

Tesis No 1

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE CIENCIAS



**CREACION DE UNA TABLA PARA EL MODELO
RELACIONAL DE BASE DE DATOS Y EXPLO-
TACION DE LA INFORMACION QUE CONTIENE**

T E S I S
QUE PARA OBTENER EL TITULO DE
M A T E M A T I C O
p r e s e n t a
MIREN BEGONA ALBIZURI ROMERO



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

TESIS CON FALLA DE ORIGEN

INDICE

	pág.
INTRODUCCION.	1
A) Funciones de un sistema de información.	
B) Aplicaciones de estos conceptos a ENVIR.	
C) Objeto de la tesis.	
CAPITULO I ANALISIS SINFACTICO.	14
CAPITULO II PREPARACION DE PARAMETROS EN EL DISEÑAMIENTO EN LA BASE DE DATOS AL TIEMPO DE CREACION.	19
CAPITULO III ALMACENAMIENTO DE INFORMACION COMPRESIDA, COMPACTADA Y RELACIONADA.	24
CAPITULO IV CONSULTA.	38
A) Cuantos.	
B) Lista.	
C) Ordena.	
CAPITULO V MINIMA	47
CAPITULO VI CORRECCION	51

CAPITULO VII RESUMEN. 62

APENDICE A MANUAL DE ENVIO EN ESPañOL PARA
EL USUARIO. 66

APENDICE B EJEMPLO CREADO INTERACTIVAMENTE
PARA ILUSTRAR EL MANEJO DE ENVIO.

BIBLIOGRAFIA.

INTRODUCCION

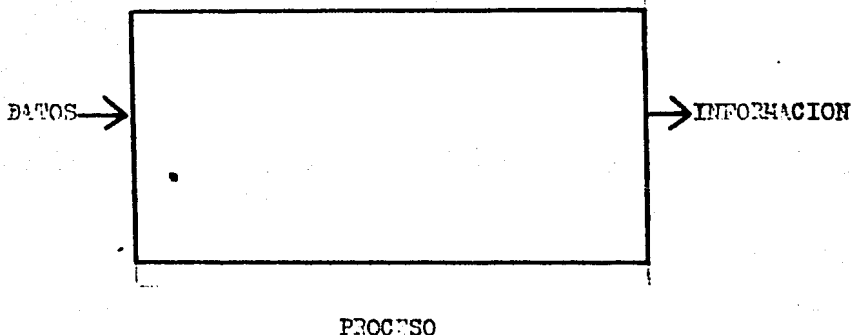
INTRODUCCION

El gran desarrollo alcanzado por las organizaciones en la actualidad, demanda una enorme cantidad de información. Por otro lado las empresas de nuestros días están obligadas a tomar decisiones cada vez más precisas y con mayor rapidez.

En la actualidad los responsables de tomar decisiones muchas veces cuentan con información inadecuada para sus fines; generalmente esta información resulta superflua, incompleta, poco clara, demasiado voluminosa; o se recibe demasiado tarde para ser de utilidad. Por ejemplo, un reporte semanal de 250 páginas que describa en detalle los cobros efectuados por una empresa no será un reporte útil para el gerente de créditos y cobranzas para que pueda decidir si su departamento operó eficientemente. Podemos ver entonces que la información útil será aquella que permita al responsable de tomar decisiones formarse con suficiente anticipación una idea clara y completa de la situación, en forma tal que pueda tomar objetivamente las decisiones convenientes.

Es importante aclarar la diferencia que existe entre información y datos. La diferencia básica consiste en que los datos son útiles o significativos como tales hasta que son procesados y convertidos a una forma útil llamada información. La siguiente figura nos muestra la relación que se establece entre datos e información a través de un proceso. El conjunto esquematiza un sistema de información.

SISTEMAS DE INFORMACION



Podemos considerar la información como el conocimiento derivado del análisis de los datos. Es importante notar que la información obtenida de un proceso puede servir como dato para otro proceso. Por ejemplo, en un sistema de nóminas se tienen como datos las horas trabajadas por cada empleado y su sueldo por hora. La información obtenida por el sistema será: los sueldos semanales de cada empleado. Esta misma información podrá servir como dato de otro sistema que determine los costos semanales por departamento.

La actividad o fenómeno que modifica un insumo para obtener un producto diferente por medio de un procedimiento establecido, constituye un proceso. Por ejemplo, un empleado que lee el contenido de una factura y de acuerdo con éste hace un asiento en la cuenta de deudores, está procesando información, una computadora que lee los datos del tiempo trabajado por los empleados y que calcula los sueldos de estos, también procesa información.

Resumimos, diciendo que el proceso de datos consiste en la transformación de insumos (datos, los cuales no son útiles por sí mismos) a través de un proceso, a fin de obtener un producto (información) que sí puede ser útil sin transformaciones ulteriores. Tal proceso estará integrado a un sistema diseñado para obtener información necesaria en la toma de decisiones.

Por otra parte, se habrá establecido el sistema que permita captar los datos necesarios y la forma como deberán ser procesados para lograr la información que se requiere.

Un "sistema" es un conjunto de elementos y procedimientos íntimamente relacionados que tienen como propósito el logro de determinados objetivos. Así, un sistema de información es el conjunto de elementos y procedimientos íntimamente relacionados que tienen como propósito manejar datos y elaborar reportes que permitan tomar decisiones adecuadas para el logro de los objetivos de una organización. Tales sistemas, pues, tienen como fin registrar, procesar y reportar información significativa; en otras palabras, constituyen un medio a través del cual es posible obtener información que nos permita elegir cursos concretos de acción; o sea, tomar decisiones.

Toda información manejada en cualquier tipo de sistema es pasada a través de una entrada, un proceso y una salida.

Una vez que los datos han sido recolectados y convertidos (si es menester) a una forma adecuada para su proceso, se hace necesario transmitirlos desde uno o más puntos de recolección, a una operación de proceso; esta función se denomina "de entrada".

La expresión transmisión de datos es usada para referir el movimiento de ellos, desde una localización a otra. Después que los datos han sido recibidos en el área de proceso, precisa organizarlos con el uso de registros, archivos, etc., para que puedan ser procesados en forma ordenada. Una buena organización facilita el manejo y proceso de datos.

El hecho de procesar datos implica dos aspectos: manipulación y cálculo.

El primero se refiere al manejo lógico y la forma de ordenar los datos; el segundo, a los procesos matemáticos que se deban realizar sobre los mismos.

Cuando se maneja información, debemos tener en cuenta las propiedades de cantidad, calidad y costo de proceso, partiendo de los datos que indican el flujo de un sistema de información. Cuando son transformados en información, los datos pueden perder o ganar algún grado de estas propiedades (calidad, cantidad o costo). Por ejemplo, la cantidad de datos puede decrecer durante el proceso; la calidad, disminuir por los errores de proceso, y el costo incrementarse por el esfuerzo requerido para producir información.

