura de un árbol está dada por el número de niveles que contenga.

Por convención, la representación gráfica de los árboles es inversa al sentido en que se da su crecimiento: la raíz es su primer nudo y de él se desprenden hacia abajo todos los subárboles. (Fig. III-15).

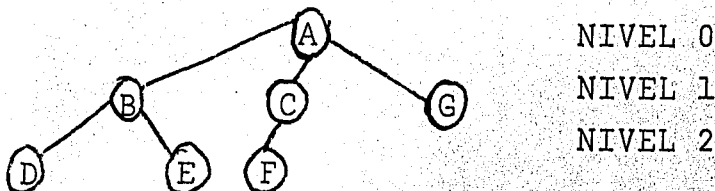


FIG. III-15 REPRESENTACION GRAFICA DE UN ARBOL

De la figura anterior podemos decir que:

- Es un árbol de tres niveles.
- A es el padre de B, C, y G.
- B es el padre de D y E.
- C es el padre de F.
- La raíz del árbol es A.
- Contiene la raíz, los subárboles B, C y G.
- Las hojas del árbol son D, E, F, y G.

Existen otras representaciones gráficas de los árboles, dependiendo de las características de las relaciones de orden, existentes en el conjunto de - nudos que definen un árbol como son:

- a) Diagramas de VENN (Fig. III-16).
- b) Anidación de paréntesis (Fig. III-17).
- c) Notación indentada por medio de espacios y márbolos (Fig. III-18)
- d) Notación decimal de Dewey (Fig. III-19).

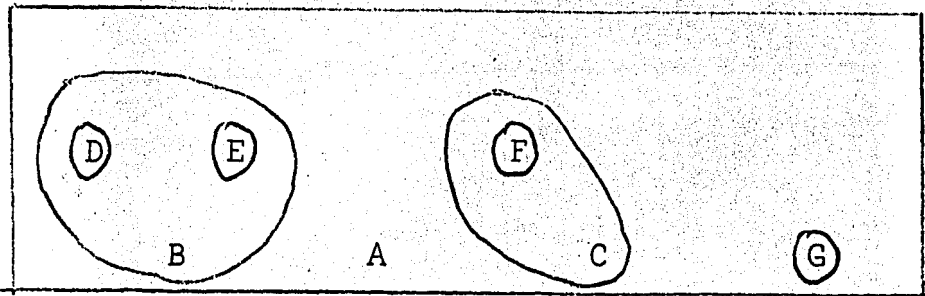


FIG. III-16 Representación gráfica de árboles por medio de Diagramas de Venn.

A ( B (D,E), C ( F ), G )

FIG. III-17 Representación de árboles por --  
Anidación de paréntesis.

A

B

D

E

C

F

G

FIG. III-18 Representación de árboles  
con Notación indentada.

1 (A), 1.1 (B), 1.1.1 (D), 1.1.2 (E) ,  
1.2 (C), 1.2.1 (F), 1.3 (G)

FIG. III-19 Representación de árboles con Nota--  
ción decimal de Dewey.

### ARBOLES BINARIOS.

Un árbol particularmente importante en computación es el árbol binario, por la razón de que cualquier tipo de árbol se puede convertir en árbol binario y además el recorrido de éstos es mucho más sencillo.

En éste tipo de árboles se permite a los nudos



tener a lo más dos subárboles y para recorrerlo es necesario distinguir si se trata del subárbol izquierdo o del subárbol derecho. Por lo tanto es un conjunto finito de nudos que: está vacío o bien consiste de un árbol de grado 2.

Para representar los árboles binarios en una computadora es evidente que no se declaran las variables como un árbol ya estructurado, sino que para tener mayor flexibilidad, se definen solo los nudos como registros de determinado tipo, en los que el grado del árbol está dado por el número de apuntadores que tengan los registros. Evidentemente la referencia con un árbol vacío es significada por la palabra NIL. De esta forma los nudos de un árbol binario serán definidos como:

```
TIPO NUDO  REGISTRO  Info : Alfa
                                     Izquierdo, Derecho :
                                     ↗ Nudo.
```

donde el campo Info engloba a la información contenida en ese registro, y los apuntadores Izquierdo y Derecho servirán para indicar la dirección a los -- subárboles descendientes de este nudo, construyendo se un árbol como el de la Figura III-20.

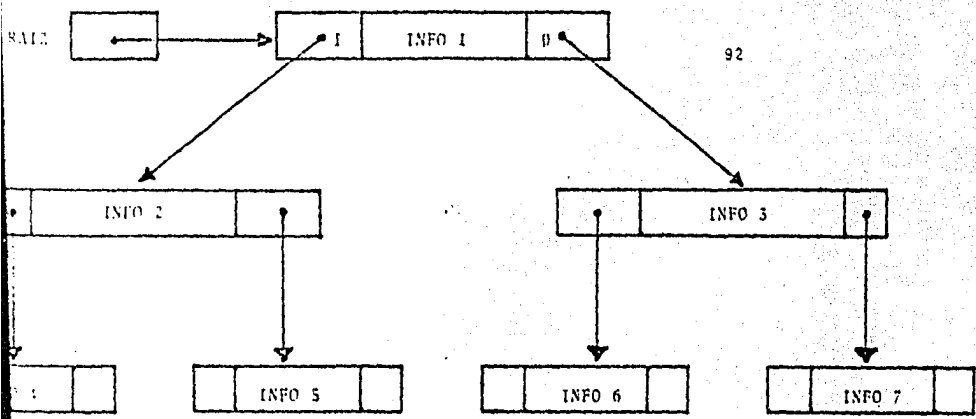


FIG. III-20 REPRESENTACION DE UN ARBOL BINARIO

Así vemos que el árbol irá creciendo de acuerdo con las -- ligas que se establezcan entre los nuevos registros a almacenar y las hojas.

Antes de hablar de las diferentes operaciones básicas en -- los árboles binarios veamos como se construyen éstos

Partiendo de la raíz los diferentes registros son ligados -- entre sí de acuerdo con la estructura de árbol binario, esto -- es, de la posibilidad de ligar dos registros con el registro -- anterior, y a cada uno de éstos dos registros nuevos ligar otros -- dos registros más, etc.

En orden a obtener la mínima altura posible en un número -- de registros dados, lo que facilitaría la búsqueda en el árbol. Se tienen que alojar el máximo número posible de registros en -- cada nivel, exceptuando el último. Esto se puede lograr distri -- buyendo los registros igualmente a la izquierda y derecha del -- registro antecesor; ésto significa que el árbol para "n" regis -- tros dados se irá construyendo de la siguiente manera: (Fig. -- III-21).

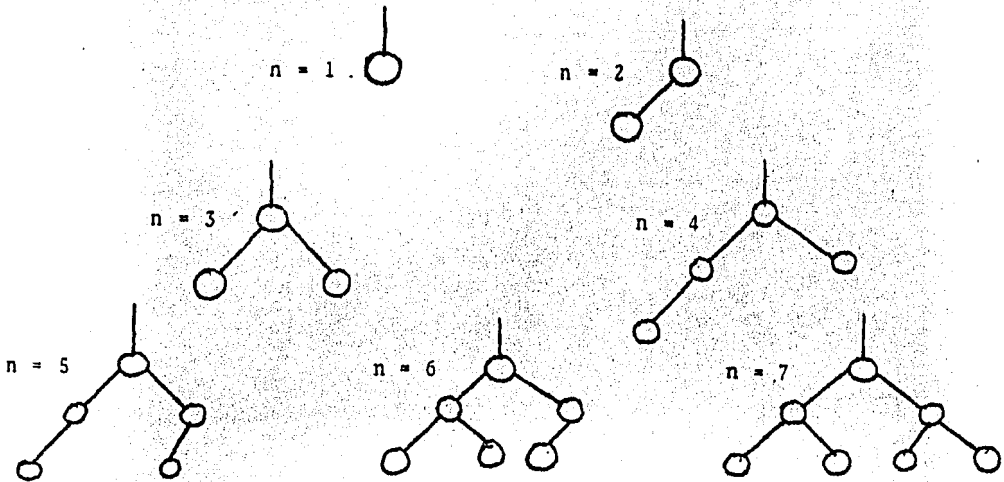


FIG. III-21 CONSTRUCCION DE UN ARBOL BINARIO

Las reglas para éste tipo de distribución serán:

1. Usar un registro para la raíz.
2. Generar el subárbol izquierdo con  $n_i = n \text{ div } 2$  registros.
3. Generar el subárbol derecho con  $n_d = n - n_i - 1$  registros.

A éste tipo de árbol se le llama árbol balanceado y se define como: "Un árbol está completamente balanceado si para cada nudo padre el número de nudos en su izquierda y derecha difieren cuanto más en uno".

#### RECORRIDO DE ARBOLES BINARIOS

Existen varias formas de recorrer un árbol binario. Aquí -- analizaremos las tres más importantes que son:

- a) Recorrido vertical (PREORDER)
- b) Recorrido simétrico (INORDER)
- c) Recorrido ascendente (POSTORDER)

La idea común es la de visitar cada nudo del árbol una sola vez, entendiendo por visitar, localizar un nudo para luego ejecutar una operación con él.

Recorrido vertical:

Si  $T = A$  el árbol se recorre sin hacer nada; si no:

- a) Se visita la raíz
- b) Se recorre el subárbol izquierdo
- c) Se recorrer el subárbol derecho.

Para la Fig. III-22 el orden del recorrido será: A, B, D, C, E, F.

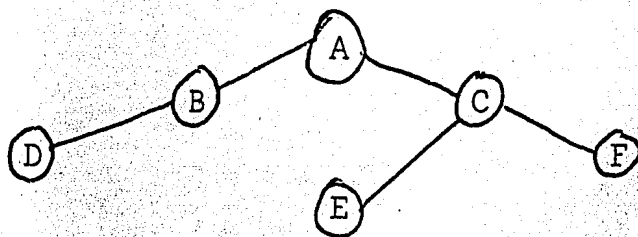


FIG. III-22

Recorrido simétrico:

Si  $T = A$  se visita solo la raíz del árbol, si no:

- a) Se recorrer el subárbol izquierdo.
- b) Se visita la raíz.
- c) Se recorrer el subárbol derecho.

Para la Fig. III-22, el recorrido será:  
D, B, A, E, C, F.

Recorrido ascendente:

Si  $T = A$  el árbol se recorre sin hacer nada; -  
si no:

- a) Se recorre el subárbol izquierdo.
- b) Se recorrer el subárbol derecho.
- c) Se visita la raíz.

Para la Fig. III-22, el recorrido será:  
D, B, E, F, C, A.

### ARBOLES DE MULTICAMINO.

Si analizamos las relaciones de parentesco de un árbol cualquiera, para un nudo P siempre podemos distinguir:

- Su padre.
- Sus hermanos
- Sus hijos.

Cuando un padre tiene más de dos hijos, entonces hablamos de un árbol de multicamino. En vista de la reciprocidad que existe entre las relaciones Padre - Hijo, podemos englobarla en una sola relación Hijo. Lo cual nos permite distinguir para cada nudo dos tipos de relaciones:

- Sus Hermanos.
- Sus hijos.

Pero para referirnos al Padre de un conjunto de hermanos basta con relacionar al padre con uno de ellos y ligar entre sí a los hermanos. Esta observación permite simplificar aún más nuestro sistema de decisiones y convertir un árbol de cualquier grado a un árbol binario, en consecuencia nuestro sistema de relaciones queda de la siguiente manera:

- Hermano siguiente.
- Primer Hijo.

Dado un nudo P de grado N como el de la Fig. III-23.

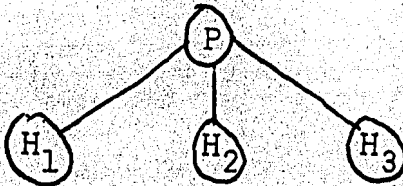
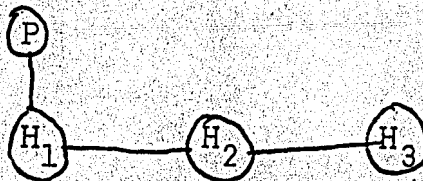


FIG. III-23

Lo podemos simplificar de la siguiente manera:



Esto implica que cualquier nudo puede tener sólo dos ligas:

- Un apuntador a su primer hijo
- Un apuntador al hermano siguiente.

Estamos, pues, en el caso de un árbol binario

en el cual se distinguen dos tipos de relaciones:

- Ligar izquierda.
- Liga derecha.

En donde lo único que tenemos que identificar:

Liga izquierda - primer hijo

Liga derecha - hermano siguiente.

Y de este modo tendremos la forma de representar cualquier árbol por medio de un árbol binario.

## TECNICAS DE DIRECCIONAMIENTO

### Introducción.

En los archivos lógicos, la identificación se hace por medio de un grupo de caracteres exclusivos al que se le llama CLAVE. Dicha clave es por lo general un campo de longitud fija que ocupa por lo general la misma posición relativa en todos los registros.

La clave puede estar formada por dos o más campos y entonces se habla de clave concatenada, una de éstas claves debe de ser exclusiva y es la que se usa para alojar o para localizar un registro en un archivo y se le da el nombre de clave prima o --identificador.

El problema básico del direccionamiento en un archivo es: "dada una clave prima, tal como un número de cuenta, ¿ Qué hace la computadora para localizar el registro correspondiente a esa clave.?" Este problema se soluciona usando ciertas técnicas que se han venido desarrollando y que son las siguientes:

1. Exploración o barrido del archivo.
2. Búsqueda por bloque.
3. Búsqueda binaria.
4. Archivos secuenciales indizados.



5. Archivos no secuenciales indizados.
6. Direccionamiento clave-igual-dirección.
7. Algoritmos para conversión de clave.
8. Desmenuzamiento (HASHING).

### TECNICA 1. EXPLORACION O BARRIDO DEL ARCHIVO.

Esta es la manera más simple de localizar un registro y consiste como su nombre lo indica en realizar un barrido del archivo inspeccionando la clave de cada registro, de ahí se deduce que es un método demasiado lento para los propósitos del procesamiento de datos, ésta técnica sólo se usa en operaciones de procesamiento en lotes basados en el empleo de archivos secuenciales.

En la siguiente figura podemos observar el diagrama de flujo que corresponde a ésta técnica:

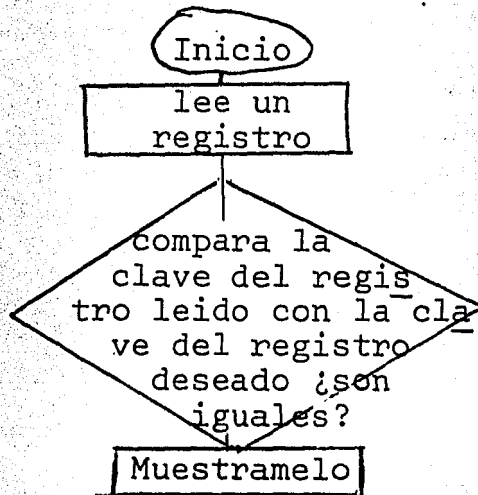


FIG. III-24 Diagrama de flujo de la técnica de Exploración o barrido del archivo.

## TECNICA 2. BUSQUEDA POR BLOQUE

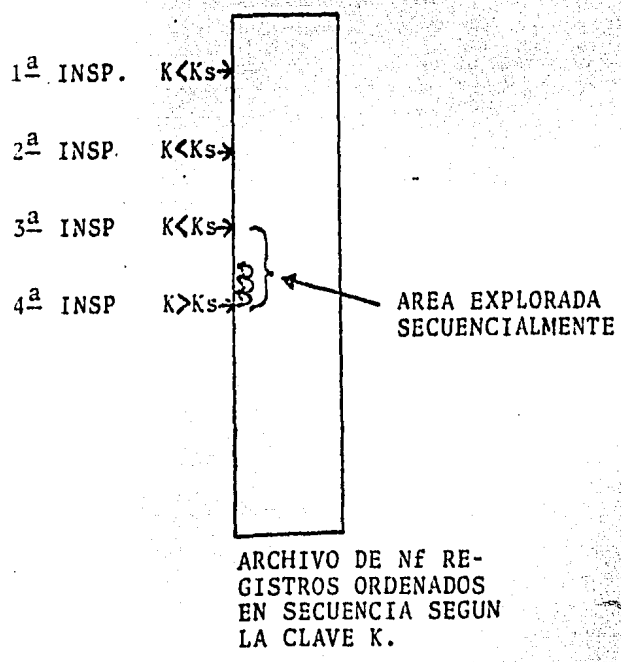
Cuando los registros están organizados secuencialmente de acuerdo con la clave, no es necesario leer todos los registros, se podría examinar por -- ejemplo, el primer registro de cada centena, ya al encontrar un registro cuya clave tiene un valor mayor que la que se busca, se exploran los 99 registros recién salteados.

Cuando hay que explorar un número total de --- ítems  $N_f$ , el tamaño óptimo de los bloques está dado por la siguiente formula:

$$N_b = \sqrt{N_f}$$

Por ejemplo en un archivo de 10000 registros, los bloques comprenderían 100 registros cada uno y habría que inspeccionar en promedio 100 registros para dar con el que se busca.

La figura III-25 muestra el diagrama de flujo correspondiente a ésta técnica:



$K_s$  = REGISTRO BUSCADO  
 $K$  = TOPE DEL BLOQUE  
 $x$  = TAMANO DE LOS BLOQUES

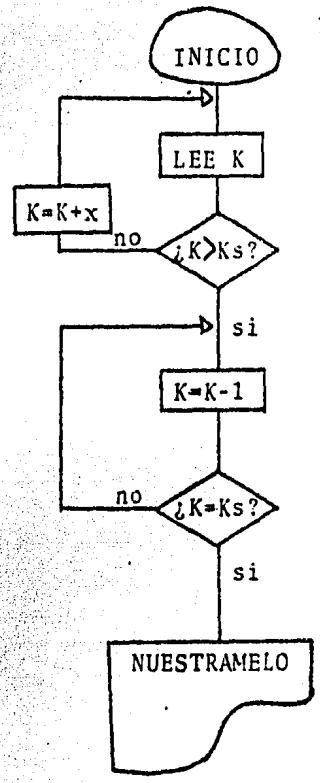


FIG. III-25 DIAGRAMA DE FLUJO DE LA TECNICA DE BUSQUEDA POR BLOQUE

TECNICA 3. BUSQUEDA BINARIA.

En una busqueda binaria se compara la clave del registro-buscado con la del registro que ocupa el centro del área que hay que explorar. De ésta manera según el resultado de la comparación, el área queda dividida en dos y el proceso se repite tantas veces como sea necesario. El número de comparaciones es aproximadamente igual a:

$$N_b = \log_2 \sqrt{N_f} - 1$$

Si  $N_f$  es grande, este número es menor que el de las  $N_f$  -- inspecciones que reclama el método por bloques.

Esta técnica es muy útil para explorar una memoria principal o un almacén de estado sólido, pero es inadecuado en el caso de dispositivos de almacenamiento de acceso directo y no sirve para la exploración de una cadena porque la computadora carece de medios para direccionar los registros.

La búsqueda binaria se prefiere más bien para la búsqueda de un índice de un archivo que la búsqueda en el archivo en sí. A continuación tenemos el diagrama de flujo de ésta técnica.

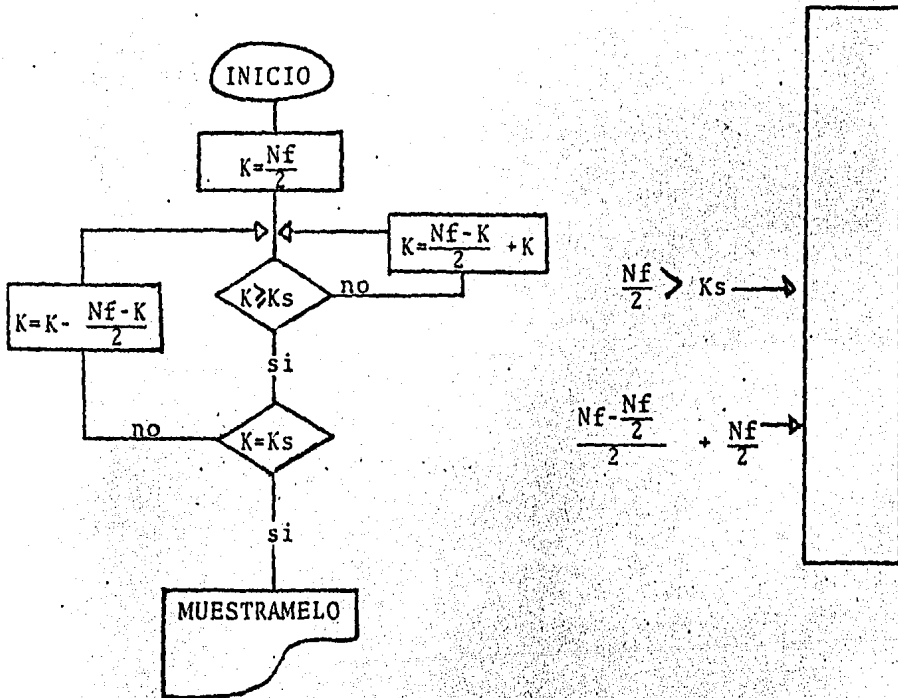


FIG III-26 DIAGRAMA DE FLUJO PARA LA TECNICA DE BUSQUEDA BINARIA.

#### TECNICA 4. ARCHIVOS SECUENCIALES INDIZADOS.

La exploración secuencial de los archivos con el objeto de localizar un ítem es un proceso que -- consume un tiempo excesivo pero es muy empleado en áreas reducidas, las que habrán sido localizadas -- apelando a alguna otra técnica. En el caso de una - pista de disco o tambor la exploración es viable -- porque se traslada con el tiempo de rotación.

Si el archivo obedece a la secuencia de la clave, el método usual de direccionamiento es el que - se basa en el empleo de una tabla llamada índice. La entrada a la tabla es la clave del registro buscado, el resultado del examen de la tabla es la dirección relativa o la dirección verdadera del registro en el archivo. La definición de índice entonces sería:

"Una tabla que opera con un procedimiento que acepta información acerca de ciertos valores de atributo, como entrada, y, provee como salida una información que permite la rápida localización del registro a que pertenecen aquellos valores de atributo".