Después que los datos han sido procesados, se distribuye la información resultante, a los usuarios, que pueden ser localizados en puntos lejanos al lugar de proceso; entonces, las mismas estaciones y medios que se usaron para recolectar sus datos, pueden ser usados para transmitir la información requerida. A esto se le llama funciones de salida.

FUNCIONES DE UN SISTEMA DE INFORMACION

Independientemente del tipo de sistema, existen funciones comunes desarrolladas por todo sistema de información; estas son:

1. Recolección de datos-fuente. O sea, la forma en que son captados los datos-fuente que más adelante, evaluados y analizados por el sistema, nos permitan obtener la información necesaria para la toma de decisiones.

Esta recolección, inicio de las operaciones del proceso de datos de cualquier sistema de información, nos hace ver la necesidad de determinar más adecuadamente los datos que deben ser recolectados y los mecanismos que habrán de hacerlo.

2. Conversión de datos. Esto es, el cambio de código original en que están los datos, a un código acorde con los medios de proceso y almacenamiento del sistema.

Muchas veces los mecanismos utilizados para el proceso hacen necesario que los datos sean convertidos desde su representación original que, por lo general, es objetiva al hombre, a una representación tal que permita su proceso a través de los mecanismos utilizados, los cuales no son afines a la objetividad del hombre.

3. Transmisión de datos. Este es el proceso de mover datos desde una localización a otra, físicamente.

En todo sistema de información es muy importante definir claramente la forma en que son transmitidos los datos de un lugar a otro, ya que por regla general esta función es la que demanda más

tiempo en el proceso de los sistemas de información.

4. Almacenamiento de datos. Se refiere a la forma en que es almacenada la información.

Esta función determina en primera instancia el carácter integral de un sistema de información; es decir, depende del tipo de almacenamiento, en primer término, establecer qué tan integrado será un sistema de información.

5. Proceso sobre datos. Forma de efectuar tanto operaciones lógicas como matemáticas, de tal manera que producen los resultados requeridos por un sistema de información a partir de los datos que le son alimentados. El proceso de datos se vale de diversos mecanismos, y la naturaleza de los sistemas de información queda determinada por esos mecanismos usados en el proceso.

6. Recuperación de información y reportes. Esta función es la de recuperar información dada con anterioridad, en forma aislada y desordenada, la cual ha de ser reportada a través de formatos diseñados para ayudar en la toma de decisiones.

Resulta importante contemplar que la salida o reporte de información vaya acorde con la velocidad de todas las funciones (anteriormente citadas) que integran un sistema de información.

Básicamente existen cuatro diferentes tipos de sistemas de información: manual, mecanizado, electromecánico y electrónico.

Es claro que las técnicas para el tratamiento de información, como las herramientas que en ellos se emplean, no son sino instrumentos al servicio de la administración y cuyo éxito depende,

en última instancia, del elemento humano, a quien siempre deberán subordinarse.

La computadora, como herramienta al servicio de la administración, cuenta con componentes que permiten alimentación de la materia prima, en nuestro caso datos; componentes que desarrollan todas las operaciones requeridas en el proceso y, finalmente, componentes que dan salida al producto terminado; en nuestro caso, información (sin que por lo anterior se eliminen los componentes de retroalimentación que pueden ser muchos y muy variados).

Los sistemas electrónicos de información son diseñados para aplicaciones en las cuales grandes masas de datos deben ser recolectadas y analizadas, con el fin de reportar información significativa. Tales sistemas son también usados para manejar casos en los cuales los resultados de un proceso deben ser comparados para determinar reglas decisivas como una base para simples decisiones (por ejemplo, cuándo y qué cantidad debe ser reordenada de un producto cuyo nivel de inventario ha bajado del mínimo especificado?). La alta velocidad de operación y la gran capacidad de almacenamiento de los sistemas electrónicos permite a estos manejar grandes volúmenes de datos y complejos procesos, económica y eficientemente.

El sistema DIVER está basado en la teoría de conjuntos. Esto permite que los datos sean almacenados en forma compacta y que estos sean recuperados rápidamente por medio de cálculos. La teoría de conjuntos nos provee de una herramienta simple, poderosa y práctica para el manejo de información sobre objetos.

El estructuramiento de objetos en conjuntos y la estructura jerárquica de conjuntos en otros conjuntos es el fundamento de todas nuestras matemáticas.

El primer postulado sobre el álgebra lógica fué publicado por George Boole a mediados del siglo XIX. Estos conceptos han sido extendidos por otros investigadores a varios contextos conocidos por los nombres de álgebra booleana, cálculo proposicional, lógica aritmética y muchos otros. Dentro del contexto del manejo de datos el álgebra booleana es la técnica usada para combinar y alterar conjuntos para formar nuevos conjuntos. El uso del álgebra booleana con conjuntos es comparable al uso de la aritmética ordinaria con números.

APLICACIONES DE ESTOS CONCEPTOS A ENVIR

Nuestras computadoras de hoy trabajen aritmética ordinaria con números. Este tipo de operaciones, en realidad, se componen de una serie de operaciones booleanas más simples sobre los bits que representan a estos números.

Como las computadoras, a nivel circuito, son máquinas de álgebra booleana, a nivel programación el usuario puede ejecutar operaciones booleanas sobre variables de su propia elección. Esta es la propiedad fundamental de las computadoras que usan el sistema ENVIR.

El almacenamiento de datos en el sistema ENVIR se basa en percibir si un objeto pertenece o no a un conjunto particular. Donde estos conjuntos tienen como elementos únicamente objetos que pueden

ser descritos de una manera particular de acuerdo a una característica dada.

Una gran cantidad de información sobre cada objeto puede ser almacenada en una forma muy compacta por medio de una cadena de ceros y unos. Esto es particularmente importante ya que la computadora está diseñada para manejar rápidamente estas cadenas de "bits".

Este sistema de almacenamiento es también ventajoso para la recuperación selectiva de datos; es decir, en el lenguaje de la teoría de conjuntos, escoger un subconjunto particular de todo el banco de datos -conjunto universal-. Usando el lenguaje del álgebra booleana, uno puede escoger aquellos elementos que forman otro conjunto cuyos elementos pueden tener ambas o una o la otra de dos características dadas o la opuesta de una característica dada.

Estas operaciones son particularmente fáciles y rápidas de realizar en una computadora cuando se trabaja con cadenas de bits. Por ejemplo, la función opuesta (o negación) requiere simplemente el cambiar todos los ceros a unos y viceversa.

OBJETO DE LA TESIS

ENVIR (ENVIRONMENTAL INFORMATION & RETRIEVAL SYSTEM) fue realizado por GULF UNIVERSITIES RESEARCH CONSORTIUM al que pertenece la Universidad Nacional Autónoma de México (UNAM) y adquirido por ésta en 1974 como "caja negra" para tener Sistemas de Información a disposición de los investigadores, profesores, funcionarios y empleados.

dos de la URM sobre una BASE DE DATOS del modelo relacional; sin embargo muchos de los usuarios tuvieron problemas para usarlo al presentárseles fallas técnicas como límites de capacidad, eficiencia en tiempos de respuesta, eficiencia en uso de disco, etc., y por otra parte no les gustaba el operarlo en inglés estando en México y descifrar códigos de error.