Tenemos dos tipos de índices, el primario que utiliza como entrada una clave prima y el secundario que utiliza una clave secundaria.

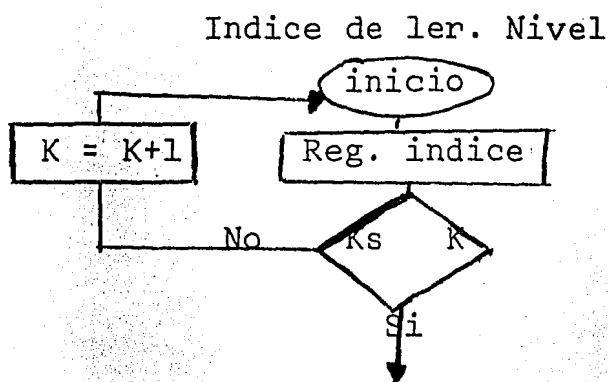
Cuando se usa un índice para el direccionamiento de un archivo, la computadora debe explorar el -

índice más bien que el archivo en sí. Con éste método se economiza tiempo a costa del espacio que necesitamos para almacenar el índice.

Si el archivo es secuencial, no es en general necesario que el índice incluya todos los registros, sino, más bien referencias a bloques de registros, los que luego se exploran para dar fin a la búsqueda.

En el caso de archivos muy grandes los índices resultan demasiado extensos y se recurre entonces a un índice de índices y no es raro recurrir a índices de más de dos niveles.

La organización secuencial junto con el índice constituye la forma de direccionamiento más común. A continuación podemos observar el diagrama de flujo correspondiente:



Las otras inspecciones se realizan con las técnicas Binaria o Secuencial.

FIG. III-27. Diagrama de flujo de la Técnica 4.

## TECNICA 5. ARCHIVOS NO SECUENCIALES INDIZADOS.

También es posible indicar los archivos no secuenciales, pero se necesita una entrada para cada registro en vez de una por cada bloque de registros lo que lo hace más voluminoso, además debe incluir las direcciones completas.

En comparación el archivo secuencial indizado es más ventajoso por el espacio ocupado y el tiempo utilizado para explorarlo, sin embargo este tipo de técnica se usa cuando un archivo es muy volátil ya que resulta muy larga y costosa la inserción y eliminación de registros en el archivo ordenado secuencialmente.

## TECNICA 6. DIRECCIONAMIENTO CLAVE-IGUAL-DIRECCION.

Existen varios métodos para la conversión directa del valor de una clave en una dirección de archivo.

Cuando cualquiera de ellos es viable, se tiene el método de direccionamiento más rápido, sin necesidad de recurrir a operaciones de exploración en el índice o en el archivo.

La manera más simple de resolver el problema del direccionamiento consiste en incorporar a la transacción de entrada la dirección de máquina rela

tiva del registro en cuestión.

Un ejemplo de ésta técnica se lleva a cabo en aplicaciones bancarias, donde los números de cuenta se modifican de modo que el número de cuenta o parte de él, coincida con las correspondientes direcciones de archivo de los registros. La dirección es igual a la clave o puede derivarse fácilmente de ésta.

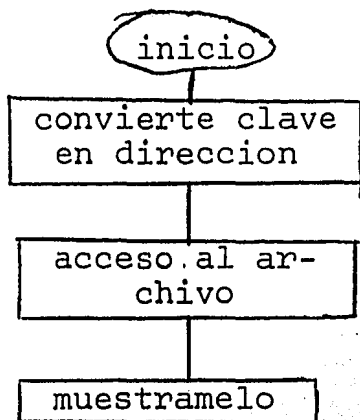
#### TECNICA 7. USO DE ALGORITMOS PARA CONVERSION DE CLAVE.

El empleo de un algoritmo para convertir la clave en una dirección constituye un método de direccionamiento casi tan rápido como el de clave - igual - dirección. En algunas aplicaciones es posible computar la dirección a partir de los identificadores de entidad, por ejemplo el nombre de una calle y el número de vivienda; éste método no es aplicable sino excepcionalmente pero es sencillo y rápido. Se le encuentra más comúnmente en los sistemas interactivos, donde es crítico el tiempo de referencia de los archivos.

Un inconveniente de los esquemas de direccionamiento directo es el de su inflexibilidad. Las direcciones de máquina de los registros suelen cambiar porque se amplían los archivos o porque los re



gistros se mudan a otras unidades, se consolidan o se modifican. Para mejorar la flexibilidad, el direccionamiento directo se hace por lo común en dos pasos. El primero convierte la clave en un número ordinal, el segundo, convierte éste número ordinal en la dirección de máquina. Los números ordinales son tales que resulta posible modificar sin problema las direcciones de máquina de los registros. En la figura III-28 podemos observar el diagrama de flujo correspondiente a las técnicas 6 y 7.



La clave es única por lo que no tiene problema pues la clave se convierte en la dirección por medio de un algoritmo.

FIG. III-28 Diagrama de flujo para las técnicas 6 y 7.

#### TECNICA 8. DESMENUZAMIENTO ( "HASHING" ).

Una útil e ingeniosa forma de la técnica de cálculo de direcciones es la que se denomina desmenuzamiento ( Hashing ), (en inglés se usan también los términos RANDOMIZING y SCRAMBLING). Con ésta técnica la clave del registro se convierte en un nú

mero casi aleatorio que sirve después para determinar donde se almacena el ítem.

Esta técnica es más económica, cuando mediante ella se individualiza un área en la que se almacena un grupo de registros, área que se llama cubo o balde ( a veces bolsillo o ranura ). El número de registros lógicos de que es capaz ésta área es la capacidad de el cubo.

Al cargar inicialmente el archivo, la ubicación de los registros se determina del siguiente modo:

- a). La clave del registro se convierte en un número casi aleatorio comprendido entre 1 y N, donde N es el número de cubos de que se dispone. Hay algoritmos para ejecutar esta operación y es preciso elegir alguno que sea adecuado para el conjunto de claves de los registros en cuestión.
- b). El número casi aleatorio n se convierte en la dirección del n-simo cubo y se copia al registro físico contenido en ese cubo.
- c). Si queda espacio disponible en el cubo, se registra allí el ítem lógico.
- d). Si el cubo ya está lleno, es preciso almacenar el registro en un cubo de desborde. Este puede ser el cubo siguiente o encontrarse en un área

aparte señalada por medio de un puntero.

Cuando se trata de leer un registro ya almacenado, el procedimiento es similar, es decir:

- a). La clave del registro buscado se convierte en el número casi aleatorio  $n$ , usando el mismo algoritmo que antes.
- b). El número casi aleatorio  $n$  se convierte en la dirección del  $n$ -simo cubo y se copia el registro físico aquí almacenado.
- c). Se inspecciona el cubo en busca del registro lógico del caso.
- d). Si el registro lógico no está en el cubo, se copia e inspecciona el cubo de desborde. En ocasiones, es preciso leer más de un cubo de desborde antes de hallar el registro de interés.

A causa de la naturaleza aleatoria del algoritmo ésta técnica no conduce al llenado completo del archivo; no obstante, en la mayoría de los casos, es fácil alcanzar una razón de llenado de entre el 80 y 90% con la ventaja adicional de que no se precisan índices. La mayor parte de los registros se encontrarán con una sola inspección de cubo, pero algunos exigirán que se inspeccione también el cubo de desborde y, en ocasiones hasta un tercero o un cuarto cubo de desborde.

### COMBINACION DE TECNICAS.

En ciertos casos, se recurre a una combinación de las técnicas anteriores para direccionar los registros de un archivo. Utilizase por ejemplo un índice para localizar un área de un archivo y ésta -- área se explora o se somete a una búsqueda binaria. A su vez un algoritmo de direccionamiento directo puede localizar una sección del índice, evitando la necesidad de recorrerlo en su totalidad.

### INSERCIONES Y ELIMINACIONES.

La organización de un archivo secuencial indizado sería cosa sencilla si no hubiera necesidad de agregar registros. La manipulación de las inserciones y eliminaciones es lo que da lugar a tanta variedad de técnicas de indización.

En el archivo tradicional de cinta magnética, la inserción y la eliminación no constituyen un problema, pero en el caso de un archivo de acceso directo, sería posible demorar las inserciones hasta el fin del día o el fin de mes, pero cuando el archivo es extenso y complejo se toma mucho tiempo para reescribirlo en el caso de tener múltiples punteros, cadenas o índices secundarios.

Cuando las inserciones se tienen que realizar

en tiempo real es recomendable evitar el uso de archivos secuenciales.

En el caso del desmenuzamiento no tenemos problema con la inserción y eliminación ya que estos se pueden hacer en tiempo real.

### CONSIDERACIONES SOBRE EL HARDWARE.

Desde el punto de vista hardware se puede adecuar la organización secuencial indizada a sus conveniencias. Si se hace de esta manera se abrevia el tiempo de acceso medio y se economiza espacio de al macen, pero también, si se cambiará el hardware o el archivo se mudara a otro tipo de dispositivo, se tendría un grave problema y entonces sería conveniente tenerlo independiente del hardware.

## DESMENUZAMIENTO ( "HASHING" ).

### INTRODUCCION

Debido a la gran aceptación que ha tenido ésta técnica desde su aparición a mediados del decenio - 1950 - 1960 con el término de "RANDOMIZING" (aleatorización) lo vamos a estudiar un poco más a fondo.

Muchos analistas de sistemas han evitado y siguen evitando el uso de ésta técnica por creer que es un método complicado pero en realidad es fácil de usar y ofrece dos ventajas sobre la indización, - en primer lugar, encuentra la mayoría de los registros con una sola operación de búsqueda localizada y, en segundo la inserción y eliminación de registros no presenta problemas complejos como ya vimos anteriormente.

### PARAMETROS QUE AFECTAN LA EFICIENCIA

Para ésta técnica los parametros que se deben considerar son los siguientes:

- a). El tamaño de los cubos.
- b). La densidad de empaquetamiento esto es, el número de cubos para un archivo de tamaño dado.
- c). La transformada clave-dirección por desmenuzamiento.
- d). El método para tratar los desbordes.

a). TAMAÑO DE LOS CUBOS

El cubo según lo decida el analista tiene cabida para un registro o más. La rutina de desmenuzamiento disemina los registros en los cubos más o menos de la misma manera que la rueda de una ruleta aloja las bolillas sucesivamente en distintas particiones.

Supongamos una ruleta donde cada bolilla representa un registro y cada partición un cubo, si alguna bolilla cae en una partición que ya está llena, hay que retirarla de la rueda y colocarla en un área de desborde, si variamos la capacidad de las particiones y el número de éstas, se varía el número de desbordes como se muestra en la siguiente figura, resultado de un cálculo estadístico básico.

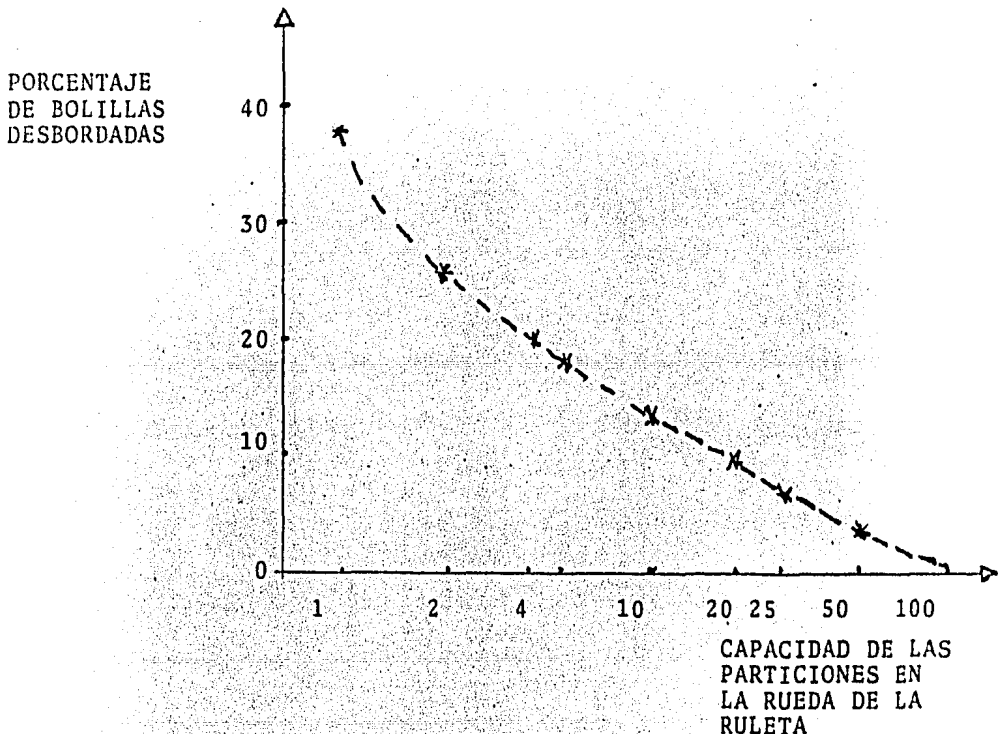


FIG. III-29

b). DENSIDAD DE EMPAQUETAMIENTO.

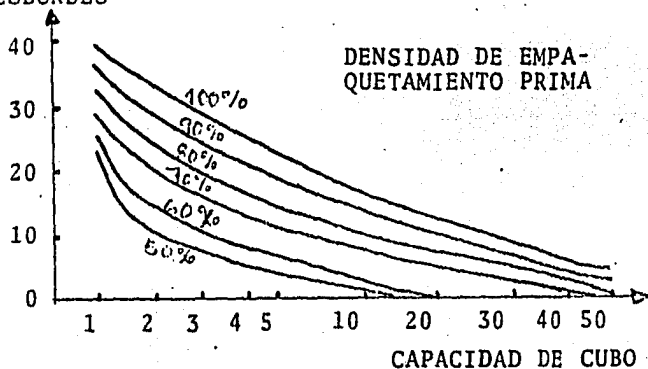
Basados en la idea de que a mayor número de registros mayor número de desbordes, se define entonces la densidad de empaquetamiento.

$$\text{Dens. de Emp.} = \frac{\text{Número de registros distribuidos en los cubos}}{\text{Capacidad total de los cubos}}$$

Por ejemplo; en la ruleta, si en vez de usar 100 bolillas - capacidad total de la rueda, usamos 80 la densidad de empaquetamiento sería de 80%.

El analista de sistemas debería concertar la economía de espacio con la economía de tiempo, ajustando al efecto la densidad de empaquetamiento prima, si el algoritmo de conversión clave-dirección diseminara los registros en los cubos al azar, como lo hace la rueda de ruleta, podríamos calcular estadísticamente el porcentaje de desborde para distintas densidades de empaquetamiento prima. Este cálculo se hace y sus resultados se grafican en las siguientes figuras:

PORCENTAJE DE  
DESBORDES





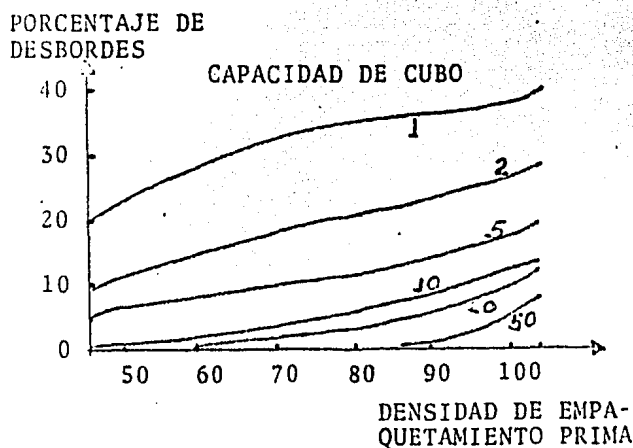


FIG. III-30 GRAFICAS TRAZADAS A PARTIR DE UNA TRANSFORMADA CLAVE-DIRECCION QUE DISTRIBUYE IDEALMENTE ALEATORIA EL CONJUNTO DE CLAVES, COMO UNA RUEDA DE RULETA

Para elegir el tamaño de los cubos y la densidad de empaquetamiento se deben tomar en cuenta entre otras cosas, el costo de la memoria, el tiempo de acceso y la eficiencia en el uso del espacio de almacenamiento.

#### c). ALGORITMOS DE CONVERSION CLAVE-DIRECCION.

El algoritmo de conversión clave-dirección consta por lo general de tres pasos que se mencionan a continuación.

- a). Si la clave no es numérica, convertirla a esta forma sin pérdida de información.
- b). Convertir las claves en una gama de números del mismo orden de magnitud que los números de dirección requeridos y distribuirlos tan homogéneamente como sea posible en esta gama de valores de dirección.
- c). Los números resultantes se multiplican por una constante para el fin de que todos entren en la gama exacta de direcciones.

Son muchas las transformadas propuestas y ensayadas para el segundo paso de las cuales tenemos:

- 1.- Método del centro de los cuadrados.
- 2.- División.
- 3.- Desplazamiento.
- 4.- Plegado.
- 5.- Análisis de dígitos.
- 6.- Conversión de raíz.
- 7.- Método de Lin.
- 8.- División Polinómica.

#### Método del Centro de los Cuadrados.

La clave se multiplica por sí misma y, tomando los dígitos centrales del cuadrado, se les ajusta - para que entren en la gama de direcciones.

Por ejemplo si se tiene una clave de 6 dígitos y se usan 7000 cubos, se tendrán que tomar los cuatro dígitos centrales del cuadrado de 12 dígitos y multiplicarlos por un factor de 0.7 y el resultado quedará entonces dentro de la gama de direcciones - de los cubos.

#### División:

En éste método la clave se divide por un número aproximadamente igual al número de direcciones - disponibles y como dirección del cubo se toma el residuo de la división. Como divisor se usa un número

primo o uno que no incluya factores primos pequeños.

Siguiendo con el ejemplo anterior si tuviéramos la clave 172148 la podríamos dividir entre 6997 y nos quedaría como residuo el número 4220, el cual sería la dirección del cubo.

### Desplazamiento.

En éste método, la clave se divide en dos partes sumando ambas y multiplicando el resultado por un factor de ajuste nos da la dirección del cubo. - Para el mismo ejemplo, la clave deberá tener 8 dígitos por lo que debemos aumentar los ceros necesarios a la izquierda para poder obtener la dirección adecuada con lo cual la clave será 00172148, dividiéndola en dos partes los sumandos serían; 0017 y 2148 que ya sumados nos darán 2165 y multiplicado por el factor 0.7 nos dan la dirección del cubo --- 1515.

### Plegado:

En éste método los dígitos externos de la clave se pliegan como si se tratase de una hoja de papel, después éstos dígitos se suman y se ajustan como en el caso anterior.

Debido a que el método de plegado tiende a ser más adecuado para claves largas, hemos seleccionado

para nuestro ejemplo una clave de 8 dígitos la cual es 17207358 y la dirección se formaría como lo vemos en la siguiente figura.

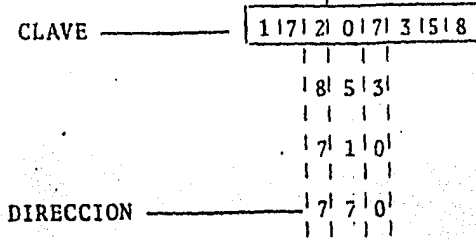


FIG. III-31 CONVERSION DE CLAVE A DIRECCION POR PLEGADO

#### Análisis de dígitos.

Algunos intentos para lograr una distribución más homogénea de las direcciones de cubo se basan en el análisis de la distribución de los valores de cada dígito o carácter de la clave. Las posiciones que revelan cierta tendencia en la distribución se eliminan de la clave en la esperanza de que cualquier transformada que se aplique a los restantes tendrá mayor probabilidad de dar una distribución homogénea.

#### Conversión de raíz.

En éste método se toma un número cualquiera que se la llame raíz o base y es multiplicado por cada uno de los dígitos de la clave elevado dicho número a la posición que ocupe el dígito dentro de la clave menos uno, tomada la posición de derecha a izquierda, y así vemos que la clave 172748 se transforma de la siguiente manera; tomando como raíz el número 11.

$$1 \times 11^5 + 7 \times 11^4 + 2 \times 11^3 + 7 \times 11^2 + 4 \times 11^1 + 8 = 267099$$

Siguiendo con nuestro ejemplo eliminamos los dígitos de más alto orden y dejamos unicamente los cuatro dígitos de más bajo orden los que vamos a multiplicar por nuestro factor de ajuste de 0.7 para obligarlo a entrar en nuestra gama de direcciones, -

quedando:

$$7099 \times 0.7 = 4969 \quad \text{Dirección.}$$

La conversión a raíz 11 es la más comunmente usada debido a que es rápidamente ejecutada por la computadora a través de una serie de desplazamientos y sumas.

### Método de Lin.

En este método la clave se toma como número hexadecimal y se convierte a binario para a su vez -- convertirlo a octal. Este resultado se divide por -- una constante  $q^m$  donde "q" es un número primo o un número que no contiene factores primos pequeños y -- "m" es un número entero positivo y al igual que el método de División el residuo es el que se usa para obtener la dirección relativa del cubo.

Para ilustrar este método usaremos la misma -- clave 172148 la que primeramente convertiremos a -- forma binaria quedando:

0001 0111 0010 0001 0100 1000

la cual la agrupamos de a tres bits para posteriormente encontrar su equivalente en octal;

000 101 110 010 000 101 001 000 = 05620510

Este resultado ahora se toma como número decimal y se le divide por la constante " $q^m$ " para obtener la dirección relativa del cubo.

### División Polinómica.

Cada dígito de la clave se toma como coeficiente de un polinomio; así, la clave 172148 se contempla como:

$$x^5 + 7x^4 + 2x^3 + x^2 + 4x + 8$$

El polinomio así obtenido se divide por otro polinomio fijo. Los coeficientes del resto forman la base de la dirección relativa del cubo.

### ELECCION DE LA TRANSFORMADA.

La mejor manera de elegir una transformada consiste en tomar una muestra del conjunto de claves en cuestión y probar con los diferentes métodos posibles, tomando en cuenta que la transformada ideal no es la que distribuye el conjunto de claves al azar sino que es aquella que los distribuye tan uniformemente como sea posible en el espacio de direcciones evitando los desbordes.