Por ésto, el objeto de la presente tesis consiste en presentar la tecnología básica de una Base de Datos del modelo relacional, para desarrollar fácilmente Sistemas de Información, utilizando como ejemplo a ENVIR que tiene como ancestro al EXIR.

El Sistema EXIR (EXECUTIVE INFORMATION RETRIEVAL) fué realizado por "EPO Biology Department" de la Universidad de Colorado para usarse en complejas bases de datos biológicas. Consiste de módulos, cada uno de ellos con una función específica. El módulo central es la información almacenada y el sistema recuperador el cual es llamado TAXIR/STERS. Este módulo permite al usuario operar con sus datos haciendo uso de consultas en lenguaje natural.

Los otros módulos mencionados son de dos tipos generales, los módulos de formato de datos (DF) y los módulos de síntesis-análisis-muestra (SAD).

El Sistema ENVIR está basado en la teoría de conjuntos, esto permite almacenar los datos en forma compacta y recuperarlos rápidamente.

Dada una colección de conjuntos D_1, D_2, \dots, D_n (no necesariamente distintos), R es una relación entre estos n conjuntos si hay

un conjunto de n-tuplas ordenados (d_1, d_2, \dots, d_n) tal que d_1 pertenece a D_1 , d_2 pertenece a D_2, \dots, d_n pertenece a D_n . Los conjuntos D_1, D_2, \dots, D_n se llaman dominios de R, n es el grado de R.

La siguiente tabla muestra una relación llamada "Partes" de grado 5, los dominios son: #P, Pnombre, Color, Peso, Ciudad.

PARTES

#P	Pnombre	Color	Peso	Ciudad
P1	tuerca	rojo	12	Londres
P2	perno	verde	17	París
P3	tornillo	azul	17	Roma
P4	tornillo	rojo	14	Londres
P5	clavo	azul	12	París
P6	tachuela	rojo	19	Londres

Como se muestra en la figura es conveniente representar una relación como una tabla.

Cada renglón de la tabla representa una n-tupla (o simplemente una tupla) de la relación. El número de tuplas en la relación se llama la cardinalidad de la relación, la cardinalidad de la relación Partes es seis.

Podemos definir el modelo relacional de una base de datos como una colección de relaciones de diferentes grados y diferente car

dinalidad.

La siguiente figura muestra un modelo relacional consistente de tres relaciones : S (vendedores), P (Partes) y SP.

S					SP		
#S	Snombre	Estado	Ciudad		#S	#P	Qty
S1	Ruiz	20	Londres		S1	P1	300
S2	Sanchez	10	Paris		S1	P2	200
S3	Vázquez	30	Paris		S1	P3	400
S4	Juárez	20	Londres		S1	P4	200
S5	Ramírez	30	Atenas		S1	P5	100
					S1	P6	100
					S2	P1	300
					S2	P2	400
					S3	P2	200
P							
#P	Pnombre	Color	Peso	Ciudad	#S	#P	Qty
P1	tuerca	rojo	12	Londres	S4	P2	200
P2	perno	verde	17	Paris	S4	P4	300
P3	tornillo	azúl	17	Roma	S4	P5	400
P4	tornillo	rojo	14	Londres			
P5	clavo	azúl	12	Paris			
P6	tuercas	rojo	19	Londres			

De aquí en adelante se describirá al sistema ENVIR, habiendo agrupado en 6 capítulos la documentación técnica y 2 apéndices; a saber:

CAPITULO I.- ANALISIS SINTACTICO

CAPITULO II.- PREPARACION DE PARAMETROS EN EL DIMENSIONAMIENTO EN LA BASE DE DATOS AL TIEMPO DE CREACION.

CAPITULO III.- ALMACENAMIENTO DE INFORMACION COMPRESIDA, COMPACTADA Y RELACIONADA.

CAPITULO IV.- CONSULTA.

CAPITULO V.- ELIMINA.

CAPITULO VI.- CORRECCION.

APENDICE A.- MANUAL DE ENVIR EN ESPAÑOL PARA EL USUARIO.

APENDICE B.- EJEMPLO CREADO INTERACTIVAMENTE PARA ILUSTRAR EL MANEJO DE ENVIR.

CAPITULO I.

ANALISIS SINTACTICO.

ANÁLISIS SINTÁCTICO

El análisis sintáctico se realiza en la subrutina `Knfse`, la cual a su vez requiere de los procedimientos auxiliares: `Setinx`, `Pasblk`, `Sname` y `Dname`, según se explica mas adelante.

`Knfse` utiliza 3 arreglos:

1.-) `Sname`.

Almacena los comandos permitidos en orden ascendente con respecto a la longitud en caracteres de éstos.

2.-) `Slength`.

En éste arreglo se encuentra la diferencia de caracteres de un comando con respecto al anterior, considerando el orden en `Sname`, ésto es, el j -ésimo comando tendrá como longitud $\sum_{i=1}^j Slength(i)$, donde `Slength(1)` es la longitud del primer comando.

3.-) `Stadd`.

Por medio de `Stadd` se obtiene la palabra de `Sname` en la cual empieza el comando, de tal forma que si `Stadd(i)=j` entonces el i -ésimo comando empieza en `Sname(j)`.

El registro de entrada es almacenado en "input", en el cual se guarda un caracter por palabra.

El registro es leído e impreso en `Setinx`, en donde también es detectado el fin de archivo y actualizado el apuntador de "input".

En Pasblk son ignorados los blancos para obtener un caracter diferente a éstos.

Se utiliza un arreglo auxiliar "name" en el cual se guarda una copia de lo parcialmente tomado del registro de entrada, juntando los caracteres (seis caracteres por palabra), esto se hace en Bname, para comparar contra las palabras clave (keywords) que rigen la operación de la Base de Datos.

En Zname se limpia "name" con ceros.

Al iniciarse el análisis, a la variable "st-tex" se le asigna el valor de uno. Tomando el número de caracteres del primer comando (Slength(1)) se considera éste mismo número de caracteres del registro de entrada, éstos se comparan con el comando, si son iguales es aceptado el registro de entrada, si no se incrementa "st-tex" para así apuntar al comando siguiente.

Si el valor correspondiente en "slength" es cero se consideran los mismos caracteres del registro de entrada anteriores para compararlos con el comando apuntado por "st-tex".

Si el valor de "slength" es diferente de cero se consideran los caracteres anteriores más los indicados por "slength" del registro de entrada para compararlos.

Si aún no son iguales "st-tex" vuelve a ser incrementado y se repite el procedimiento.