Debido a muchos experimentos realizados se podría decir que el mejor método para convertir claves en direcciones es el método de División o al menos es mejor que los demás.

### METODOS PARA TRATAR LOS DESBORDES.

Existen dos métodos básicos para tratar los --

desbordes que son el de almacenarlos en un área --- aparte o el de almacenarlos en el área primaria, -- aquí analizaremos dos técnicas que son las más comu nes:

-Encadenamiento de los desbordes.

-Espacio de desborde distribuido.

La primera técnica es muy simple, ya que cuando un cubo se encuentra lleno y le es asignado otro registro, éste registro se guarda en un cubo de desborde localizado por supuesto en el área de desborde, la dirección del cubo de desborde se guarda entonces en el cubo asignado. En el caso de que le -- sea asignado otro registro al cubo ya saturado, éste registro será guardado en otro cubo de desborde si es que el primero ya se encuentra lleno, en este caso la dirección del último cubo de desborde se -- guarda en el anterior formándose una cadena y de -- ahí el nombre de la técnica.

Con la segunda técnica, los cubos de desborde se encuentran distribuidos a intervalos regulares -- entre los cubos del área primaria cuando un cubo -- asignado se satura, se ubica al registro desbordado en el cubo de desborde más cercano, en el caso de -- que el cubo de desborde más cercano también se satu re el registro se ubicará en el siguiente cubo de -- desborde. Esta técnica ofrece una ventaja sobre la

primera y es la de ahorrar tiempo de desplazamiento del brazo cuando el archivo se encuentra en disco - ya que entonces el cubo de desborde estará por lo general en la misma pista o en la siguiente.

### TECNICAS DE INDIZACION

En el caso de usar un archivo indizado se da por un hecho que el archivo está secuencialmente ordenado de otra manera se necesitaría una entrada -- por cada registro del archivo. Ya que el índice es un archivo, su organización debe encararse como el de la organización de cualquier otro archivo de datos, la diferencia consiste en que el índice es un archivo simple y ofrece más posibilidades para la aplicación de las técnicas de computación y optimización. El índice al igual que el archivo principal debe admitir adiciones y eliminaciones y no consumir un tiempo excesivo.

Para la inspección de un índice son viables todas las técnicas de direccionamiento que hemos estudiado con algunos pequeños cambios y algunas otras que veremos a continuación:

1. Exploración o barrido del archivo.
2. Inspección por bloques.
3. Inspección binaria.
4. Inspección por árbol binario.
5. Índice de árbol balanceado con exploración se--



cuencial de los nodos.

6. Índice de árbol balanceado con inspección por bloques de los nodos.
7. Índice en árbol balanceado con inspección binaria de los nodos.
8. Operaciones de conjuntos de secuencias.
9. Almacenes intermedios de mirada lateral.
10. Árboles no balanceados.
11. Direccionamiento por estimación calculada.
12. Indización algorítmica.
13. Indización por desmenuzamiento.

Ahora analizaremos algunas de éstas técnicas para poder elegir la más adecuada para el presente trabajo.

### 1. Exploración o barrido.

Se examinan secuencialmente todos los registros del índice hasta encontrar el que se busca. Esta técnica es poco recomendable en índices grandes suponiendo que los registros tengan la misma probabilidad de ser buscados ya que el número de sondajes sería dado por la siguiente fórmula:

$$E ( N_p ) = ( N_1 + 1 ) - 2$$

Esta técnica es muy poco usada cuando los archivos se encuentran en disco o memoria, solo se usa cuando el archivo se encuentra en cinta donde -

de todas maneras se tienen que leer todos los registros.

## 2. Inspección por bloques.

Es la misma técnica usada para explorar los archivos no indizados pero secuenciales que ya vimos y el número de sondajes está dado por:

$$E ( N_p ) = \sqrt{N_1}$$

Esta técnica difícilmente se aplica para la exploración completa de un índice, pero es muy valiosa para inspecciones parciales. Por ejemplo, una porción de índice puede ocupar una pista en una unidad de almacenamiento (disco), se copia la pista (previamente localizada) de la unidad de almacenamiento a memoria y allí se inspecciona según la técnica de los bloques.

En el caso de archivos mucho muy grandes es necesario, en ocasiones tener varios niveles de índices, es en estas ocasiones cuando es muy útil esta técnica.

## 3. Inspección binaria.

Esta es la misma técnica usada para búsqueda de archivos de datos, y es preferible a las mencionadas anteriormente cuando el archivo índice cabe en memo-

ria principal ya que el número de búsquedas está da do por la siguiente fórmula:

$$E ( N_p ) = \log_2 N_1 - 1$$

Pero cuando el índice es muy grande y se guar da en dispositivos electromecánicos disminuye la -- eficiencia por el vaivén producido por las búsqüe-- das, otro inconveniente radica en la dificultad pa-- ra eliminar o insertar ítems, ya que se trata de un archivo secuencial. Existen técnicas mejores para -- tratar los índices cuando se encuentran en disposi-- tivos electromecánicos pero en alguno de sus pasos o cuando menos al final es recomendable el uso de -- ésta técnica ya que generalmente la última búsqueda se realiza en memoria principal.

#### 4. Inspección por árbol binario.

Si los ítems que se inspeccionan están dispues tos a modo de árbol binario vinculado con punteros es posible agregar ítems sin tener que reorganizar -- los existentes. El número medio de sondajes es el -- mismo que el de inspección binaria.

Las desventajas de esta técnica son el espacio adicional necesario para los punteros y cuando se -- necesitan los distintos ítems en una secuencia de -- clave.

Las técnicas 5, 6 y 7 están básicamente diseñadas para mejorar el tiempo de búsqueda pero tienen el mismo principio que la técnica anterior por lo que no se estudiarán más a fondo, solo se mencionan.

#### 8. Operaciones de conjunto de secuencias.

El nivel inferior de un índice en árbol se llama a veces conjunto de secuencias. Las operaciones de índice pueden ejecutarse descendiendo por las ramas desde el nodo hasta el conjunto de secuencias y su inspección podría determinarse siguiendo cualquier técnica del presente capítulo. Por otra parte esta técnica sólo ayuda en el caso de operaciones -secuenciales o secuenciales con salteos.

#### 9. Almacenes intermedios de mirada lateral.

Esta técnica es usada casi exclusivamente con archivos que tienen la característica de amontonamiento; es decir, que contienen registros que son -accesados frecuentemente durante un periodo de tiempo y después son olvidados, o que están de moda. La técnica consiste en tener en memoria principal un grupo de referencias recientes del índice, entonces en el caso de intentar acceder un registro se inspecciona este grupo de referencias y si no se encuentra, entonces se procede a acceder el índice. A este grupo de referencias del índice se le llama --

"Almacenes intermedios de mirada lateral".

10. Arboles no balanceados.

Desbalancear un árbol se hace con la finalidad de mejorar el tiempo de acceso lo cual se logra solo con los registros de más alta popularidad como es el caso de la técnica anterior y empeora los registros de acceso poco frecuente, esta técnica tiene una desventaja muy significativa que es que necesita un mantenimiento más complicado que las técnicas para árboles balanceados por lo que no es muy recomendable.

11. Direccionamiento por estimación calculada.

Cuando es posible encontrar un algoritmo para estimar la posición aproximada de una entrada en un índice y la estimación es lo suficientemente precisa para reducir la búsqueda al 1% del índice entonces estamos hablando de una economía importante en cuanto al número de sondajes necesarios para encontrar un registro.

La posibilidad de una estimación semejante depende en gran parte de la naturaleza de la aplicación.

12. Indización algorítmica.

Esta técnica basada en la combinación de algo-

rítmo e índice es más rápida para encontrar una entrada en un índice que las técnicas que usan índices en árboles balanceados o desbalanceados pero -- tiene una gran desventaja que consiste en perder la independencia que debe existir entre la ubicación física de un registro en los archivos y su clave lógica. La ubicación física no puede ser modificada -- independientemente del algoritmo de direccionamiento.

Esta técnica es muy conveniente usarla en los sistemas de reservaciones de las líneas aéreas, en estos sistemas debido a la volatilidad de los archivos no es conveniente usar archivos secuenciales y es deseable que los accesos no exijan más de dos -- búsquedas locales, lo cual es factible con esta técnica.

### 13. Indización por desmenuzamiento (hashing).

La combinación de indización con desmenuzamiento es a menudo más conveniente que la indización sola pues requiere menos sondajes del índice o menos movimientos del brazo de acceso cuando el índice se encuentra en disco. Sin embargo ya habíamos visto -- que la principal ventaja del desmenuzamiento era la de poder encontrar un registro con una única búsqueda local sin embargo tiene también sus ventajas sobre el desmenuzamiento puro que consisten la prime-

ra en que los espacios vacios en los cubos ahora se encontrarían en el índice con lo que se tendría un ahorro de espacio y la segunda sería que los desbordes inevitables en la técnica del desmenuzamiento - ocurren en el índice más que en el archivo de datos donde toman mucho tiempo las búsquedas, consecuentemente tendremos otro ahorro de tiempo además y por último los cubos en el índice pueden tener una capacidad adecuada para que el número de desbordes -- sea comparativamente pequeño.

## MANEJO DE STACKS (PILAS).

### INTRODUCCION.

El mecanismo de pila (stacks) es extensamente usado para evaluar expresiones aritméticas y para guardar información relativa al estado de la computadora en cierto instante (al momento de interrumpir un proceso temporalmente, para efectuar otro, por ejemplo).

Los stacks son considerados una de las estructuras lineales más importantes de tamaño variable.

### QUE ES UN MECANISMO DE PILA ?.

Un mecanismo de pila es un dispositivo (teórico) que puede guardar una secuencia de datos  $d_1, d_2, d_3, \dots, d_n$ , metidos uno por uno y en ese orden --- (primero  $d_1$ , despues  $d_2$ , etc.) y los puede devolver (entregar o mostrar) solamente en orden contrario,  $d_n, d_{n-1}, \dots, d_2, d_1$ ), esto es, una pila es un mecanismo que puede guardar cualquier número "n" de datos, pero que para poder leerlos hay que sacarlos en el orden inverso al que fueron almacenados.

La operación de guardar, meter o empujar un dato en una pila se llama "Push", esta operación no destruye la información que había sido guardada con anterioridad, pero la hace inaccesible en forma di-



recta. Para acceder el penúltimo dato que fué metido, primero se tiene que leer, sacar o jalar el último que fué metido, a la operación de sacar un dato se le llama "Pop ", si el dato obtenido de la pila no es guardado en otro lugar se pierde. A este tipo de mecanismos también se les llama LIFO's (last in first out).

El mecanismo de pila o stack puede ser visto como una pila de cajas que contienen información. Si se quiere guardar más información en la pila, entonces hay que agregar una caja nueva en el tope de la pila, operación que hace que las cajas que quedaron abajo no sean accesibles hasta que se saque la de arriba.

Como en este tipo de memorias no existe el concepto de dirección, solo existe el concepto de meter (push) o sacar (pop) un dato, una computadora que cuente con una pila puede especificar muchos movimientos de información empleando un programa con muy pocos bits. Por supuesto que la limitación en la forma de acceder la información contenida induce una limitación en el tipo de problemas que se pueden resolver con un mecanismo de este tipo.

Una aplicación típica del uso de pilas es en la evaluación de expresiones aritméticas en notación polaca.



- 1)  $SP=n?$ . Si  $SP=N$  entonces imprimir mensaje de sobreflujo e irse a la salida.
- 2) Si  $SP \neq N$  entonces  $SP=SP+1$
- 3) Insertar el elemento, hacer dato  $X \longrightarrow M(SP)$
- 4) Terminar.

El algoritmo para suprimir un elemento de un stack sería:

Algoritmo POP. Dado un vector  $S$  con "n" elementos que representan a un stack, y un apuntador  $SP$  al elemento tope del stack, requiriéndose almacenar al elemento tope del stack, en la variable  $POP$ .

- 1)  $SP \neq 0?$ . Si  $SP = 0$  entonces imprimir el mensaje de STACK VACIO e irse a la salida.
- 2) Si  $SP \neq 0$  Entonces sacar el dato fuera del STACK, hacer  $M(SP) = POP$ .
- 3) Decrementar el apuntador, hacer  $SP = SP-1$
- 4) Terminar.

C A P I T U L O IV.TRABAJO PRACTICOINTRODUCCION

Existen algunos parámetros al través de los -- cuales una Base de Datos puede ser optimizada, de-- pendiendo de la aplicación y de la naturaleza de -- ella. Los parámetros Básicos por los cuales la Base de Datos puede ser optimizada son:

- 1.- Tiempo de Desarrollo.
- 2.- Tiempo de Ejecución.
- 3.- Almacenamiento de los Datos.
- 4.- Flexibilidad en la Utilización de la Base de Datos.

Como el estudio de cada uno de los parámetros\_ es muy extenso, enfocaremos nuestro estudio al tiempo de ejecución el cual está dividido en tres par-- tes principales que son:

- 1.- Tiempo de Acceso.
- 2.- Tiempo de Procesamiento.
- 3.- Tiempo de Trasmisión.

Los dos últimos dependen de las característi-- cas de la computadora, por lo que centraremos el -- presente trabajo en el Tiempo de Acceso, el cual de-- pende directamente de las técnicas de acceso que se

estén usando.

El objetivo del presente capítulo es el de realizar un análisis de las técnicas de acceso, para poder conocer de acuerdo al tipo de Base de Datos y al volumen de movimientos en ella, cual es la más adecuada.

Para poder cumplir nuestro objetivo tendremos primero que proponer una Base de Datos, la cual consistirá de un Sistema de Crédito, con información referente a empresas, a sus trabajadores y a los créditos otorgados a ellos.

El segundo paso será diseñar la organización de dicha base, tanto lógica como física basándonos en la metodología vista en el capítulo III. Tomando en cuenta algunas consideraciones como son:

El fijar el número de trabajadores en 10,000 - De los cuales 4,000 cuentan con algún tipo de crédito, y estos trabajadores se encuentran distribuidos en 100 empresas.

El siguiente paso será el someter a nuestra Base a ciertas pruebas de volumen, usando las técnicas de acceso más comunes, las cuales arrojarán como resultado los tiempos de I/O. de CPU y Absoluto (Tiempo de I/O más tiempo de CPU), que nos llevarán a realizar algunas conclusiones.

## DISEÑO DE LA BASE DE DATOS

Haciendo un análisis de diversos sistemas de base de datos de algunas empresas, llegamos a la conclusión de que un ejemplo bastante ilustrativo sería el de el otorgamiento de crédito para vivienda, ya que es uno de los problemas más preocupantes de nuestra sociedad, y de más difícil solución dado los altos costos de los materiales para construcción así como de los terrenos y el bajo poder adquisitivo de los trabajadores.

Partiendo de que el objetivo del sistema de información es el de agilizar el otorgamiento de crédito y que este se logra por medio de las aportaciones que realizan los trabajadores al través de las empresas en que laboran y tomando en cuenta que la información que maneja la empresa es:

Registro Federal de Causantes

Nombre de la Persona

Calle y Número de la Persona

Colonia de la Persona

Zona Postal de la Persona

Municipio de la Persona

Estado de la Persona

Estado Civil

Número de Subordinados

Salario Diario

Número de Crédito

Dictamen Crédito  
Crédito Conyugal  
Situación Crédito  
Forma de Pago  
Línea de Crédito  
Monto de Crédito  
Valor de la Vivienda  
Fecha de Otorgamiento  
Fecha de la Situación  
Fecha del Aviso de Retención  
Número de las Escrituras  
Amortización  
Incremento de la Cuota Fija  
Abonos Bimestrales  
Expediente de la Empresa  
Nombre de la Empresa  
Reg. Fed. Caus. de la Empresa  
Calle y Núm. de la Empresa.  
Colonia de la Empresa  
Zona Postal de la Empresa  
Municipio de la Empresa  
Estado de la Empresa  
Giro  
Fecha de Alta en el Infonavit  
Número de Acreditados  
Monto del Crédito  
Número de Trabajadores  
Monto Aportado.

## Número de Pagos

## Pagos.

Obtenemos las entidades que definen nuestro -- sistema y de acuerdo a las propiedades de cada una de estas, clasificamos la información arriba mencionada, quedando de la siguiente manera:

Tipo Empresa = Registro Expediente: Numérico 4  
 Nombre: Alfanumérico 40  
 RFC: Alfanumérico 10  
 Calle y Número: Alfanumerico 40  
 Colonia: Alfanumérico 20  
 Zona Postal: Numérico 2  
 Municipio: Alfanumérico 40  
 Estado: Alfanumérico 19  
 Giro: Numérico 2  
 Fecha Alta Infonavit: Numérico 6  
 Número de Acreditados: Numérico 8  
 Monto de Crédito: Numérico 10  
 Número de Trabajadores: Numérico 8  
 Monto Aportado: Numérico 10

Tipo Trabajadores = Registro RFC: Alfanumérico 10  
 Nombre: Alfanumérico 40  
 Calle y Número: Alfanumérico 40  
 Colonia: Alfanumérico 20  
 ZP: Numérico 2  
 Municipio: Alfanumérico 40



Estado: Alfanumérico 19  
 Estado Civil: Numérico 1  
 Número de Subordinados: Numérico 2  
 Salario Diario: Numérico 6  
 Expediente Empresa: Numérico 4  
 Número de Crédito: Numérico 10  
 Número de Pagos: Numérico 2  
 Pagos: Numérico 8

Tipo Trab. con Crédito = Registro RFC:Alfanumérico 10.

Núm. de Crédito: Numérico 10  
 Expediente Empresa: Numérico 4  
 Dictamen del Crédito: Numérico 1  
 Crédito Conyugal: Numérico 10  
 Situación del Crédito: Numérico 1  
 Forma de Pago: Numérico 1  
 Línea del Crédito: Numérico 1  
 Monto del Crédito: Numérico 9  
 Valor Vivienda: Numérico 9  
 Fecha Otorgamiento: Numérico  
 Fecha Situación: Numérico 6  
 Fecha Aviso Retención: Numérico 6  
 Número Escrituras: Numérico 7  
 Amortización: Numérico 6  
 Incremento Cuota Fija: Numérico 4  
 Abonos Bimetrales: Numérico 8

Dado lo anterior podemos pasar a dibujar nuestro esquema de la base de datos, mostrando en ella la dependencia funcional al través de relaciones y referencias cruzadas, con lo cual queda el esquema de la siguiente manera:

La información contenida en el esquema se almacenará en la computadora por medio de registros, -- los cuales formarán archivos y de acuerdo a la técnica de acceso que se vaya utilizar se podrán generar estructuras como son:

- 1.- Archivos secuenciales
- 2.- Listas Ligadas
- 3.- Estructuras Jerárquicas (ARBOLES)

## ESTUDIO PRACTICO DE TECNICAS DE ACCESO

Debido a que uno de los parámetros más importantes para que el uso de la base de datos sea óptimo como mencionamos anteriormente es el tiempo de acceso a ella, a continuación haremos un estudio de las principales técnicas de acceso, las cuales son:

- 1.- Secuencial.
- 2.- Secuencial Indexado.
- 3.- No Secuencial Indexado
- 4.- Desmenuzamiento (HASHING).

Solo estudiamos estas 4 técnicas ya que todas las otras son una derivación de estas o solo cambia el algoritmo de conversión de clave dirección, además de que son las más comunmente usadas e implementadas en los sistemas de computo existentes.

Para realizar este estudio se hicieron pruebas de volumen a los archivos generados de la siguiente manera:

- 1.- Se implementaron programas independientes para cada tipo de acceso, contemplando los 3 tipos de movimientos básicos (altas, bajas y cambios).
- 2.- Se varió el volumen de 20% en 20% hasta el 100% para cada uno de los tipos de movimiento.

A continuación haremos un análisis de la lógica seguida para la implementación de los programas en las diferentes técnicas de acceso.

### ACCESO SECUENCIAL.

Debido a que este tipo de acceso es óptimo en archivos secuenciales ordenados en esta sección se describe la manera de como se implementó la base de datos del ejemplo con archivos secuenciales, -- (" Ver la Sección de Diseño de la Base de Datos"), y de cual fue la secuencia de pasos que se siguieron para hacer las pruebas de volumen.

Como primer paso se elaboró un programa el -- cual se encarga de acceder la base de datos secuencialmente, a continuación mostramos diagramas de -- el archivo de tal.

## DESCRIPCION DE ARCHIVOS

## ARCHIVO MOVES.

Es el archivo que contiene movimientos (altas, bajas o cambios) los cuales afectarán a los archivos.

- 1.- Trabajadores - V
- 2.- Trabs-Cred-V
- 3.- Emp-V.

Este archivo contiene 3 tipos de registros, -- los cuales describiremos a continuación uno por uno.

PRIMER REGISTRO (contiene información referente a -- la empresa).

<u>No.</u>	<u>CAMPO</u>	<u>DESCRIPCION</u>	<u>LONGITUD</u>	<u>TIPO</u>
1		Tipo de Registro	1	Núm.
2		Núm. de Expediente	10	Núm.
3		Nombre	40	Alfanum.
4		R.F.C.	10	Alfanum.
5		Calle	40	Alfanum.
6		Colonia	20	Alfanum.
7		Z.P.	10	Núm.
8		Municipio	40	Alfanum.
9		Estado	20	Alfanum.
10		Giro	10	Núm.
11		Fecha de Alta en el		
		Infonavit	10	Núm.

<u>No. CAMPO</u>	<u>DESCRIPCION</u>	<u>LONGITUD</u>	<u>TIPO</u>
12	Núm. de Acreditados	10	Núm.
13	Monto de los Crédi- tos.	10	Núm.
14	Núm. de Trabajado- res.	10	Núm.
15	Monto Aportado	10	Núm.

## SEGUNDO REGISTRO (información de trabajador)

<u>No.</u>	<u>CAMPO</u>	<u>DESCRIPCION</u>	<u>LONGITUD</u>	<u>TIPO</u>
1		Blanco	1	Alfanum.
2		R.F.C.	10	Alfanum.
3		Nombre	40	Alfanum.
4		Calle	40	Alfanum.
5		Colonia	20	Alfanum.
6		Z.P.	10	Núm.
7		Municipio	40	Alfanum.
8		Estado	20	Alfanum.
9		Estado Civil	10	Núm.
10		Núm. Subordinado	10	Núm.
11		Salario Diario	10	Núm.
12		Expediente de la Emp.	10	Núm.
13		Núm. de Crédito	10	Núm.
14		Blancos	20	Alfanum.