La subrutina Infse tiene dos posibles terminaciones:

Cuando el registro de entrada es igual al comando apuntado por "st-tex" en "Bname" o cuando "st-tex" es mayor a "nosn" (número

de comandos permitidos), lo cual indica que el registro de entrada no fué encontrado en "sname" por lo que no es aceptado mandándose un mensaje de error.

Cuando el registro es aceptado Knfse regresa el valor de "stater" el cual corresponde al número del comando en "sname" al cual se refiere el registro, para así transferir la operación del comando deseado a la parte que le corresponde en el Sistema, es decir, según el valor de "stater" el control del programa va al procedimiento requerido para llevar a cabo la función del comando.

La tabla siguiente muestra las palabras clave que rigen la operación de la Base de Datos y la forma de colocarlos lógicamente para identificación sintáctica; también se presenta la instrucción de transferencia de control a las etiquetas que atienden los comandos según el comando elegido correspondiente al índice STATEX.

SEATEX	SMNTP	STNGPTU	SEADD
1	fin	3	1
2	a'to	1	2
3	coma	0	3
4	nota	0	4
5	abajo	1	5
6	lista	0	6
7	vacía	0	7
8	arriba	1	8
9	codigo	0	9
10	depura	0	10
11	titulo	0	11
12	cuantos	1	12
13	literal	0	14
14	sistemas	1	16
15	lee banco	1	18
16	corrección	1	20
17	idem=falso	0	22
18	interactivo	1	24
19	bbbbbbbbbbbb	1	26
20	decimal=libre	1	28
21	escribe banco	0	31
22	idem-verdadero	1	34
23	ordena y lista	0	37
24	agrega dominios	1	40
25	oooooooooooo	1	43
26	elimina registros	1	46
27	envia a la salida	0	49
28	redefine dominios	0	52
29	oooooooooooo	1	55
30	agrega registros de	1	58
31	mensaje a la salida	0	62
32	selecciona dominios	0	66
33	dddddddddddd	1	70
34	lee comandos de cinta	1	74
35	oooooooooooo	1	78
36	ffffffffffff	1	82

continuación

STATEX	SHARE	SINGTH	STADD
37	#####	1	86
38	estructura de la relacion	1	90
39	ordena y envia a la salida	1	95
40	agrega y lista registros de	1	100

GO TO (475,2024,2001,21,2049,11,22,2050,2021,9999,17,11,2005,9998,
33,11,2004,2012,476,2003,11,2000,11,11,476,11,11,11,476,11,
2014,44,476,2006,476,476,476,476,11,11,11), STATEX

CAPITULO II.

PREPARACION DE PARAMETROS EN EL
DIMENSIONAMIENTO EN LA BASE DE
DATOS AL TIEMPO DE CREACION.

PREPARACION DE PARAMETROS

EN EL DIMENSIONAMIENTO EN LA BASE DE DATOS AL TIEMPO DE CREACION

Para el almacenamiento y manipulación de la Base de Datos ENVIR utiliza dos archivos : el archivo compactado o directorio, en el cual los estados son almacenados una sólo vez aunque hayan sido declarados varias veces y el archivo comprimido, en el cual se guardan los registros en forma binaria con referencias al directorio.

Con el comando "Selecciona Dominios" ENVIR hace la reservación del área que necesitará para el almacenamiento de la base, en el archivo compactado y crea los apuntadores por medio de los cuales se van a relacionar las diferentes áreas que utilizará posteriormente para realizar las operaciones con los datos.

El número de descriptores definidos para la base no puede exceder al valor del parámetro N(400). El cual se puede cambiar manualmente en el programa principal al igual que todos los parámetros; entre paréntesis se muestra el valor asignado en el programa.

Una vez que se ha determinado el número de descriptores que se utilizarán ("nodde") se prosigue a analizar las características de cada uno de ellos.

Los nombres de los descriptores no pueden repetirse, para ver que esta condición se cumpla la subrutina Orchdd hace una búsqueda binaria, los descriptores se almacenan en "DD" por orden alfabético, utilizando éste procedimiento guarda el descriptor en or

den, o sea, el almacenamiento y ordenamiento de los descriptores es simultáneo.

Puesto que generalmente los descriptores no quedan almacenados en el mismo orden en que fueron definidos se requiere de un control de éste lo cual se hace con "Dperm", que relaciona el orden de entrada con el ordenamiento alfanumérico.

Si el nombre del descriptor contiene más de seis caracteres o NCIV, del carácter séptimo en adelante se almacena en "ovrfl", siendo "ovrdd" igual al índice de "ovrfl" en donde fué guardado el sobreflujo y "now" toma el valor del número de palabras de sobreflujo excluyendo la primera.

El número total de palabras de sobreflujo debe de ser menor o igual al parámetro D (5000).

Los descriptores pueden ser definidos como alfabéticos, numéricos o de código. Según esta definición "option" toma un valor (option=1 para alfabético, option=3 para numérico o código).

Si el descriptor es numérico se debe de especificar el rango en el cual los estados referentes a éste dominio se van a encontrar, el valor de la base o límite inferior se le asigna a "from" y el del tope o límite superior a "to". Pueden contener decimales y/o etiqueta calificadora; ambas definiciones son opcionales, el número de decimales se almacena en "idp" y la etiqueta (kilos, años, peso, etc) en "in".

El número de descriptores numéricos definidos no puede ser mayor al parámetro G (300).

Para cada descriptor alfabético se debe especificar el número máximo de estados diferentes posibles, el cual debe de ser menor o igual al parámetro B (1000). Este valor es asignado a "reserv".

Un descriptor puede ser declarado igual a uno anteriormente definido, en este caso ambos tendrán la misma opción y compartirán el área de almacenamiento en el archivo compactado pero no en el comprimido.

En "nostrg" se guarda el número de bits necesarios para representar los valores en el archivo comprimido de los registros.

"Reserv" indica cuántos registros se reservaron en el diccionario para alfabéticos, la diferencia entre "to" y "from" para numéricos, el número de opciones posibles para los de código.

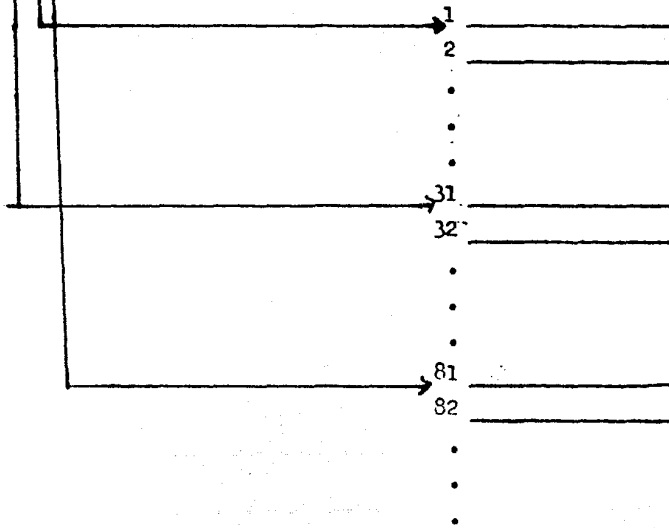
El procedimiento que sigue ENVIR para calcular "nostrg" está basado en el arreglo Bmask, el cual contiene potencias de dos, busca la primera potencia mayor (o igual) a "reserv", el índice en que fué encontrada menos uno indica el número de bits requeridos para representar éste número ($2^{\text{nostrg} - 1} = \text{reserv}$).