TERCER REGISTRO (información del trabajador con --  
crédito):

<u>No. CAMPO</u>	<u>DESCRIPCION</u>	<u>LONGITUD</u>	<u>TIPO</u>
1	Blanco	1	Núm.
2	R.F.C.	10	Alfanúm.
3	Núm. de Crédito	11	Núm.
4	Expediente Empresa	10	Núm.
5	Dictamen Crédito	10	Núm.
6	Crédito Conyugal	10	Núm.
7	Situación Crédito	10	Núm.
8	Forma de Pago	10	Núm.
9	Línea de Crédito	10	Núm.
10	Monto del Crédito	10	Núm.
11	Valor de la Vivienda	10	Núm.
12	Fecha de Otorga- miento.	10	Núm.
13	Fecha de Situación	10	Núm.
14	Fecha Aviso Reten- ción	10	Núm.
15	Núm. de Escritura	10	Núm.
16	Amortización	10	Núm.
17	Incremento o Cuota Fija	10	Núm.
18	Abonos Bimestrales	10	Núm.
19	Blancos	80	Alfanum.



## ARCHIVO TRABAJADORES -V

Este archivo contiene los datos del trabajador, los cuales se ponen a continuación:

<u>No. CAMPO</u>	<u>DESCRIPCION</u>	<u>LONGITUD</u>	<u>TIPO</u>
1	R.F.C.	10	Alfanum.
2	Nombre	40	Alfanum.
3	Calle	40	Alfanum.
4	Colonia	20	Alfanum.
5	Z.P.	10	Núm.
6	Municipio	40	Alfanum.
7	Estado	20	Alfanum.
8	Estado Civil	10	Núm.
9	Núm. Subordinado	10	Núm.
10	Salario Diario	10	Núm.
11	Expediente de la Emp.	10	Núm.
12	Núm. de Crédito	10	Núm.
13	No. de Pagos	10	Núm.
14	Pago Núm. 1	10	Núm.
N + 13	Pago Núm. N	10	Núm.

## ARCHIVO TRABAJADORES - N

Este archivo contiene la información del trabajador actualizada. Los datos que contiene son los mismos que el archivo de trabajadores -V

## ARCHIVO TRABS-CRED-V.

Este archivo contiene la información referente al crédito tomado por algún trabajador, a continuación mostramos sus datos:

<u>No.</u>	<u>CAMPO</u>	<u>DESCRIPCION</u>	<u>LONGITUD</u>	<u>TIPO</u>
1		R.F.C.	10	Alfanum.
2		Núm. de Crédito	10	Núm.
3		Expediente Empresa	10	Núm.
4		Dictamen Crédito	10	Núm.
5		Crédito Conyugal	10	Núm.
6		Situación Crédito	10	Núm.
7		Forma de Pago	10	Núm.
8		Línea de Crédito	10	Núm.
9		Monto de Crédito	10	Núm.
10		Valor de la Vivien- da.	10	Núm.
11		Fecha de otorga- miento.	10	Núm.
12		Fecha de Situación	10	Núm.

<u>No. CAMPO</u>	<u>DESCRIPCION</u>	<u>LONGITUD</u>	<u>TIPO</u>
13	Fecha de Aviso Re tención.	10	Núm.
14	Núm. de Escritura	10	Núm.
15	Amortización	10	Núm.
16	Incremento o Cuota Fija	10	Núm.
17	Abonos Bimestrales	10	Núm.

## ARCHIVO TRABS-CRED-N.

Este archivo contiene la información del crédito del trabajador actualizada. Los datos que contiene son los mismos del archivo arriba descrito.

## ARCHIVO EMP-V.

Este archivo contiene la información referente a alguna empresa. Tal información se describe abajo:

<u>No. CAMPO</u>	<u>DESCRIPCION</u>	<u>LONGITUD</u>	<u>TIPO</u>
1	Núm. de Expediente	10	Núm.
2	Nombre	40	Alfanum.
3	R.F.C.	10	Alfanum.
4	Calle	40	Alfanum.
5	Colonia	20	Alfanum.
6	Z.P.	10	Núm.
7	Municipio	40	Alfanum.
8	Estado	20	Alfanum.
9	Giro	10	Núm.
10	Fecha de Alta en el Infonavit	10	Núm.
11	Núm. de Acreditados	10	Núm.
12	Monto de los Créditos	10	Núm.
13	Núm. de Trabajadores	10	Núm.
14	Monto Aportado	10	Núm.

## ARCHIVO EMP-N.

Este archivo contiene la información de la empresa actualizada. Los datos que contiene son los mismos del archivo EMP-V.

## ARCHIVO ESTADISTICA.

Contiene las cifras de control de los movimientos realizados.

## SEGUNDO PASO:

Fue el de crear programas que generan la información de los archivos bases (Trabajadores-V, Trabs Cread-V y EMP-V), así como los que generan el archivo Movs haciéndolo variar en su volumen de información, como lo mencionamos anteriormente, para cada uno de sus tipos de registro.

## TERCER PASO.

Fue ejecutar el "Programa de Acceso Secuencial en Archivos Secuenciales", el cual se describe a -- continuación:

## DESCRIPCION DEL PROGRAMA.

Este programa accesa secuencialmente a los archivos descritos anteriormente, para obtener los -- tiempos de procesador, entrada/salida y absoluto, -

en cada ejecución. Estas ejecuciones diferirán en el volumen de información que afectará a los archivos de la base de datos, variando también el tipo de movimiento.

Con lo anterior se obtendrán gráficas de tiempo (de procesador, entrada/salida y absoluto) contra volumen.

## ACCESO SECUENCIAL INDEXADO

Esta sección es igual a la de acceso secuencial en archivos secuenciales con las siguientes modificaciones:

- 1.- En el diagrama de archivos cambian de una Organización Secuencial a Secuencial Indexado los siguientes archivos.
  - 1.1.- Trabajadores -V
  - 1.2.- Trabs-Cred-V
  - 1.3.- Emp -V
  - 1.4.- Trabajadores -N
  - 1.5.- Trabs--Créd-N
  - 1.6.- Emp-N

### ACCESO NO SECUENCIAL INDEXADO

Para llevar a cabo pruebas con este tipo de -- acceso se generó un programa que se encarga de accesar la base de datos aleatoriamente, el diagrama de archivos de tal se muestra a continuación.



## DESCRIPCION DE ARCHIVOS

Debido a que las descripciones de los archivos de esta sección son iguales a las descripción de -- los archivos de la sección "diseño con archivos secuenciales", se pone una relación para mayor claridad.

<u>Organización</u> <u>Secuencial Indexado</u>	<u>Organización</u> <u>Secuencial</u>
Movs	Movs
Trabajadores	Trabajadores- V
Trabs-Cred.	Trabs-Cred-V
Emp	Emp
Estadística	Estadística

## SEGUNDO PASO.

Fue el de crear programas que generan los archivos bases (Trabajadores, Trabs-cred y Emp), así como utilizar los que generan el archivo Movs.

## TERCER PASO

Fue ejecutar el "programa de acceso aleatorio en archivos No Secuenciales Indexados".

## DESCRIPCION DEL PROGRAMA.

Este programa accesa aleatoriamente la base de datos, para obtener los tiempos de procesador, Entrada/Salida y Absoluto, en cada ejecución. Estas ejecuciones diferirán en el volumen de información que afectara a los archivos de la base de datos, variando también el tipo de movimiento (Altas, Baja o Cambios).

Con lo anterior se obtendrán gráficas de tiempo (de Procesador, Entrada/Salida y Absoluto) contra volumen.

## ACCESO DESMENUZAMIENTO (HASHING)

En esta sección se hacen pruebas con el acceso desmenuzamiento y se describe de la manera de como se implementó la base de datos de ejemplo (Ver la Sección de "Diseño de la Base de Datos") con archivos de organización Hashing, y de cual fue la secuencia de pasos que se siguieron para hacer las pruebas de volumen.

## PRIMER PASO.

Fue elaborar un programa el cual accederá la base de datos directamente (Acceso Hashing), a continuación mostramos el diagrama de archivos de tal.

## DIAGRAMA DE ARCHIVOS.

Es el mismo que el diagrama del programa de acceso aleatorio de la sección "Diseño con Archivos - Secuencial Indexados", cambiando unicamente la organización de los archivos trabajadores, Trabs-Cred y Emp de No Secuencial Indexado o Hashing.

## DESCRIPCION DE ARCHIVOS.

Igual que la descripción de archivos del programa de acceso aleatorio en la sección "Diseño con Archivos Secuencial Indexado".

## SEGUNDO PASO.

Igual a la del programa de acceso aleatorio en la sección "Diseño con Archivos Secuencial Indexados".

## TERCER PASO.

Fue el de ejecutar el "Programa de Acceso Directo en Archivos de Organización Hashing.

## DESCRIPCION DEL PROGRAMA.

Igual que el programa de Acceso Aleatorio en la Sección " Diseño con Archivos Secuencial Indexados".

## RESULTADO DE LAS PRUEBAS

Como resultado de las ejecuciones de los programas de acceso a la base de datos, se obtuvieron cuatro tablas de tiempos que son:

- a).- Tiempo de Procesador
- b).- Tiempo de Entrada/Salida
- c).- Tiempo Absoluto
- d).- Tiempo Total de Movimientos.

En la tabla de tiempo de procesador así como en la de entrada/salida nos referimos únicamente al tiempo usando para cada uno de los tipos de movimiento (Altas, Bajas o Cambios).

En la tabla de tiempo absoluto, tenemos la suma del tiempo de procesador más el tiempo de entrada/salida para cada uno de los tipos de movimiento.

Finalmente en la tabla de tiempos totales de movimientos tenemos la suma de los tiempos absolutos de los tres tipos de movimientos.

Estas tablas han sido graficadas para visualizar mejor los resultados, haciendo gráficas de los tiempos para cada tipo de movimiento obteniendo las siguientes:

- 1.- Altas (Tiempo de Procesador)
- 2.- Bajas (Tiempo de Procesador)
- 3.- Cambios (Tiempo de Procesador)
- 4.- Altas (Tiempo de E/S)
- 5.- Bajas (Tiempo de E/S)
- 6.- Cambios (Tiempo de E/S)
- 7.- Altas (Tiempo Absoluto)
- 8.- Bajas (Tiempo Absoluto)
- 9.- Cambios (Tiempo Absoluto)
- 10.- Tiempos Totales.

Tanto las tablas de tiempos como las gráficas se muestran a continuación.

## TIEMPO DE CPU

N	ALTAS		BAJAS		CAMBIO
10	81.507	87.455	82.611	87.114	40.471
20	114.102	87.455	112.721	87.114	85.807
30	110.440	87.455	115.589	87.114	85.807
40	126.008	107.409	119.079	111.507	107.707
50	116.210	107.409	124.603	111.507	119.079

## TIEMPO DE I/O

N	ALTAS		BAJAS		CAMBIO
10	83.100	87.455	86.763	87.019	50.730
20	116.197	87.455	113.775	87.114	73.471
30	140.370	87.455	107.500	114.114	10.707
40	101.000	120.451	117.177	120.451	110.349
100	198.454	139.453	122.058	127.747	113.025

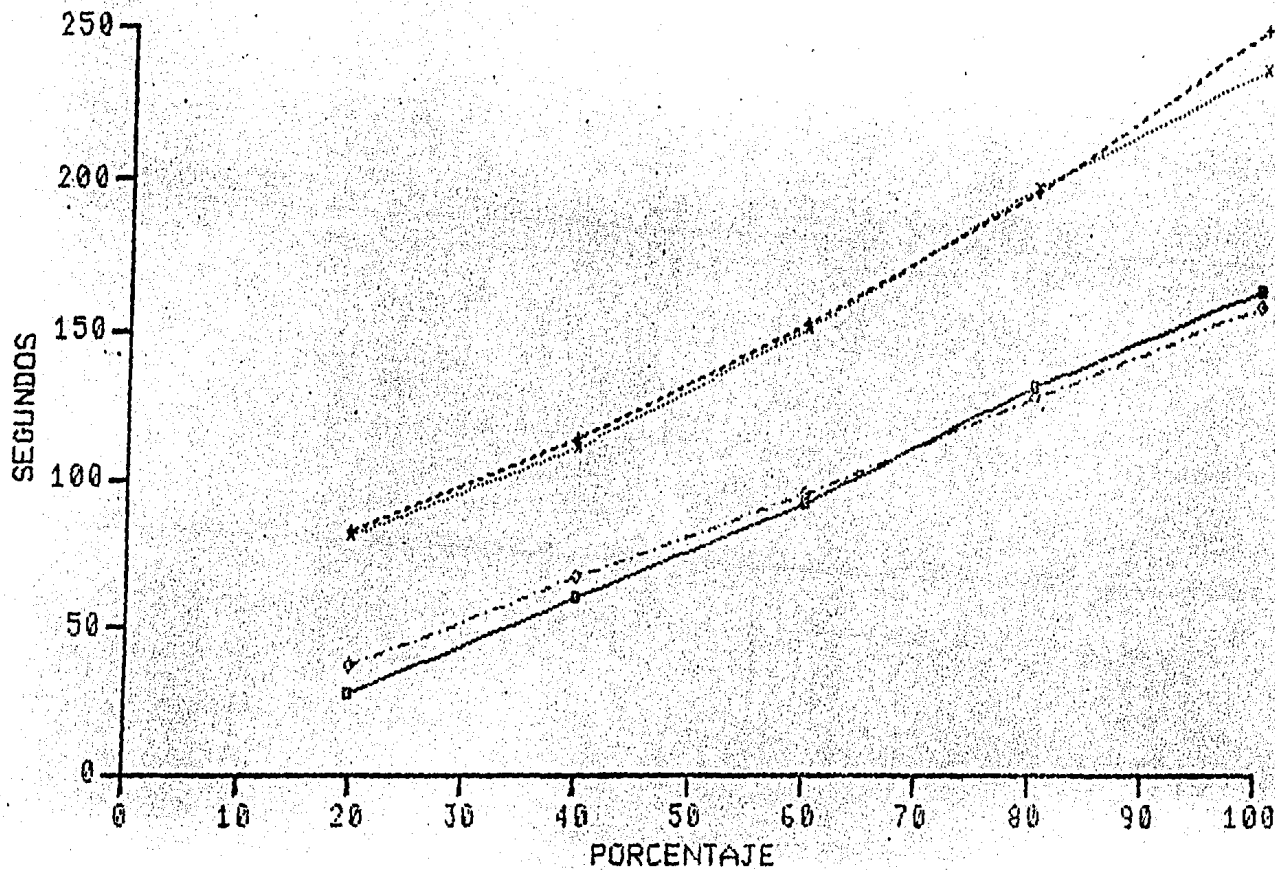
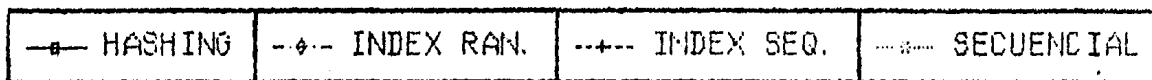


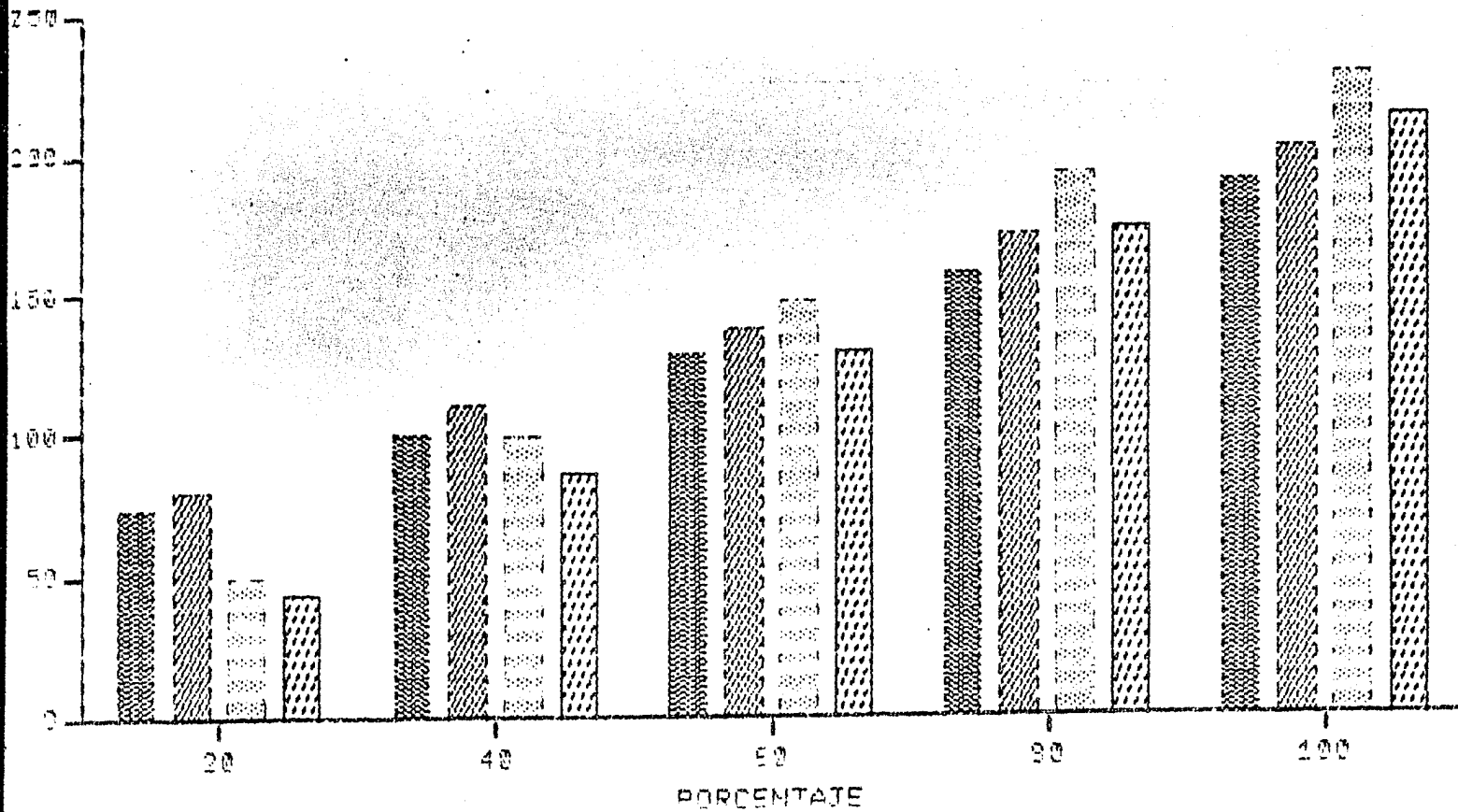


TIEMPO TOTAL

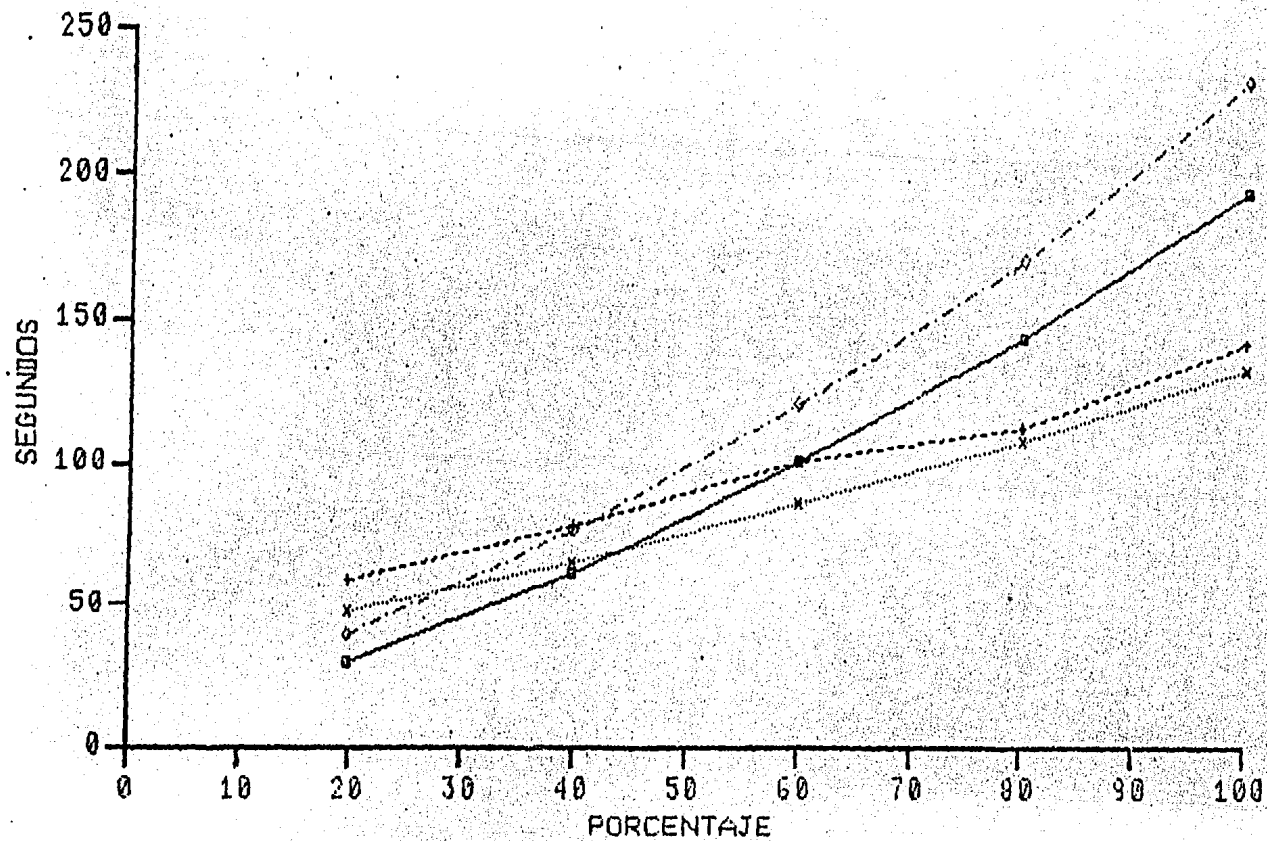
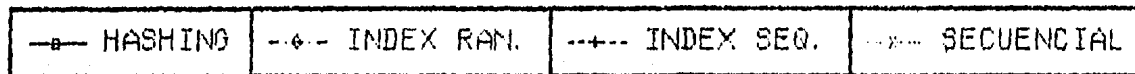
%	SECUENCIAL	NO SECUENCIAL INDEXADO	SECUENCIAL INDEXADO	RASKING
20	426.054	262.975	457.750	
40	575.594		614.35	508.174
60	731.781		762.385	767.058
80		924.995	938.757	1052.088
100		1248.377	1142.628	1314.742

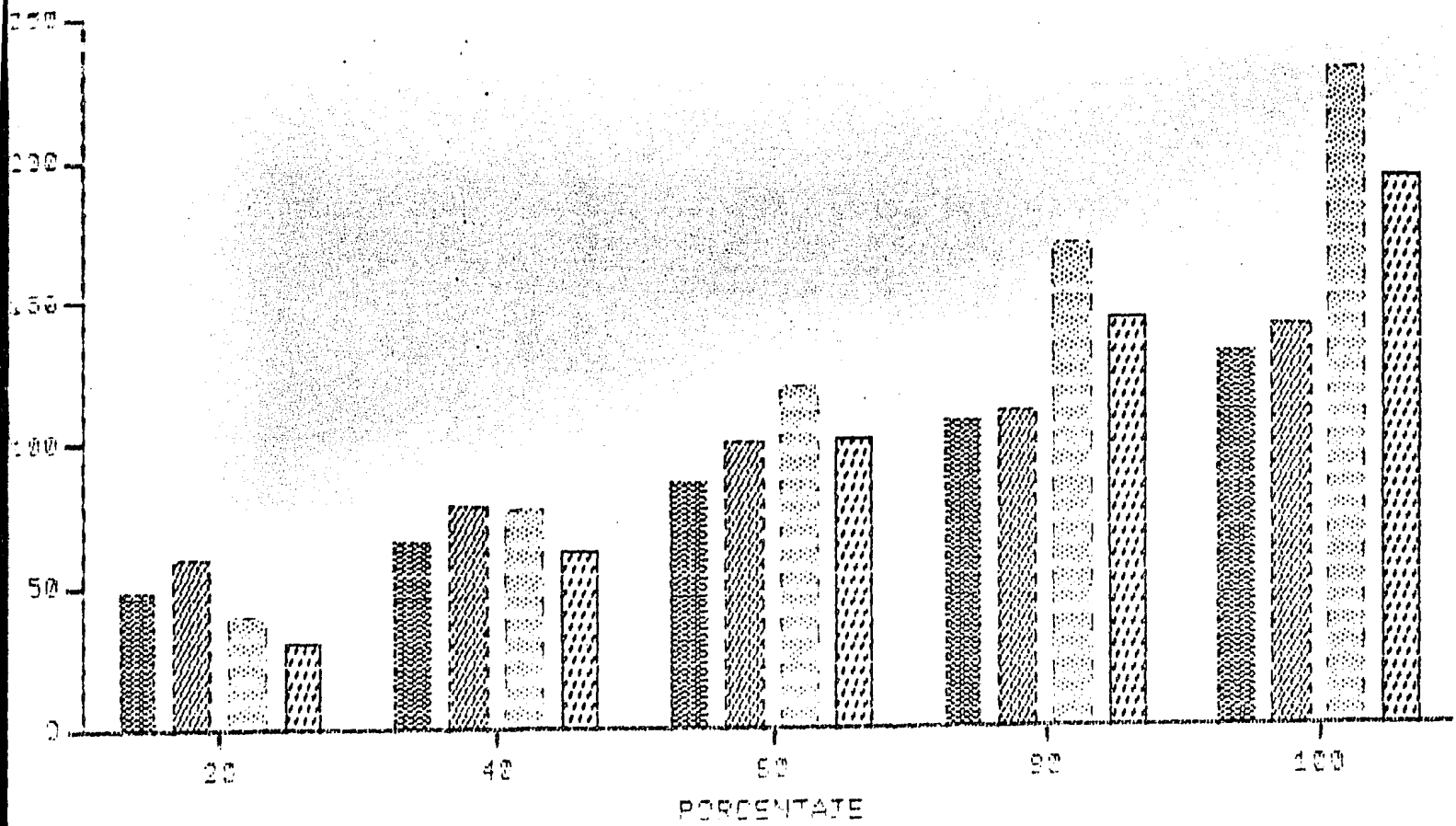
# TIEMPO DE CPU ALTAS



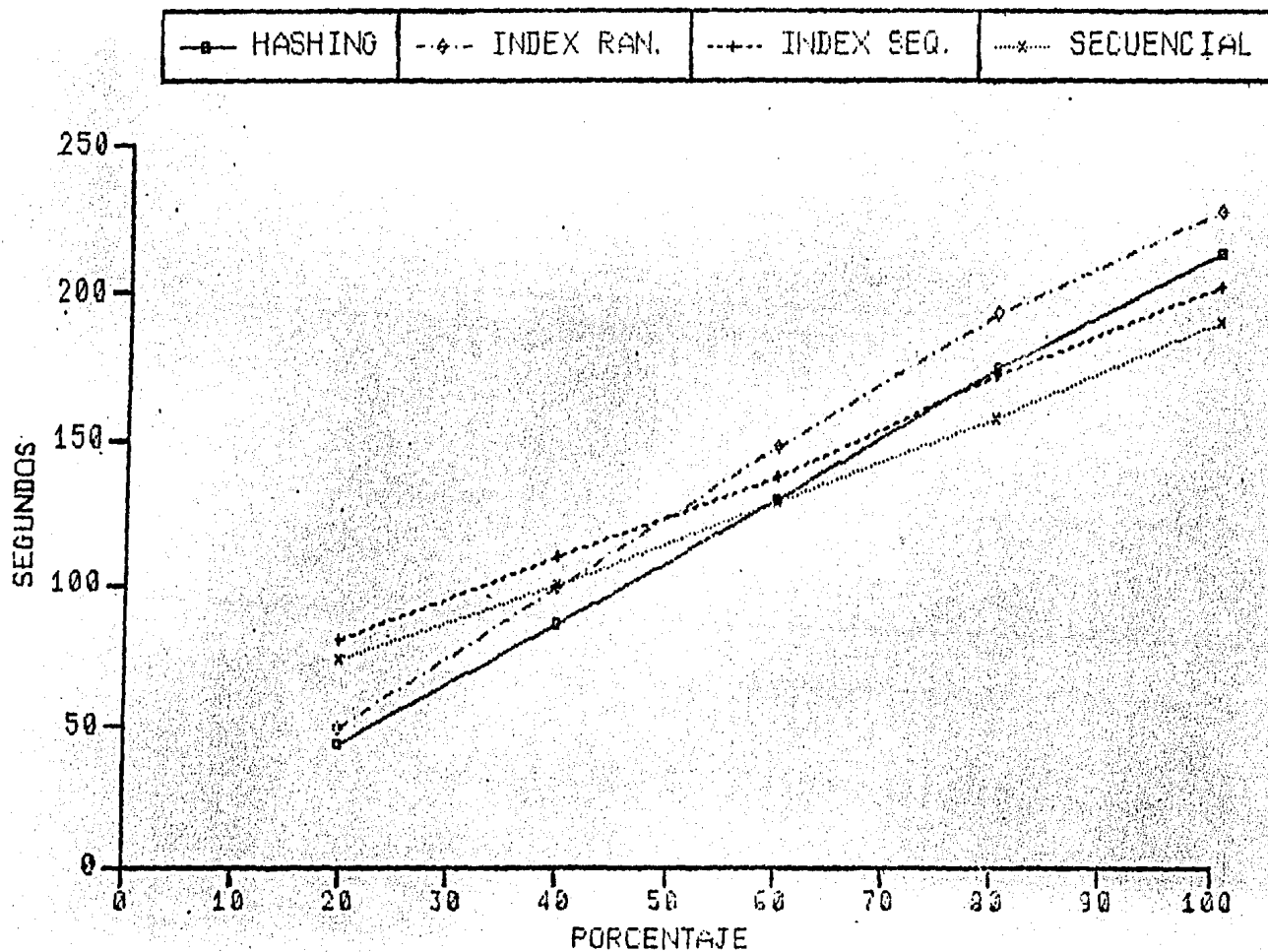


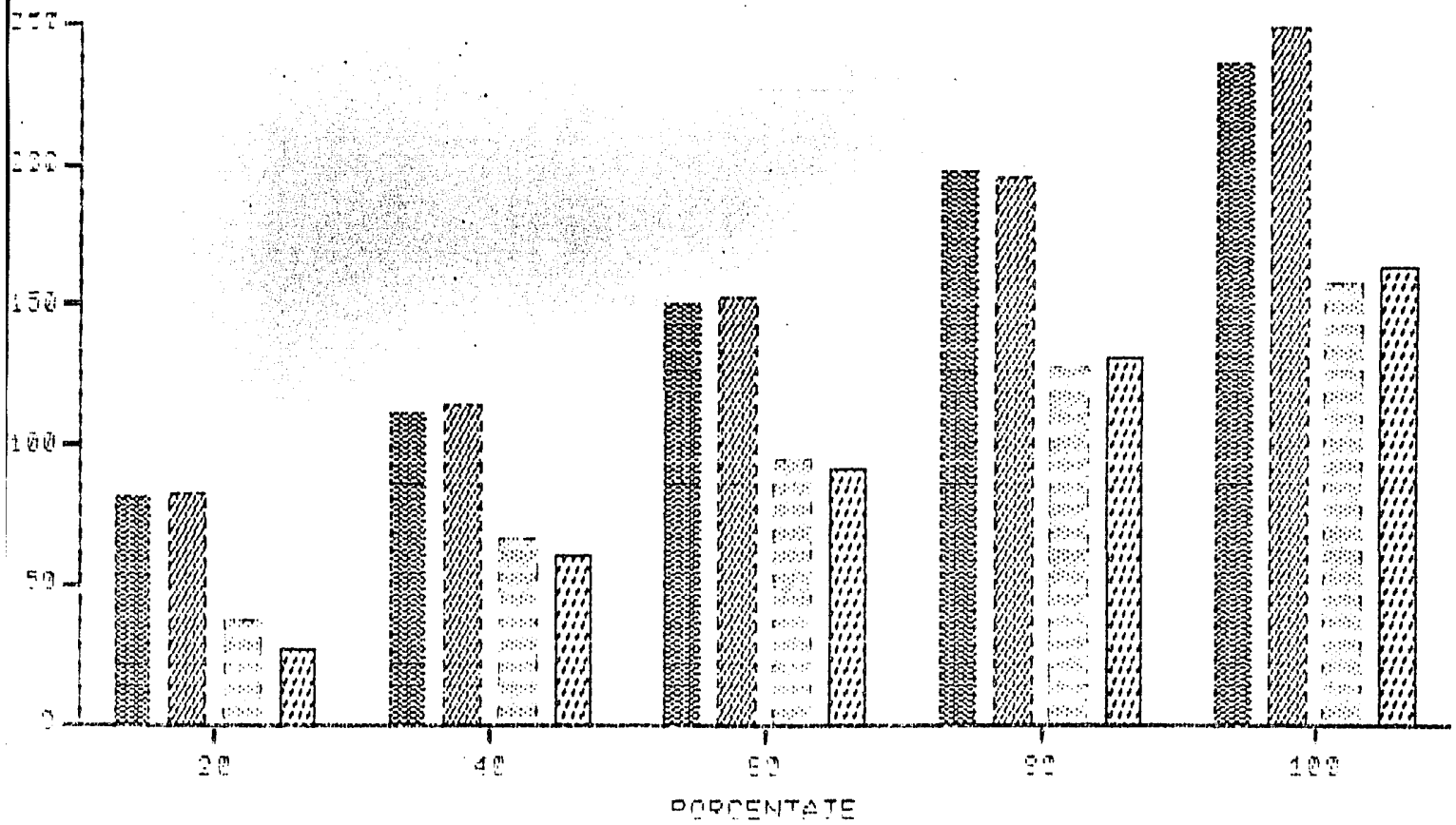
# TIEMPO DE CPU BAJAS



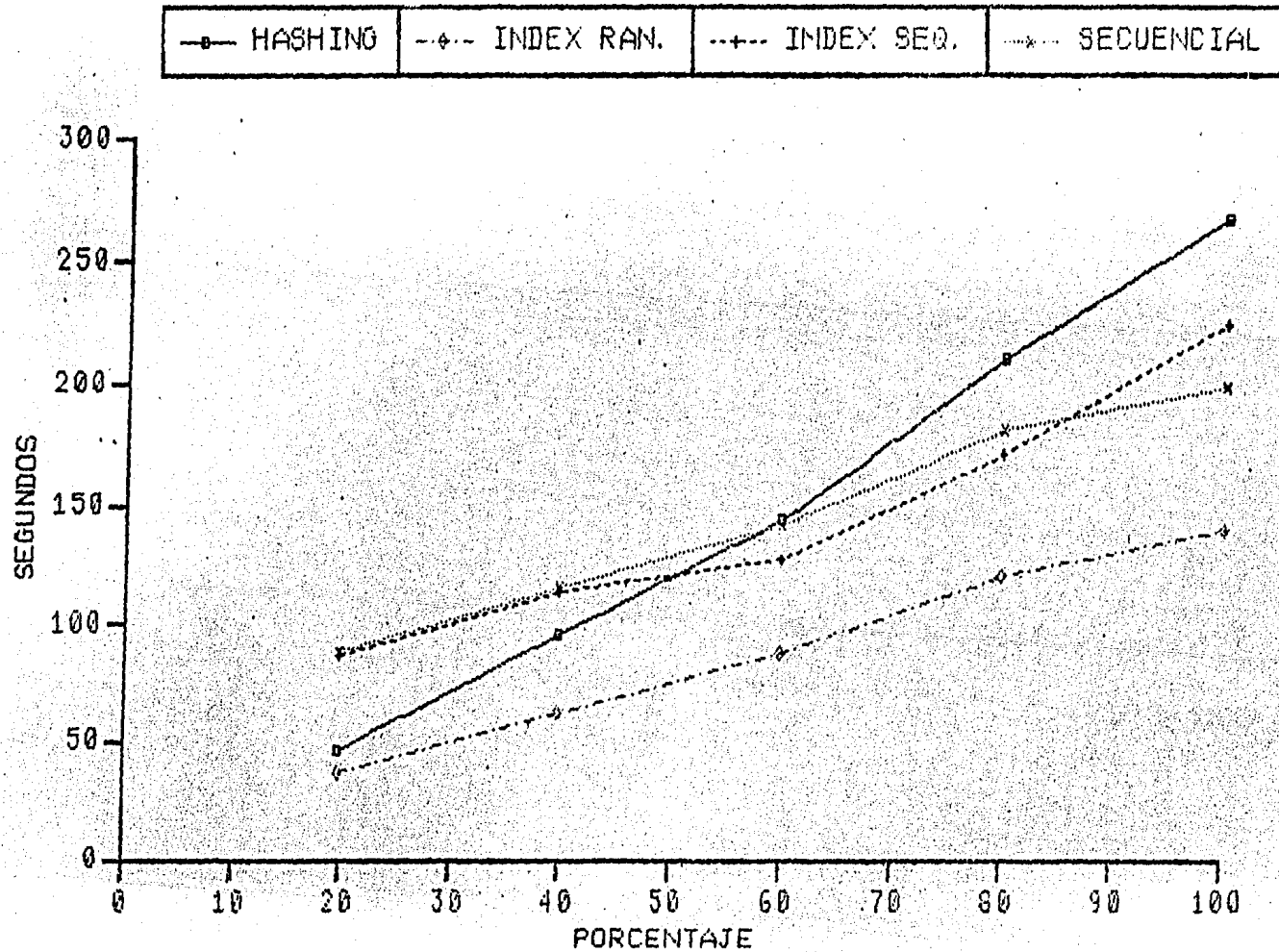


# TIEMPO DE CPU CAMBIOS



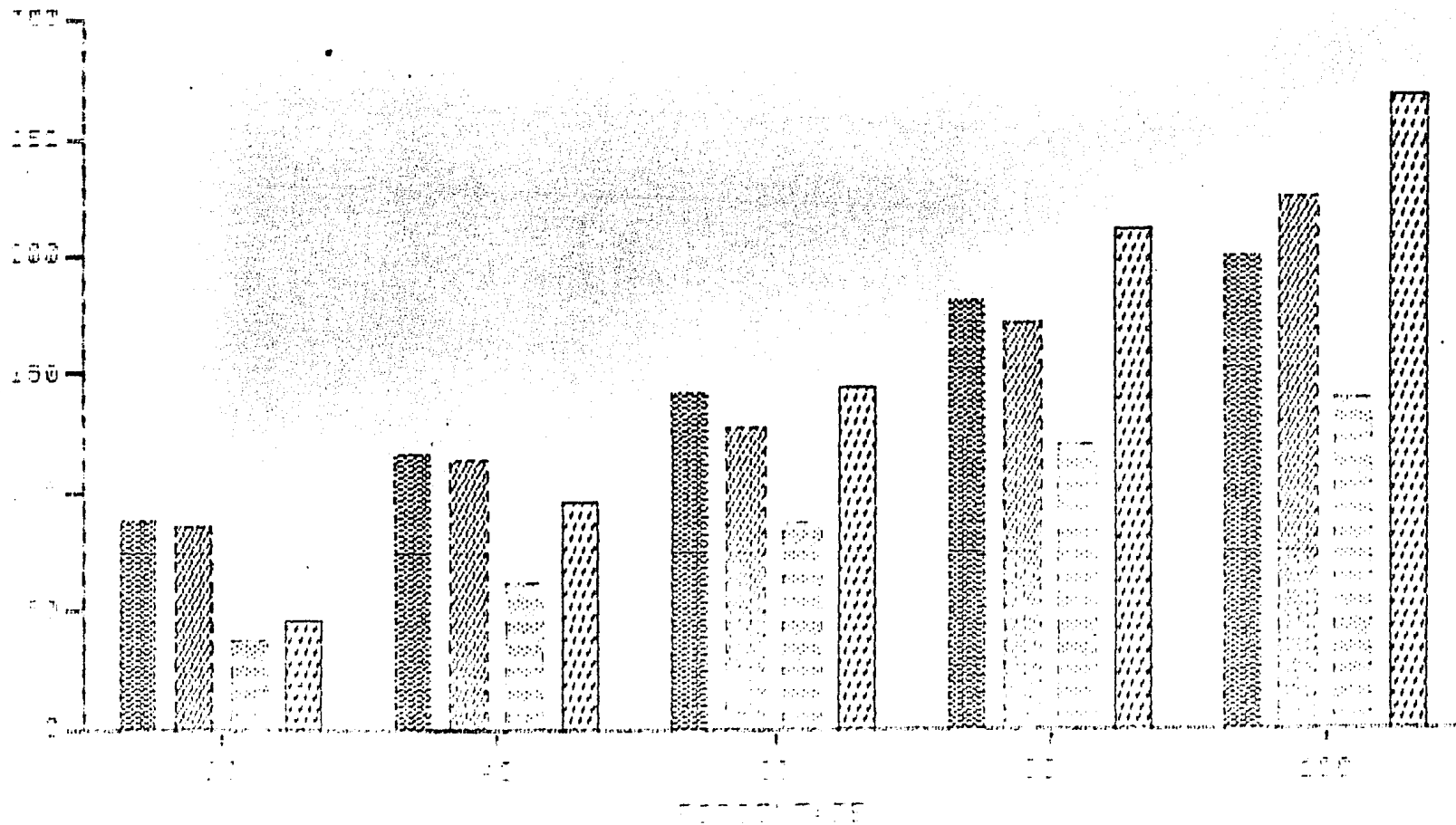
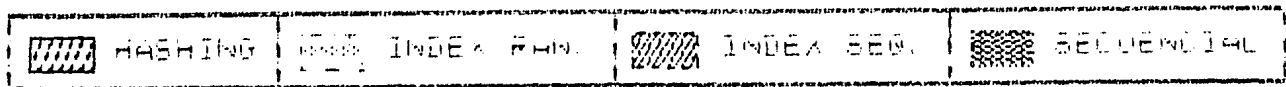