En "ifsa" se guarda la palabra a partir de la cual será almacenada la información de los estados del descriptor en el archivo comprimido. Este valor se obtiene sumando "ifsa" y "nostrg" del descriptor anterior.

La ilustración siguiente muestra las áreas asignadas y controles después de "Selecciona Dominios".

DD	NOSER	IFSA	ISSA	OTPA	MONTEH	RESERV	OPTION	NOW	OVADD
apelli	6	13	31	31	3	50	1	0	2
apelli	6	7	31	31	3	50	1	0	1
edad	7	19	0	1	3	0	3	0	0
especi	3	26	81	85	11	0	3	0	3
nombre	5	2	1	1	3	30	1	0	0

ARCHIVO COMPRIMIDO



El archivo comprimido es dimensionado por medio de dos parámetros: F y G.

El parámetro F se calcula sumando los valores de "nuestro" de todos los descriptores más uno. El parámetro G depende de la palabra de memoria de la computadora y de los discos que use; por ejemplo, si la memoria es de 48 bits y el disco usa segmentos de 30 palabras, el bloque óptimo debe ser múltiplo de 30 palabras pero debe existir un balance entre el uso de memoria del buffer del archivo y el tiempo de procesamiento, suponiendo que el buffer adecuado del archivo comprimido es 1800 palabras y que $F=28$, entonces $G = \text{tamaño buffer (bloque)} \div F = 1800 \div 28 = 64$, para utilizar en la mejor forma posible el disco; ahora el número de items que se cargan a memoria en cada lectura es $G \times \text{tamaño palabra de la computadora en bits} = 64 \times 48 = 3072$ registros.

Es conveniente que el valor del parámetro F sea mayor al obtenido por la forma anteriormente especificada, ya que ENVIR permite definir más descriptores después de que el banco ha sido creado y aún consultado o modificado, esto se puede hacer con el comando "ACRUEGA COMPTIOS".

Para optimizar la ejecución del programa, el usuario debe dimensionar los arreglos usados por ENVIR, en base a las características de los datos que serán almacenados en el banco y con consideración de las consultas que pueden ser anticipadas.

Los valores deben ser dados para el conjunto básico de los parámetros especificados en éste capítulo. Estos parámetros ENVIR los almacena en un programa "driver", el cual debe de ser recompilado cuando cualquiera de éstos es cambiado.

CAPITULO III

ALMACENAMIENTO DE INFORMACION
COMPRESIDA, COMPACTADA Y REA-
CIONADA.

ALMACENAMIENTO DE INFORMACION COMPACTADA Y RELACIONADA

Bajo la terminología del sistema ENVIR a cada elemento del banco de datos se le denomina un item. El item mismo es simplemente toda la información que describe a ese elemento.

Una vez que han sido definidos los descriptores, el segundo paso consiste en definir los items, es decir, preparar la matriz de datos como entrada para la computadora. Cada item está constituido por una cadena de estados de descripción correspondientes a cada uno de los descriptores de la lista definidos anteriormente.

Los estados de descripción deben estar en el orden en que fueron definidos los descriptores, si no se reordenan o ajustan a la presentación de los items. Además, los estados de descripción deben ser apropiados al tipo declarado para ese descriptor.

El sistema al recibir un estado de descriptor analiza que corresponda al tipo del que fué declarado y que el área reservada aún no se haya agotado.

Guarda el estado en "dss" en orden alfabético haciendo las modificaciones necesarias para ésto (movimientos de registros) e incrementa el valor de "dsea".

Si el estado recibido ya está almacenado por ser igual en otro item no se almacena por segunda vez, por lo que el directorio es un archivo compactado, ya que no hay repetición de información.

Con los estados al igual que con los nombres de los descriptores se puede tener sobreflujo, este es tratado en ambos casos

igual.

Como ya se mencionó anteriormente los estados son almacenados en orden alfabético, entonces el sistema para no perder el orden en que fueron declarados asigna éste número a "dscode". De tal forma que el estado que tenga el número 1 en dscode será el primer estado declarado.

Supongamos que en nuestro banco definimos un descriptor nombre, y tenemos 4 items declarados en el siguiente orden:

Luis

Ana

Juan

Ana

entonces ENVIR los almacenará de la siguiente forma:

DSS	DSCODE
ANA	2
JUAN	3
LUIS	1

Los valores de "dscode" son la información almacenada en el archivo comprimido.

Para la declaración de los items se usa el postulado

cinta

AGREGA REGISTROS DE disco

tarjs

el cual precede al que permite la descripción de los dominios.

Entonces al momento de definir los items que constituirán nuestro banco ya el sistema hizo la reservación de área especificada para los estados de cada descriptor.

En "ifsa" se tiene el apuntador al registro del archivo comprimido en donde la información referente al descriptor comienza y "nostrg" nos dice cuántos bits de éste se van a utilizar.

El sistema traduce los valores de "dscode" a binario y los guarda en ésta área en forma vertical de izquierda a derecha siguiendo el orden en que fueron declarados.

Con el ejemplo anterior el archivo comprimido sería

```
ifsa  1010
      0111
      0000
```

de tal forma que el primer número nos indica el valor de "dscode" del primer elemento declarado perteneciente al dominio, el segundo el "dscode" del segundo elemento y así sucesivamente.

Si un estado es numérico no se almacena éste valor sino que el resultado de restar el elemento menos la base del rango del descriptor más uno.

Cuando un descriptor fue definido de tipo código se debe especificar en la definición de éste la lista de elementos que lo constituyen. El primer nombre (de izquierda a derecha) en la lista de es-

datos tiene asignado por el sistema el número de código 1, el siguiente el 2, etc.

Cuando se declara un estado perteneciente a un descriptor de código el sistema almacena en el archivo comprimido el número de código que le fué asignado.

En el archivo compactado sólo se almacena la lista de elementos definidos en orden alfabético y "dscode" tiene el número de código.

Dentro de la información de los estados ENVIR guarda un número "dsxode". El número leído en el archivo comprimido indica que en "dsxode" de ese valor se encuentra el estado buscado.