# TIEMPO DE I/O ALTAS

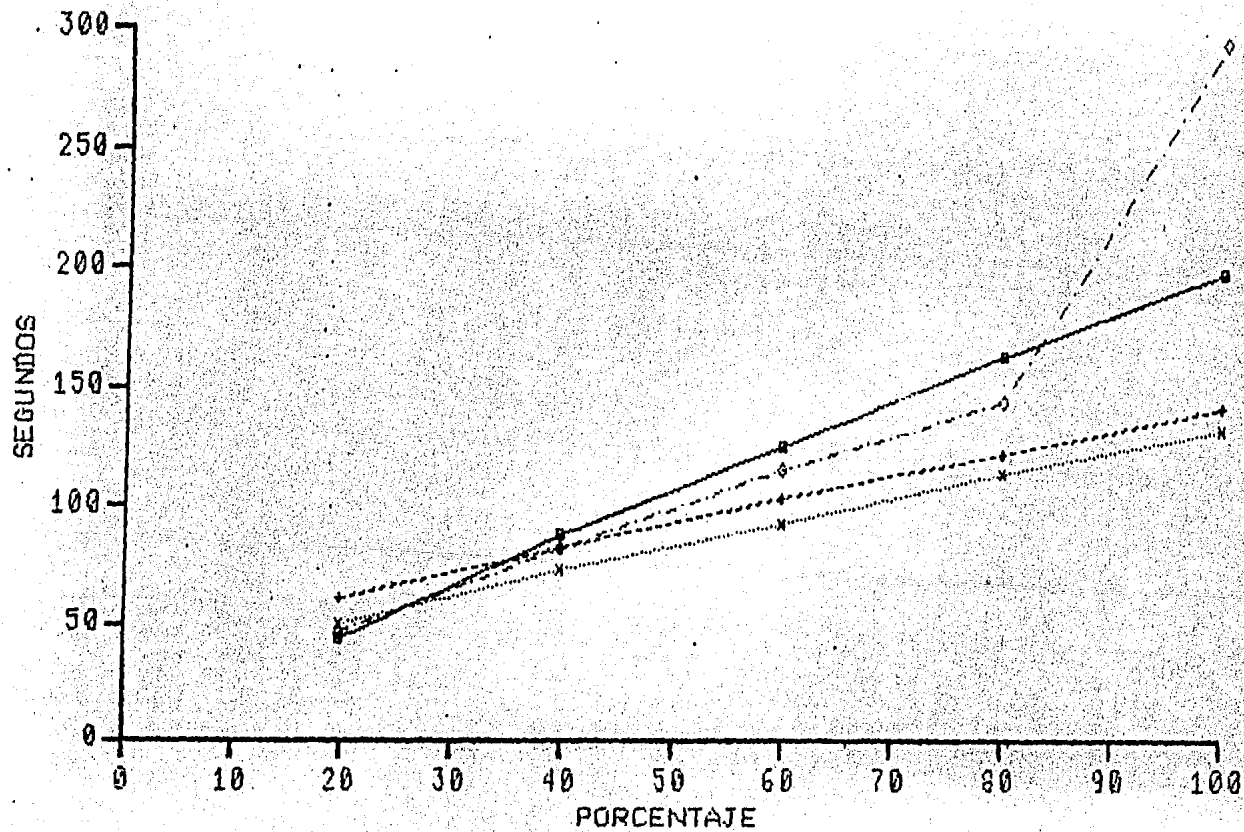
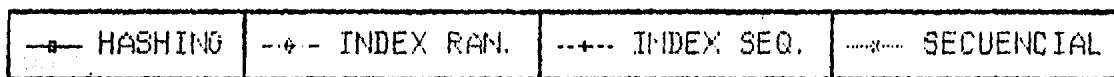




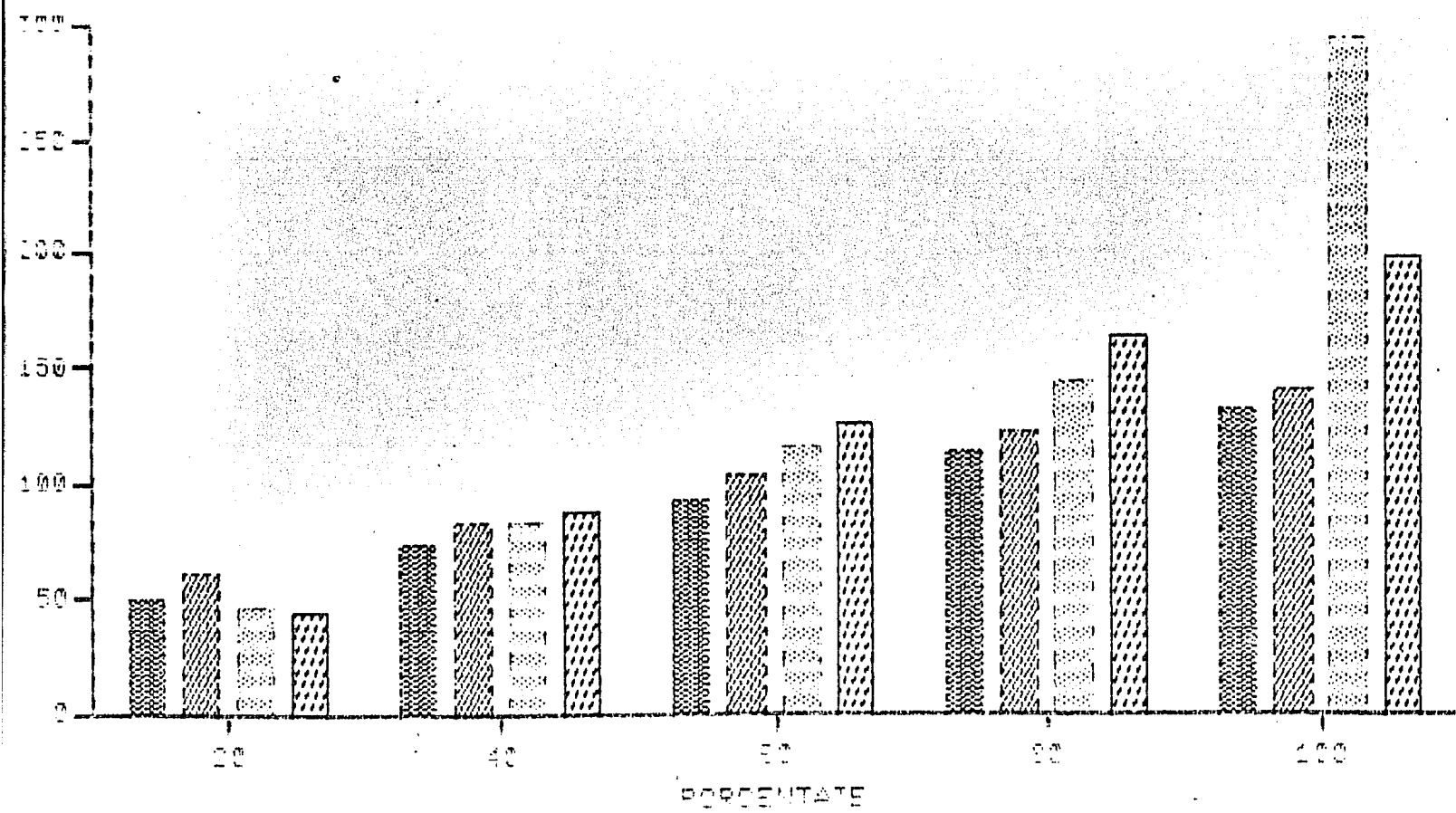
# TIEMPO DE I/O ALTAS



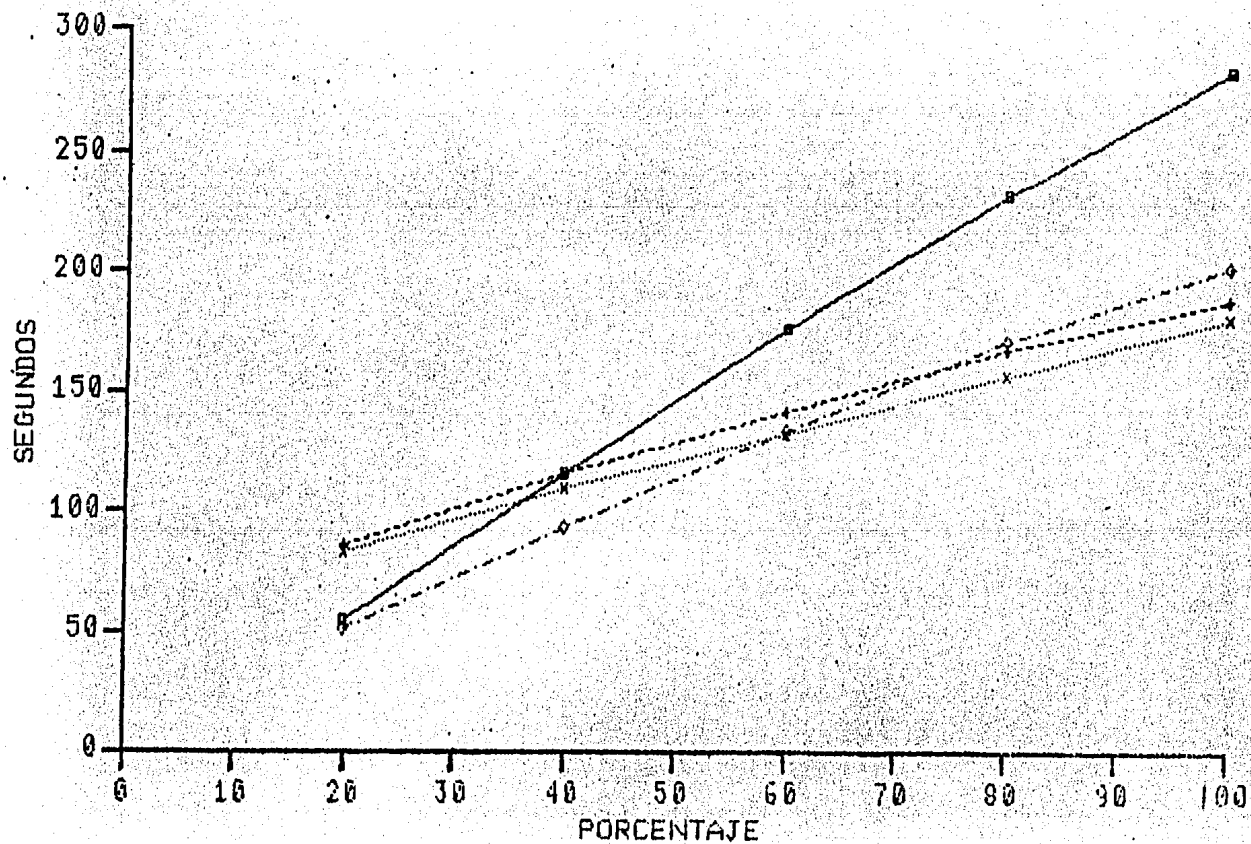
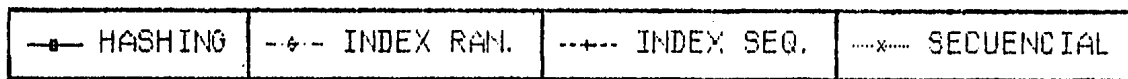
# TIEMPO DE I/O BAJAS



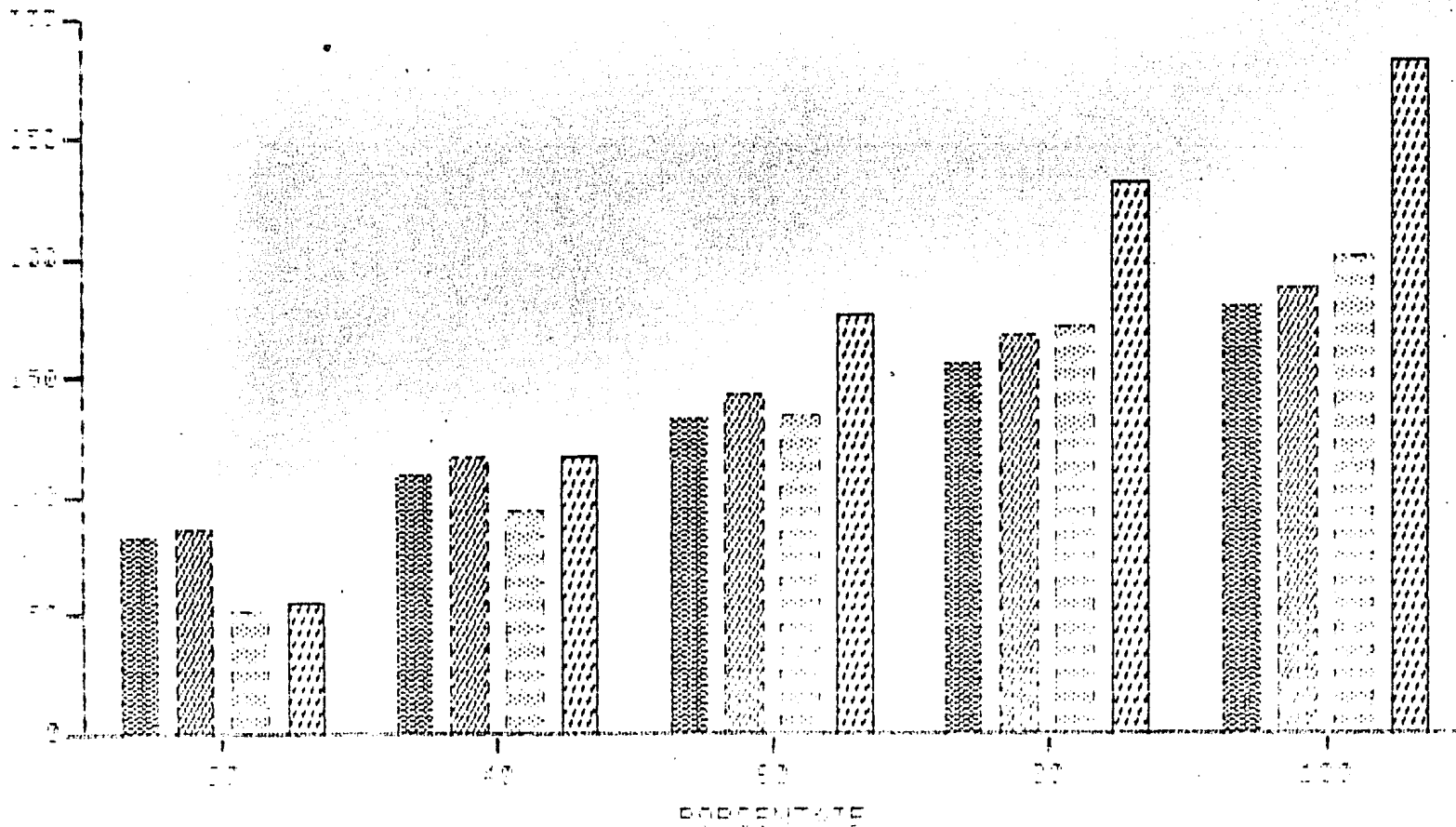
# TIEMPO DE I/O BAJAS



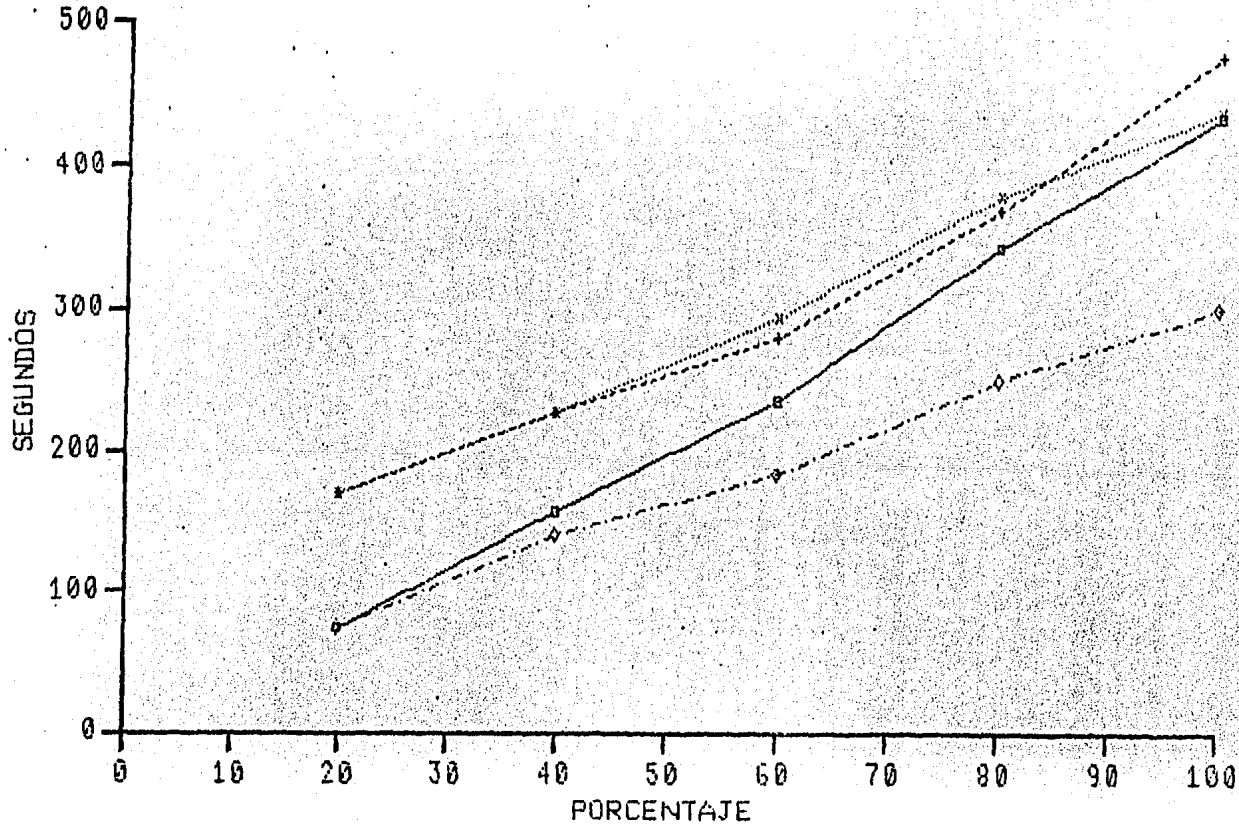
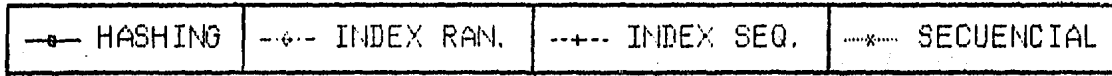
# TIEMPO DE I/O CAMBIOS

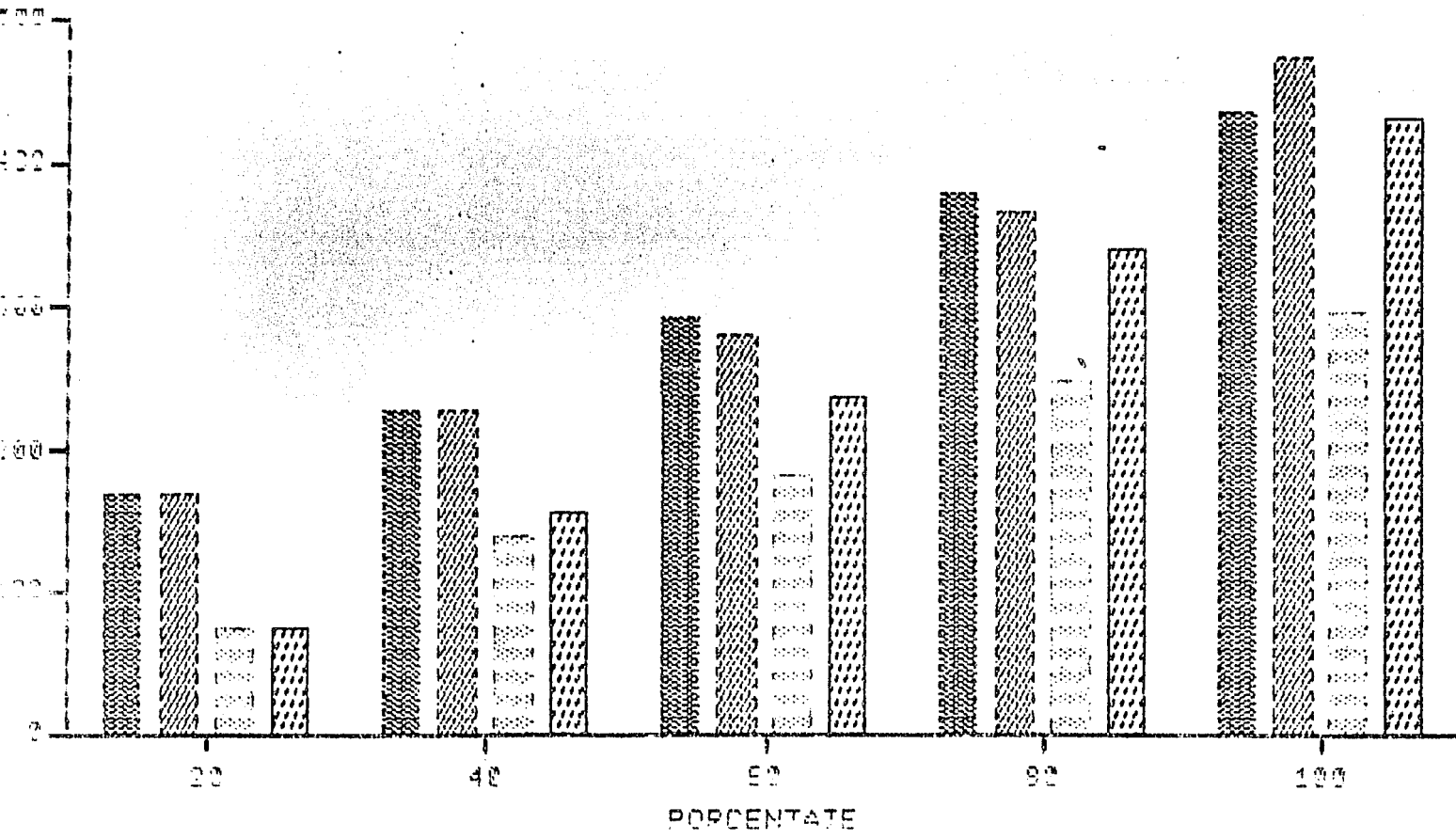


# TIEMPO DE L.V.O CAMBIOS

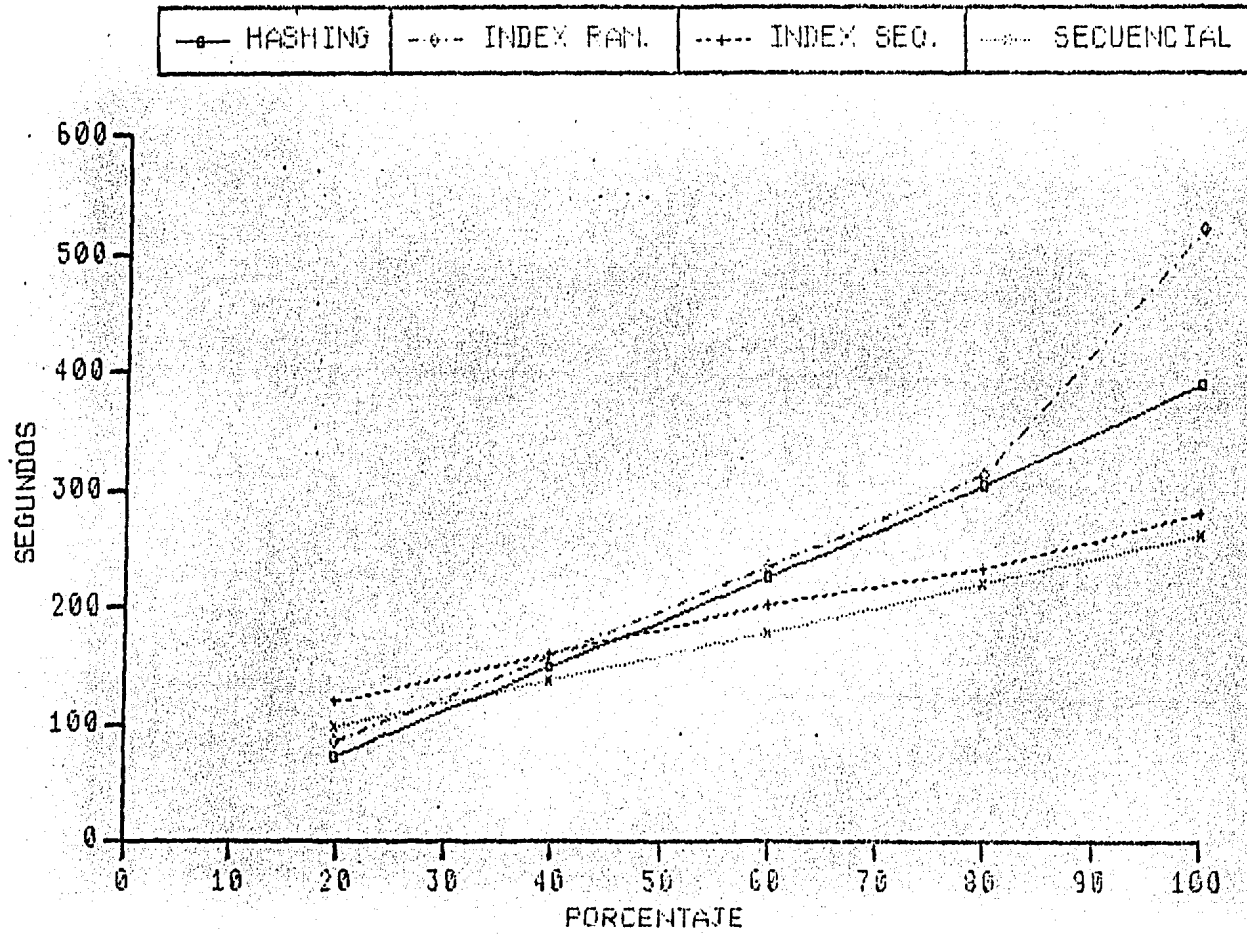


# TIEMPO ABSOLUTO ALTAS



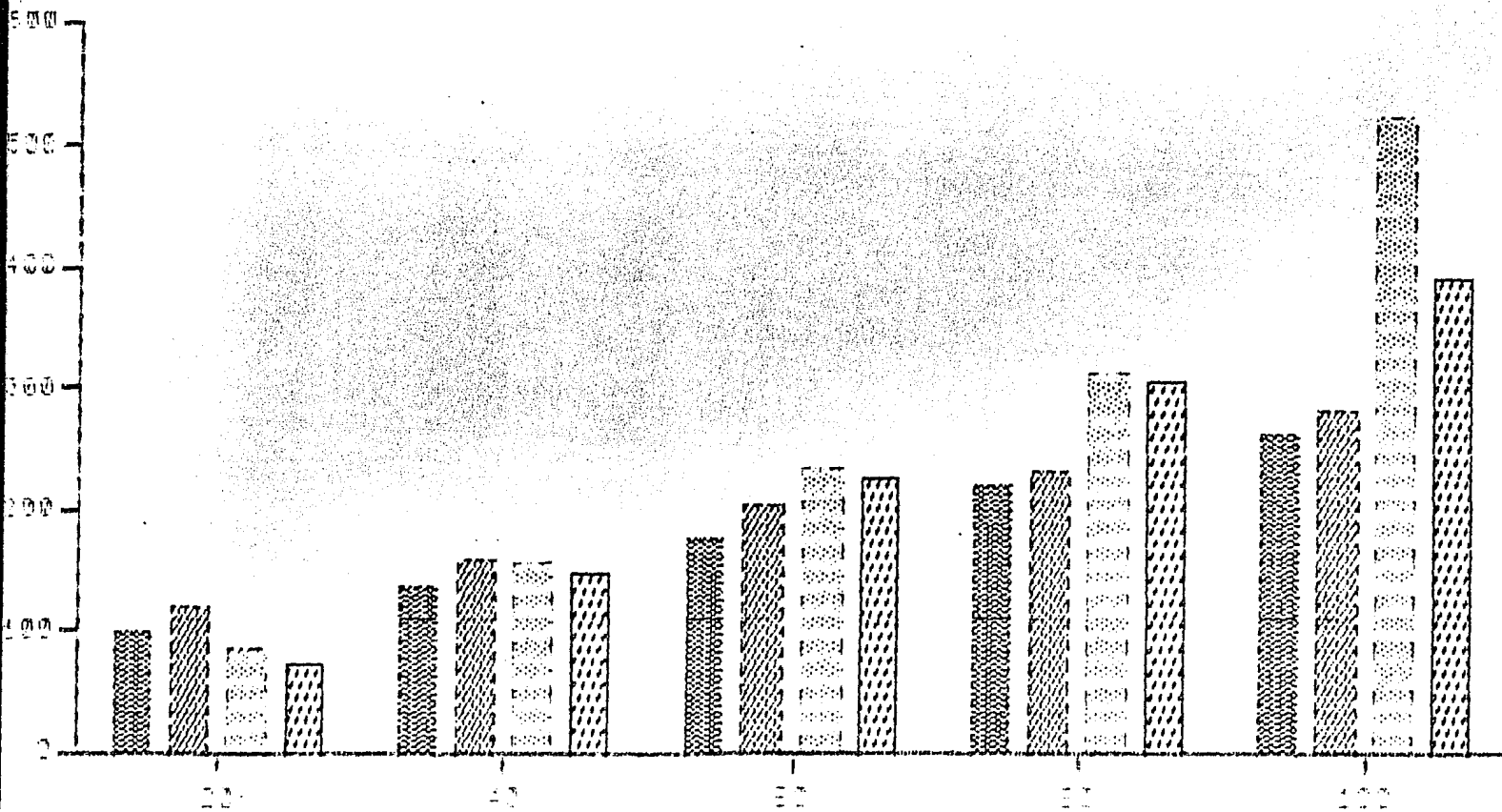


# TIEMPO ABSOLUTO BAIAS

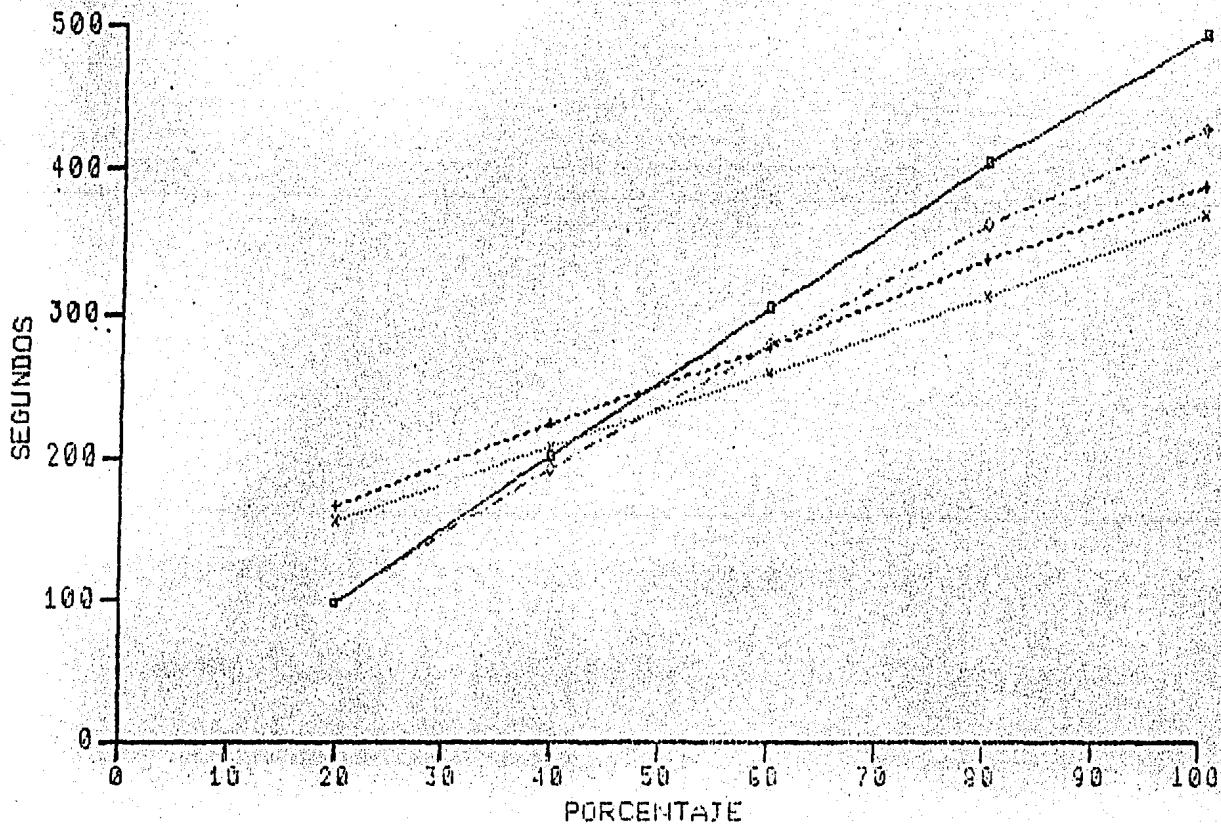
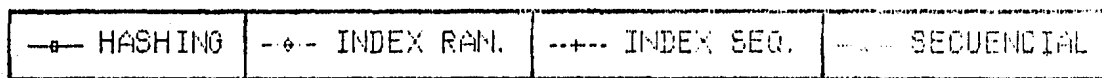




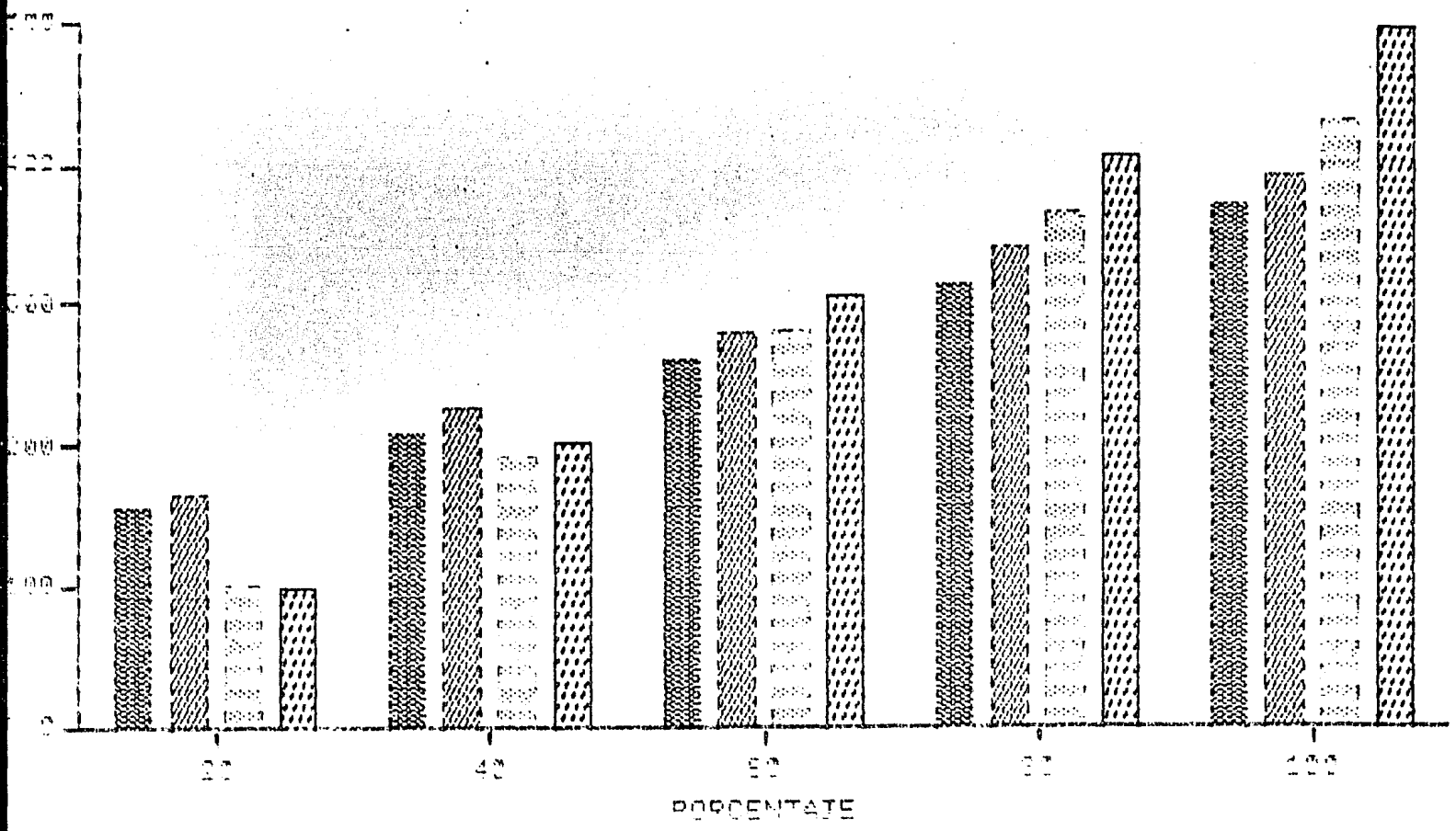
# TIEMPO ABSOLUTO BAJAS



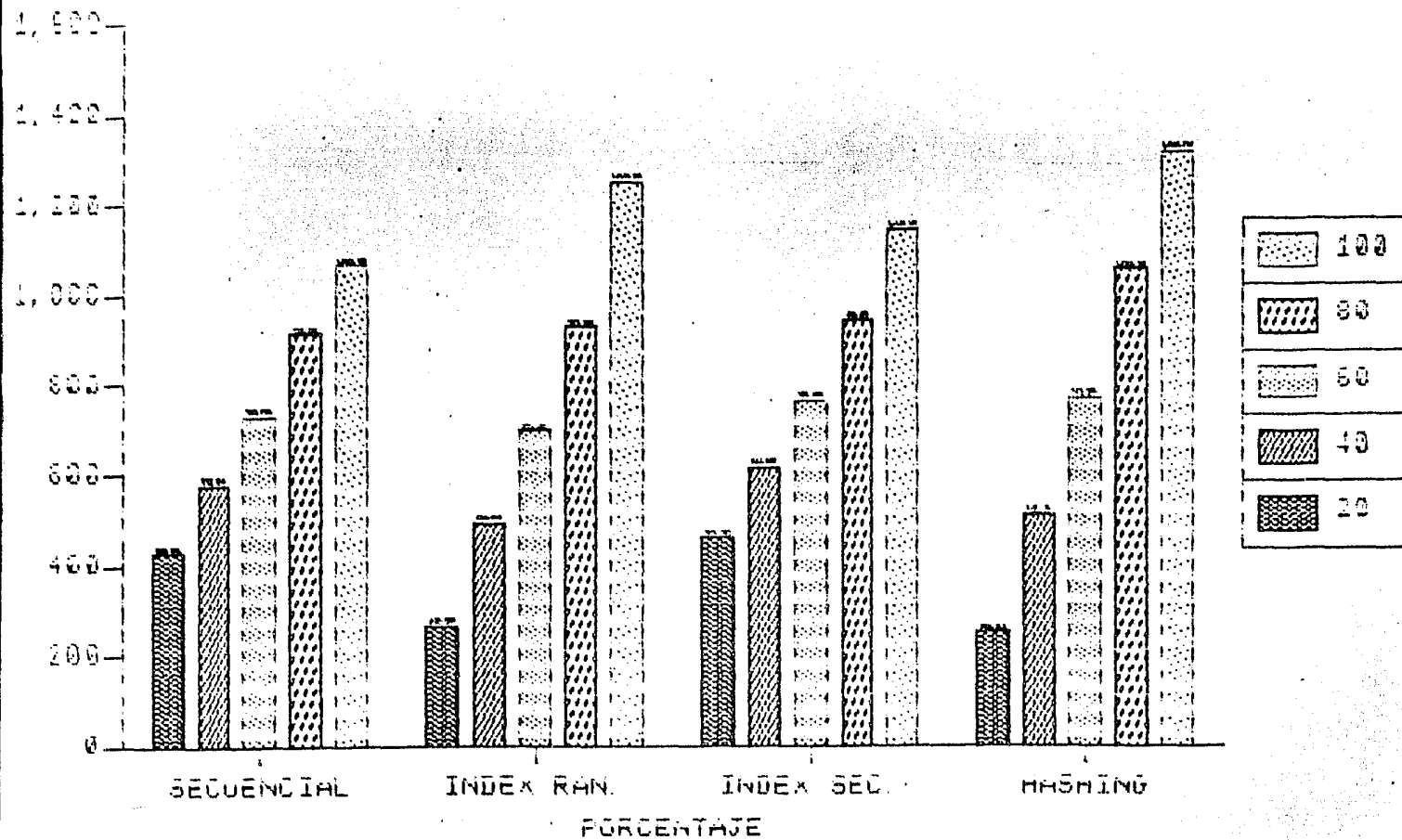
# TIEMPO ABSOLUTO CAMBIOS



# TIEMPO ABSOLUTO CAMBIO



# TOTALS



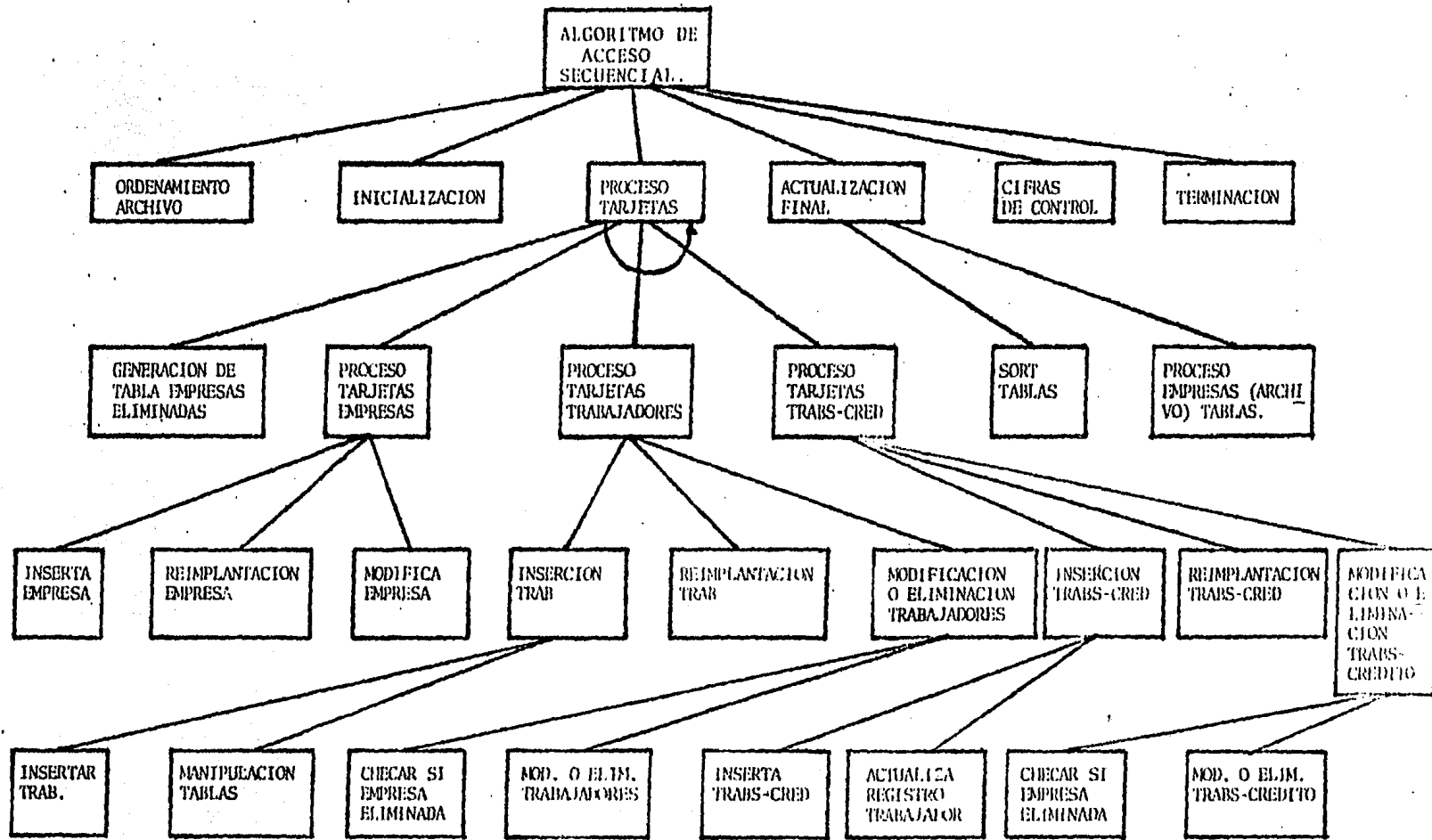
## DESCRIPCION DEL SISTEMA DE ACCESO

Para poder realizar el objetivo de esta tesis se generó un sistema el cual consta de lo siguiente:

- 1.- Un Subsistema de GENERACION DE DATOS Y ARCHIVOS MAESTROS, cuyos programas utilizados, así como la función de ellos es descrita en el --diagrama jerárquico que se muestra a continuación y el cual da un panorama general de todo el sistema.
- 2.- Un Subsistema de ACCESO SECUENCIAL EN ARCHIVOS SECUENCIALES para hacer las pruebas de dicho acceso, también se muestran los programas utilizados así como su funcionamiento.
- 3.- Un Subsistema de ACCESO SECUENCIAL INDEXADO - para realizar las pruebas de dicho acceso, en el diagrama jerárquico del sistema se muestran sus programas, así como sus objetivos, - (en este subsistema hay bloques de programas sin llenar, esto es debido a que tales programas ya fueron definidos en los subsistemas anteriores.