Para el ejemplo anterior tenemos:

DSX	DSKODE	DSCODE	COMPRESO
Ana	3	2	1010
Juan	1	3	0111
Luis	2	1	0000

EJEMPLO 1

SELECCIONA DATOS 5

nombre(1 ALFA 30)

apellidopat(2 ALFA 50)

apellidomat(3=2)

edad(4 desde 15 a 30)

especialidad(5 CODIGO programador,analista,operador,otro)*

AGREGA Y MUESTRA REGISTROS DE DISCO

carlos, strassburger, frias, 35, operador*

victor, guerra, ortiz, 30, analista*

boris, dubin, rusenthal, 26, programador*

begonia, albizuri, ronero, 22, otro*

carlos, santamaria, ortiz, 21, otro*

miguel, cota, solis, 20, otro*

REGISTRO DE CONTROL

HODSS7	84	Area total reservada en el registro para almacenamiento de estados.
OVX	10	Palabras utilizadas en el registro de sobreflujo.
FTBI	1	Buffers utilizados.
HODDE	5	Número de descriptores declarados.
NOIF	28	Area total utilizada en el archivo comprimido
AMAX	30	
LMAIF	1	
LBAIF	6	
PMASK		
IBUF	1	
NODESC	5	Número de descriptores declarados.
DATE	100478	Fecha de creación de la base de datos.
TIME		Hora de creación de la base de datos.
EQUAME	1	Número de descriptores en los que se utilizó la igualdad.
LITRAI	0	
COMEA	107	Código de la coma (,) en EPCDIC.
IPIZN	1300	Dimensión del archivo comprimido.
J	3	
JJ	1	
JJJ	3	
H	0	
H	0	
L	0	

REGISTRO PARA EL ANÁLISIS LINEAL DE DATOS DE ESTADO

	DSS	EGOVAD	ESKON	DECODE	ESLON
1	begona	0	3	4	0
	boris	0	5	3	0
	carlos	0	2	1	0
	miguel	0	1	5	0
	victor	0	4	2	0
31	albizu	9	41	7	0
	cota	0	34	10	0
	dubin	0	35	5	0
	frias	0	36	2	0
	guerra	0	33	3	0
	ortiz	0	38	4	0
	ronero	0	31	8	0
	rusent	3	37	6	0
	santam	10	39	9	0
	solis	0	32	11	0
	strass	7	40	1	0
81	analis	5	34	2	0
	operad	6	31	3	0
	otro	0	32	4	0
	progra	4	33	1	0

REGISTRO DE ARMA DE SOBREFUJO

dopat

domat

alidad

mador

ta

or

burger

hall

ri

aria

REGISTRO DE RANGOS NUMERICOS

FROM	TO	IDP	IN
15	80	0	0

DEPERM

5

2

1

3

4

REGISTRO DE INFORMACION SOBRE DESCRIPTORES

DD	NOSTRG	IFSA	DSSA	DSEA	LENGTH	RESERV	OPTION	NOW	OVADD
apelli	6	13	31	42	12	50	1	0	2
apelli	6	7	31	42	12	50	1	0	1
edad	7	19	0	1	3	0	3	0	0
especi	3	26	81	85	11	0	3	0	3
nombre	5	2	1	6	6	30	1	0	0

LISTEVO DE NOMES

000000	
101011	
011000	
000101	1,2,3,4,1,5
000000	
000000	nombre
111110	
010101	
001100	
000011	1,3,5,7,9,10
000000	
000000	apellidonat
000001	
101001	
011010	
000101	2,4,6,8,11
000000	
000000	apellidonat
100010	
000011	
101011	
001100	
110000	21,16,13,2,7,6
000000	
000000	ed-d
101100	
110000	2,7,7,1,1,4
000111	apellidonat

ESTADOS Y NOMBRES DE DEMONSTRACIONES REFERENCIADOS AL AREA DE SOBREVIVUJO

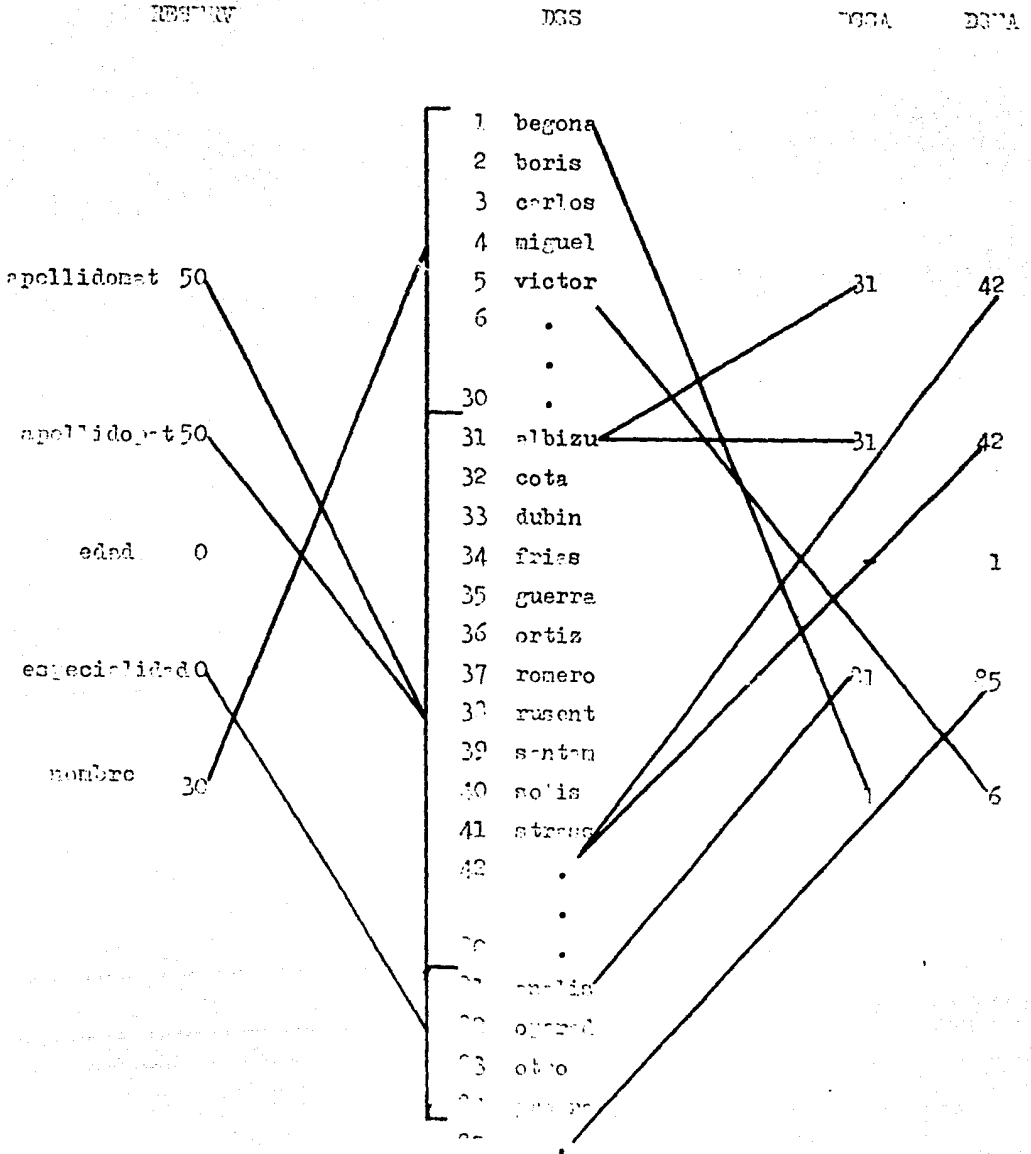
DSI	DSOVAD	SORRELIUJO	OVADO	DS
begona	0	1 donat	2	apelli
boris	0		1	apelli
carlos	0	2 donat	0	edad
miguel	0		3	especi
victor	0	3 alidad	0	nombre
.				
.		4 mador		
.				
albizu	9			
cota	0	5 ta		
dubin	0	6 or		
frias	0			
guerra	0	7 burger		
ortiz	0			
romero	0	8 hall		
rusent	8			
santam	10	9 ri		
solis	0			
stress	7	10 aria		
.				
.				
.				
enclis	5			
operad	6			
otro	0			
PROGR	4			