- 4.- Un Subsistema de ACCESO NO. SECUENCIAL INDEXADO para realizar las pruebas de tal acceso, - con respecto a los programas con que se implemento se muestran en el diagrama del sistema (los bloques de los programas en blanco fueron definidos anteriormente).
  
- 5.- Un Subsistema de ACCESO HASHING para realizar las pruebas de este acceso, sus programas, -- así como la función que realizan se muestra - en el diagrama del sistema (los bloques de -- los programas en blanco fueron definidos anteriormente).







## DESCRIPCION DEL ALGORITMO DE ACCESO SECUENCIAL EN ARCHIVOS SECUENCIALES.

Para mayor entendimiento de la descripción del algoritmo, pondremos con letras mayúsculas los nombres de los bloques que se muestran en el diagrama jerárquico anterior y a continuación de tales la descripción del funcionamiento de ellos.

### ORDENAMIENTO DE ARCHIVOS MOV.S.

Aquí se ordena el archivo MOV.S, de tal manera que entren registros de EMPRESAS (a su vez ordenados por el NUMERO DE EXPEDIENTE), TRABAJADORES --- (que también iran ordenados POR REGISTRO FEDERAL DE CAUSANTES) y de TRABAJADORES CON CREDITO (ordenados igual que TRABAJADORES) en este orden. El archivo ordenado lo llamamos ARCHST.

### INICIALIZACION.

Esta rutina hace las siguientes funciones:

- 1.- Abre los archivos de ENTRADA y los archivos de SALIDA (ver diagrama de archivos de esta sección).

- 2.- Inicializa todas las variables, así como las tablas del algoritmo.
- 3.- Lee los primeros registros de todos los archivos de ENTRADA (ver diagrama de archivos de esta sección).

#### PROCESO TARJETAS.

Esta rutina se ejecuta hasta que se agota el archivo ARCHST (que es el archivo ordenado de MOVES)

Las funciones de esta rutina son las siguientes:

- 1.- Generar la TABLA DE EMPRESAS ELIMINADAS
- 2.- Procesar los registros de EMPRESAS del archivo ARCHST.
- 3.- Procesar los registros de TRABAJADORES del archivo ARCHST.
- 4.- Procesar los registros de los TRABAJADORES CON CREDITO del archivo ARCHST.

#### GENERACION TABLA EMPRESAS ELIMINADAS.

Quando el tipo de registro es igual a cero, el proceso entra en esta rutina la cual mete el NUMERO DE EXPEDIENTE de tal EMPRESA en la TABLA DE -

EMPRESAS ELIMINADAS.

PROCESO TARJETAS EMPRESAS.

El proceso entra en ésta rutina cuando se tiene un registro de EMPRESA para darlo de alta o cambio (esto lo sabe el algoritmo al checar el tipo de registro, el cual debe de ser igual a 1) en el archivo MAESTRO DE EMPRESAS.

INSERTA EMPRESA.

Esta rutina da de alta el nuevo registro de EMPRESA.

REIMPLANTACION EMPRESA.

La función de este bloque es la de reimplantar el registro de EMPRESA al cual no se le afectado en nada.

MODIFICA EMPRESA.

Al checar que el NUMERO DE EXPEDIENTE del registro del archivo ARCHST se encuentra en el archivo MAESTRO de EMPRESAS, se modifica este y se registra en tal archivo.

## PROCESO TARJETAS TRABAJADORES.

El proceso entra en esta rutina cuando se tiene un registro de TRABAJADORES de archivo ARCHST, el cual va a actualizar el archivo MAESTRO DE TRABAJADORES (esto sucede cuando el tipo de registro es igual a 2).

## INSERCIÓN TRAB.

Esta rutina inserta el registro del TRABAJADOR nuevo (registro leído del archivo ARCHST) y actualiza la TABLA-EMPRESA, lo anterior lo hace ejecutando las rutinas de INSERTAR TRAB Y MANIPULACIÓN TABLAS respectivamente.

## REIMPLANTACIÓN TRAB.

La función de este bloque es regrabar los registros del archivo MAESTRO de TRABAJADORES los cuales no hayan sido afectados en nada.

## MODIFICACIÓN O ELIMINACIÓN TRAB.

Como su nombre lo dice, esta rutina modifica o elimina el registro del TRABAJADOR en el archivo MAESTRO DE TRABAJADORES.

## CHECAR SI EMPRESA ELIMINADA.

Checa si la EMPRESA de trabajador existe o no, esta información la pasa a la rutina siguiente por medio de una bandera.

MOD. O ELIM. TRAB.

Esta rutina elimina el registro del TRABAJADOR siempre y cuando suceda lo siguiente:

- 1.- Su empresa haya sido eliminada.
- 2.- Haya dejado de trabajar dicha persona (SALARIO DIARIO igual a cero).

Si no sucede lo anterior entonces se modifica rá el registro del archivo MAESTRO de TRABAJADORES por el registro del TRABAJADOR contenido en el archivo ARCHST.

Como todo lo anterior modifica la información de las tablas: TAB-PARA-EMP (en su campo NUMERO DE TRABAJADORES) y TABLA-EMP-C en los campos NUMERO - DE ACREDITADOS, MONTO CREDITOS), estas se tienen - que actualizar también.

PROCESO TARJETAS- TRABS-CREDITO

El proceso entra en esta rutina cuando se tie

ne un registro de TRABAJADORES CON CREDITO del archivo ARCHST, el cual afectará el archivo MAESTRO de TRABAJADORES CON CREDITO (esto sucede cuando el tipo de registro es igual a 3).

#### INSERCIÓN TRABS-CREDITO.

Esta rutina tiene la siguiente función:

- 1.- Inserta el nuevo registro en el archivo MAESTRO DE TRABAJADORES CON CREDITO (por medio de la rutina INSERTA TRAB-CREDITO).
- 2.- Actualiza el registro correspondiente en el archivo MAESTRO DE TRABAJADORES, en el campo de PAGO, el cual no necesariamente contiene su valor correcto, ya que si tiene tal trabajador algún crédito, a tal campo hay que sumarle el ABONO BIMESTRAL para que contenga el valor correcto. Aquí también se tiene que actualizar la tabla TABLA-EMP-C en sus campos de NUMERO DE ACREDITADOS, MONTO DE LOS CREDITOS Y MONTO APORTADO.

Todo lo anterior es por medio de la rutina --  
ACTUALIZA REGISTRO TRABAJADOR.

## REIMPLANTACION TRABS-CREDITO.

Esta rutina reimplanta los registros de los - TRABAJADORES CON CREDITO que no fueron afectados - por el proceso en su archivo (MAESTRO DE TRABAJADOS RES CON CREDITO)

## MODIFICACION O ELIMINACION TRABS-CREDITO.

Checa que la EMPRESA de tal trabajador exista o no (por medio de la rutina CHECAR SI EMPRESA ELIMINADA) y tal información la pasa a la rutina MOD. O ELIM. TRAB-CREDITO por medio de una bandera.

También esta rutina elimina los registros de TRABAJADORES CON CREDITO cuando sucede lo siguiente.

- 1.- Sí la EMPRESA del trabajador ya no existe.
- 2.- Sí ya terminó de pagar el TRABAJADOR su deuda.

Además modifica el registro del TRABAJADOR -- CON CREDITO del archivo MAESTRO y la tabla TABLA--EMP-C en sus campos de NUMERO DE ACREDITADOS, MONTO CREDITOS Y MONTO APORTADO.

Todo lo anterior lo ejecuta la rutina MOD. O ELIM TRAB-CREDITO.

## ACTUALIZACION FINAL.

Debido a que cuando se procesan los registros de EMPRESAS no está toda la información para actualizarlos, ya que ésta se genera cuando se procesan los registros de TRABAJADORES. Esta información es guardada en las tablas:

- 1.- TAB-PARA-EMP
- 2.- TABLA-EMP-C

En esta rutina ordenamos estas TABLAS por el método de la Burbuja, para así procesarlas secuencialmente con el archivo MAESTRO DE EMPRESAS.

## SORT TABLAS

Es la rutina que se encarga de ordenar los dos TABLAS arriba mencionadas.

## PROCESO EMPRESAS (ARCHIVO) TABLAS.

Es la rutina que actualiza los datos siguientes de los registros de la empresa (con ayuda de las TABLAS).

- 1.- NUMERO DE TRABAJADORES
- 2.- NUMERO DE ACREDITADOS
- 3.- MONTO DEL CREDITO.



## 4.-MONTO APORTADO.

## CIFRAS DE CONTROL.

Esta rutina saca las siguientes cifras de control:

ALTAS EN EMPRESAS  
BAJAS EN EMPRESAS  
CAMBIOS EN EMPRESAS  
ALTAS EN TRABAJADORES  
BAJAS EN TRABAJADORES  
CAMBIOS EN TRABAJADORES  
ALTAS EN TRABS. CREDITO  
BAJAS EN TRABS.CREDITO  
CAMBIOS EN TRABS. CREDITO.

## TERMINACION.

Cierra todos los archivos y termina el PROCESO DE ACCESO SECUENCIAL.

DESCRIPCION DEL ALGORITMO DE ACCESO SECUENCIAL EN ARCHIVOS INDEX SECUENCIAL.

Es el mismo algoritmo descrito en la sección de acceso secuencial en organización de archivos secuenciales.

ALGORITMO DE ACCESO  
NO SECUENCIAL INDEXADO

ORDENAMIENTO DE  
ARCHIVO MOVOTOS.

INICIALIZACION

PROCESO  
TARJETAS

CIFRAS DE  
CONTROL.

TERMINACION

ELIMINACION DE  
EMPRESAS

PRECESO TARJETAS  
EMPRESAS

PROCESO TARJETAS  
TRABAJADORES

PROCESO DE TARJETAS  
TRABAJADORES CON  
CREDITO

INSERTA  
EMPRESA

MODIFICA  
EMPRESA

INSERTA  
TRABAJADORES

MODIFICA O ELI-  
MINA TRABAJADO-  
RES

INSERTA  
TRABAJADORES  
CON CREDITO

MODIFICA O ELIMINA  
TRABAJADORES CON  
CREDITO

CHECAR SI  
EMPRESA  
ELIMINADA

MODIFICA  
TRABAJADORES

ELIMINA  
TRABAJADORES

CHECAR SI  
EMPRESA  
ELIMINADA

MODIFICA  
TRABAJADORES  
CON CREDITO

ELIMINA  
TRABAJADORES  
CREDITO

## DESCRIPCION DEL ALGORITMO DE ACCESO NO SECUENCIAL INDEXADO

Para mayor entendimiento de la descripción de este algoritmo, pondremos con letras mayúsculas -- los nombres de los bloques que se muestran en el diagrama jerárquico anterior y a continuación de -- tales la descripción del funcionamiento de ellos.

### ORDENAMIENTO DEL ARCHIVO MOVOTOS.

Aquí se ordena el archivo MOVOTOS, de tal manera que entren registros de EMPRESAS (a su vez ordenados por el NUMERO DE EXPEDIENTE), TRABAJADORES -- (que también irán ordenados por REGISTRO FEDERAL -- DE CAUSANTES) y de TRABAJADORES CON CREDITO (ordenados igual que los registros de TRABAJADORES) en este orden. El archivo ordenado lo llamaremos --- ARCHST.

### INICIALIZACION

Esta rutina realiza las siguientes funciones:

- 1.- Abre los archivos de ENTRADA, ENTRADA/SALIDA y SALIDA (ver diagramas de archivos de esta -- sección)

- 2.- Limpia la tabla TABLA-EMPS-ELIMS que contendrá los números de expediente de las EMPRESAS que se van a eliminar.
- 3.- LEE el primer registro del archivo MOVTOS para comenzar la ejecución de este algoritmo.

#### PROCESO TARJETAS.

Esta rutina se va a ejecutar hasta que el archivo ARCHST (que es el archivo ordenado de MOVTOS) se agote.

La función de esta rutina es ver que el tipo de registro es el leído en el archivo ARCHST y llamar a la rutina correspondiente. Los tipos de registro son los siguientes:

- 1.- Si TIP-REG = 0, se trata de un registro de EMPRESA, la cual va a ser eliminada.
- 2.- Si TIP-REG = 1, el registro es de EMPRESAS, la cual va a ser modificada o dada de alta.
- 3.- Si TIP-REG = 2, el registro es de TRABAJADORES.
- 4.- Si TIP-REG = 3, el registro es de TRABAJADORES CON CREDITO.

#### ELIMINACION DE EMPRESAS

Esta rutina lo que hace es eliminar el regis

tro de EMPRESA indicado del archivo de EMPRESAS.

#### PROCESO TARJETAS EMPRESAS.

Las funciones de esta rutina son las siguientes:

- 1.- Si no existe la EMPRESA que menciona el registro actual del archivo ARCHST, se da alta con su información correspondiente por medio de la rutina INSERTA EMPRESA.
- 2.- Si existe la empresa que menciona el registro actual del archivo ARCHST, se actualiza su información correspondiente por medio de la rutina MODIFICA EMPRESA.

#### PROCESO DE TARJETAS DE TRABAJADORES

Esta rutina se encarga de procesar los registros de TRABAJADORES que vengan en el archivo - ARCHST.

#### INSERTA TRAB.

Si no existe el TRABAJADOR que indica el registro actual del archivo ARCHST, se da de alta -- con su información correspondiente y actualiza el registro de su EMPRESA.

## MODIFICA O ELIMINA TRAB.

Esta rutina hace las siguientes funciones:

- 1.- Checa que la EMPRESA del TRABAJADOR exista o no exista y tal información la pasa a través de una bandera, ésta función la realiza a través de la rutina CHECAR SI EMPRESA ELIMINADA.
- 2.- Si existe la EMPRESA de dicho TRABAJADOR, se modifica la información del TRABAJADOR en el archivo TRAB. (archivo de trabajadores) a través de la rutina MODIFICA TRAB.
- 3.- Si no existe la EMPRESA de dicho TRABAJADOR, no tiene caso que se encuentre al trabajador en el archivo TRAB. y se elimina a éste por medio de la rutina ELIMINA TRAB.

## PROCESO DE TARJETAS DE TRABAJADORES CON CREDITO.

Esta rutina se encarga de procesar los registros de TRABAJADORES CON CREDITO que traiga el archivo ARCHST (que es el ordenado de MOVTOS).

## INSERTA TRAB. CREDITO.

Si no existe el TRABAJADOR que indica el registro actual del archivo ARCHST, se da de alta -- con su información correspondiente en el archivo --

TRABC (archivo de TRABAJADORES CON CREDITO) y actualiza los registros de TRABAJADOR y EMPRESA en los datos afectados.

#### MODIFICA O ELIMINA TRAB-CREDITO.

Esta rutina realiza las siguientes funciones:

- 1.- Checa que la EMPRESA DEL TRABAJADOR CON CREDITO exista o no exista, esta información la pasa a través de una bandera a las rutinas que la requieran, por medio del bloque de instrucciones llamado CHECAR SI EMPRESA ELIMINADA.
- 2.- Si existe la EMPRESA de dicho TRABAJADOR CON CREDITO, se modifica su información en el archivo TRABC por medio de la rutina MODIFICA - TRAB-CREDITO.
- 3.- Si no existe la EMPRESA de dicho TRABAJADOR CON CREDITO, se elimina del archivo TRABC debido a que es ilógico que exista en tal. Lo anterior se realiza por medio de la rutina -- ELIMINA TRAB-CREDITO.

#### CIFRAS DE CONTROL.

Esta rutina saca las siguientes cifras de con

trol:

ALTAS EN EMPRESAS

BAJAS EN EMPRESAS

CAMBIOS EN EMPRESAS

ALTAS EN TRABAJADORES

BAJAS EN TRABAJADORES

CAMBIOS EN TRABAJADORES

ALTAS EN TRABS. CON CREDITO

BAJAS EN TRABS. CON CREDITO

CAMBIOS EN TRABS. CON CREDITO.

TERMINACION.

Cierra todos los archivos y termina el PROCESO DE ACCESO NO SECUENCIAL INDEXADO.



## DESCRIPCION DEL ALGORITMO DE ACCESO HASHING

Es el mismo algoritmo descrito en la sección de acceso no secuencial indexado.

## CONCLUSIONES

Partiendo de nuestro objetivo podemos ver que la optimización de la organización de una base de datos está unida a los siguientes aspectos.

Primero.- De acuerdo al sistema de información al cual va hacer implementada como - vimos en la introducción ya que existen sistemas en los cuales va a ser más importante la rapidez con que la información pueda ser proporcionada, O bien el volumen de información que se va a manejar.

Segundo.- Otro aspecto sería considerar la volatilidad de la información que repercutirá en la razón de crecimiento de los archivos.

Tercera.- La diversidad de usos que se va a tener.

Se pueda evitar la redundancia de información, facilidad de acceso siendo transparente para ellos cualquier complejidad interna, que los datos podrán ser utilizados de manera flexible que la base de datos pueda crecer y variar sin interferir con las -- formas ya establecidas de utilización, por lo tanto

para poder optimizar una base de datos, tendremos - que tomar en cuenta cada uno de los elementos que - la constituyen como son la organización tanto físi- ca como lógica de los datos y los métodos de acceso.

Como vimos en el capítulo IV centramos nuestro análisis al estudio de las técnicas de acceso que - consideramos una de las partes más tangibles dentro de la optimización de una base de datos, utilizando cuatro técnicas que son la secuencial, secuencial - indexada, no secuencial indexada y desmenuzamiento (hashing).

#### TIPOS DE BASES DE DATOS EN FUNCION DE SU USO.

De los puntos 1 y 2 podemos decir que depen--- diendo del uso que vaya a tener la base de datos se pueden clasificar en tres grupos principales que -- son:

- De Archivo Maestro
- De Archivo Histórico
- De Registro Dinámico.

Del resultado de las pruebas realizadas y de - las características de estos tipos de base de datos se concluye que:

El archivo maestro tiene muchos cambios y las altas y bajas son poco frecuentes por lo que para -

este tipo de base de datos se recomienda se usen -- una técnica de acceso no secuencial indexada si el volumen de cambios es menor al 50% debido a que las altas y bajas no repercuten en la eficiencia.

Si tomamos en cuenta las características del sistema de cómputo y vemos que el cuello de botella se encuentra en los acceso a disco, por ejemplo un sistema de cómputo con configuración de disco en serie, es conveniente el uso máximo de CPU por lo que se recomienda usar la técnica de acceso de hashing hasta un volumen del cambio del 60% y técnica de -- acceso secuencial para volumen de cambios mayor al 60%.

Si el sistema de cómputo tiene un clock de baja frecuencia y varias vías de acceso a disco entonces es recomendable usar los resultados de la gráfica de cambios I/o donde vemos que para un volumen de cambios menor al 58% es recomendable usar método No secuencial Indexado y para bases con más del 58% de volumen de cambios se usaría la técnica de acceso secuencial.

El tipo de base de datos de archivo histórico como su nombre lo indica no permite bajas, por su estructura no admite cambios solo altas, por lo que para saber que técnicas de acceso se pueden usar necesitamos basarnos en las gráficas de donde vemos --

que no influyen las características del sistema de cómputo pues siempre la técnica de acceso de no secuencial indexado es óptima tanto para tiempo de I/O como para tiempo de CPU donde casi va a la par con la técnica de acceso hashing pero se compensaría con los I/O básicos.

El tipo de base de datos de registro dinámico donde las altas son muy frecuentes, las bajas frecuentes y las modificaciones son permitidas por lo que se tienen que usar todas las gráficas obtenidas para saber la técnica de acceso más recomendable.

Haciendo caso omiso de las características del sistema de cómputo podemos ver que basados en la gráfica de tiempos totales para volúmenes bajos se usaría la técnica de acceso hashing para volúmenes medios la técnica "No secuencial Indexado" y para volúmenes altos la técnica Secuencial.

## B I B L I O G R A F I A

1. DONALD E. KNUTH  
THE ART OF COMPUTER PROGRAMMING  
VOL. 1. FUNDAMENTAL ALGORITHMS  
ED. ADISSON-WESLEY, 1973
2. NICLAUS WIRTH  
ALGORITHMS + DATA STRUCTURES = PROGRAMS  
ED. PRENTICE HALL, 1976
3. JAMES MARTIN  
COMPUTER DATA-BASE ORGANIZATION  
ED. PRENTICE HALL, 1977
4. CONCEPTS OF MANAGE DATA-BASE DESIGN  
ED. COMPUTER SCIENCES CORPORATION, 1979
5. C WEST CHURCHMAN  
EL ENFOQUE DE SISTEMAS  
ED. DIANA, 1973
6. S. ATRE  
DATA BASE: STRUCTURED TECHNIQUES FOR  
DESIGN, PERFORMANCE AND MANAGEMENT  
ED. JOHN WILEY & SONS, 1980
7. THOMAS C. BARTEE  
DIGITAL COMPUTER FUNDAMENTALS  
ED. Mc. GRAW-HILL KOGAKUSHA, 1977
8. JOHN J. DONOVAN  
SYSTEMS PROGRAMMING  
ED. Mc. GRAW-HILL KOGAKUSHA
9. DANIEL MARTIN  
DATA-BASE DESIGN AND IMPLEMENTATION  
ON MAXI-AND-MINI COMPUTERS  
ED. VAN NOSTRAND REINHOLD
10. WILLIAM H. INMON  
EFFECTIVE DATA BASE DRSIGN  
ED. PRENTICE HALL
11. MORDICK ROBERT G.  
SISTEMAS DE INFORMACION BASADOS EN  
COMPUTADORA PARA LA ADMINISTRACION MODERNA  
ED. DIANA, 1974
12. FLORES IVAN  
DATA STRUCTURES AND MANAGEMENT  
ED. PRENTICE HALL, 1970

13. WIEDERHOLD GIO  
DATA BASE DESIGN  
ED. Mc. GRAW-HILL, 1977
14. LUCAS HENRY  
INFORMATION SYSTEMS CONCEPTS FOR MANAGEMENT  
ED. Mc. GRAW-HILL, 1978