RELACIONES ENTRE EL ARCHIVO COMPACTO Y LOS DOMINIOS DEL DIRECTORIO

DD	ITSA	ARCHIVO COMPACTO	NOTAS
1 apelli	13	1	000000
		2	101011
		3	011000
2 apelli	7	4	000101
		5	000000
		6	000000
3 edad	19	7	111110
		8	010101
		9	001100
4 especi	26	10	000011
		11	000000
		12	000000
5 nombre	2	13	000001
		14	101001
		15	011010
		16	000101
		17	000000
		18	000000
		19	100010
		20	000011
		21	101011
		22	001100
		23	110000
		24	000000
		25	000000
		26	101000
		27	110000
		28	000111

27
+ 1
28 = noif

INDICACIONES ENTRE LOS DOMINIOS Y SUS NOTAS EN EL DIRECTORIO



RECUPERACION DE LOS DATOS DEL
TERCER ITEM A PARTIR DEL ARCHIVO COMPRIMIDO

ARCH.COMPRIMIDO	INTERPRETACION	DSCODE	DSXODE	DS
1	000000			
2	101011	4	1) 3	begone
3	011000	3	(3+0) 2) 5	boris
4	000101	1	3) 2	carlos
5	000000	5	4) 1	miguel
6	000000	2	5) 4	victor
7	111110	7	31)41	albizu
8	010101	10	32)34	cota
9	001100	5	33)35	dubin
10	000011	2	(5+30) 34)36	frias
11	000000	3	35)33	guerra
12	000000	4	(6+30) 36)38	ortiz
13	000001	8	37)31	romero
14	101001	6	38)37	rusent
15	011010	9	39)39	santam
16	000101	11	40)32	solis
17	000000	1	41)40	strass
18	000000			
19	100010			
20	000011			
21	101011	21 16 12 8 7 6		
22	001100		12 + 15 - 1 = 26	
23	110000			
24	000000	+ from - 1		
25	000000			
26	101000		(1+80)	
27	110000	3 2 4 4 4	2 81) 84	anal's
28	000111		3 82) 81	operad
			4 83) 82	otro
			5 84) 83	progra

El tercer item es:

boris,dubin,rusenthall,26,programador

CAPITULO IV

CONSULTA.

CONSULTA

Una vez que el banco de datos ha sido construido por medio de los procesos anteriores, el usuario puede inquirir al Sistema sobre la selección de la información a ser extraída.

El usuario puede hacer preguntas, o sea, un programa consistente de postulados de LINGUA con expresiones booleanas con el objeto de recuperar o seleccionar subconjuntos del banco de datos. Estas expresiones están formadas por palabras escogidas de un vocabulario de control.

Todo vocabulario de control tiene elementos permanentes (o, no, y, desde, a) además de un balance formado por los nombres de descriptores y nombres de estados de descripción únicos para ese banco de datos.

Tipos específicos de inquisiciones:

4.1. CUANTOS

Esta inquisición cuenta el número de elementos de un subconjunto dado.

Su forma general es:

100110

CUANTOS ruido expresión booleana *

BT111110

El Sistema al recibir el comando QUINCE crea una cuerda llamada "string" en la cual está la decodificación de la expresión booleana.

OPERADOR	CODIGO
(-1
)	-2
o	-3
y	-4
no	-5

Esta decodificación se hace con relación al archivo comprimido, indicando cómo deben de estar los bits. Esto es, si el nombre Juan tiene en "decode" un 3, entonces en el archivo comprimido, en el lugar correspondiente a éste descriptor, tendremos un tres, en sentido vertical, lo cual nos determina algunos bits prendidos y otros apagados, si la condición que deseamos se cumpla es la que tienen nombre Juan, en "string" tendremos qué bits deben de estar prendidos y cuales apagados.

Supongamos que nuestra base de datos es la creada por el ejemplo 1 y consideramos la siguiente pregunta:

QUINCE: "¿QUE nombre, carlos o apellido pat, dubin *"

Carlos en "decode" tiene el número 1, utilizando el área

reservada para el descriptor nombre, éste valor requiere el bit 2 prendido y de los bits 3 a 6 apagados.

Dubin tiene en "dscodex" el número 5, considerando ahora el área reservada para el dominio apellidoat, el número 5 se representa con el bit 7 prendido, el 8 apagado, el 9 prendido y del 10 al 12 apagados.

Entonces la expresión booleana con respecto a los bits es:

$((2 \text{ y } no3 \text{ y } no4 \text{ y } no5 \text{ y } no6) \text{ o } (7 \text{ y } no8 \text{ y } 9 \text{ y } no10 \text{ y } no11 \text{ y } no12))$

nombre,car'os

apellidoat,dubin

Decodificada: (teniendo presente los códigos para los símbolos de paréntesis y operadores lógicos)

-1-1 2-4-5 3-4-5 4-4-5 5-4-5 6-2-3 7-4-5 8-4 9-4-5 10-4-5 11-4-5 12-2-2....(1)

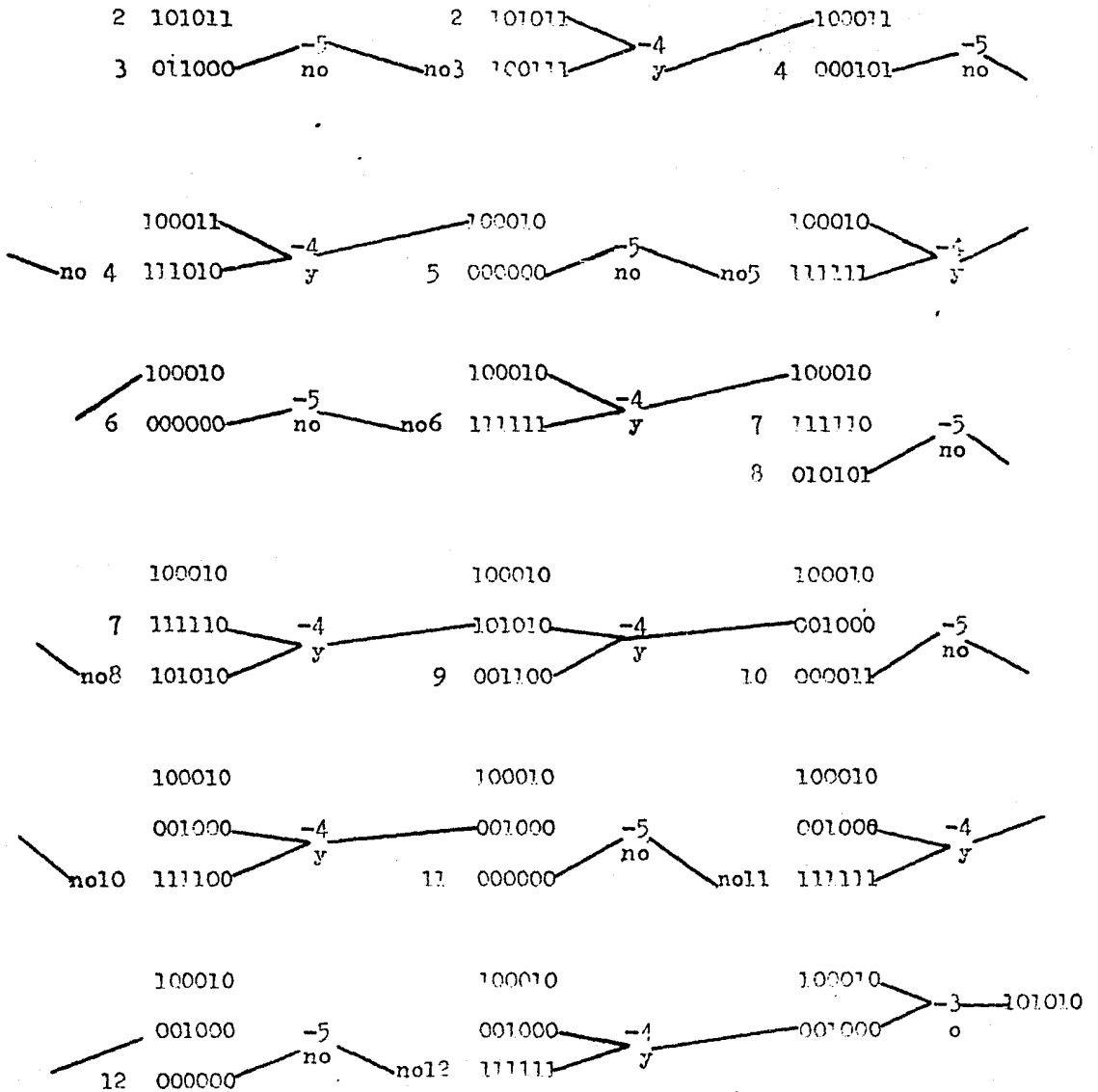
Posteriormente el Sistema transforma ésta expresión en notación polaca.

La expresión (1) en notación polaca es:

2 3-5-4 4-5-4 5-5-4 6-5-4 7 8 -5-4 9-4 10-5-4 11-5-4 12-5-4-3....(2)

Una vez de que la expresión booleana ha sido decodificada y transformada en notación polaca el Sistema utiliza ésta última cuerda resultante para evaluarla, utilizando un stack, y así obtener la respuesta a la pregunta hecha.

Evaluación de la expresión (2) utilizando un stack:



Por lo que el resultado final de la evaluación de la expresión (2) es 101010, ésta combinación de 1 y 0 el Sistema la almacena en la primera localidad del archivo comprimido, los tres bits prendidos indican que hay 3 items que cumplen con la condición.

El lugar que ocupan los bits prendidos indican el item que cumple la condición, siguiendo el orden de declaración (de entrada), en el ejemplo son: el primero, el tercero y el quinto.

El conteo de los items que cumplen la expresión booleana se lleva a cabo por medio de "icount".

Cuando en la expresión booleana se hace mención de algún rango numérico entre el cual estarán los items deseados la cadena booleana creada por el Sistema resulta muy grande, ya que en ella deben quedar especificadas todas las posibles combinaciones de los bits para representar los números que se encuentran en ése rango.

Consideremos la siguiente consulta referente a la base de datos del ejemplo 1:

CUANTOS TIENEN nombre,carlos o apellidoopat,guerra y edad,de 15 a 36 *

La expresión booleana produce la siguiente cuerda:

((2 y 3 y no4 y no5 y no6)) o ((7 y no8 y no9 y no10 y no11 y no 12))

_____ nombre,carlos _____ apellidoopat,guerra _____

y ((no25 y (no24 y ((no23 y (22 o (no22 y (21 o (no21 y (20 o (no20 y 19))))))) o (23 y no22 y (no21 o (21 y (no20 o (20 y no19))))))))))

_____ edad,de 15 a 36 _____

la cual decodificada y en notación polaca es:

2 3 -4 4 -5 -4 5 -5 -4 6 -5 -4 7 8 -5 -4 9 -5 -4 10 -5 -4 11 -5 -4 12
-5 -4 13 -5 14 -5 15 16 -5 17 18 -5 19 20 -5 21 22 -5 23 24 -5 25 26 -5 27 28 -5 29 30 -5 31 32 -4 -3 -4 -3 -4
-3 -4 23 22 -5 -4 21 -5 21 20 -5 20 19 -5 -4 -3 -4 -3 -4 -3 -4 -4 -4 -3

Claramente se observa en éste ejemplo el incremento tan grande en la longitud de la cadena cuando uno de los dominios que constituyen la expresión booleana es numérico, nótese que en éste ejemplo el rango es bastante chico ($36-15=21$) y sin embargo la cuerda creció mucho.

4.2. LISTA

Este tipo de postulado es uno de los mas usados en el lenguaje del Sistema ENVIR, tiene la forma:

bCOMb

LISTA ruido: lista de descriptors bPARAb ruido expresión booleana

bTIENEMB

La respuesta a esta pregunta es del mismo tipo que para la pregunta CUANTOS. A esta pregunta le siguen los estados de cualquier descriptor seleccionado por el usuario (en la lista de descriptors) para cada item en el conjunto formado por la expresión booleana.

La lista de descriptors consiste de una sucesión de nombres de descriptors escogidos del vocabulario de control separados por medio de comas. Esta lista puede tener desde uno hasta tantos descriptors como existen en el banco de datos.

El Sistema imprimirá el estado de descripción apropiado, a cada descriptor de la lista, para cada uno de los items del subconjunto especificado por la expresión booleana.

El Sistema crea una tabla llamada "dlist" en donde se guardan dos informaciones referentes a los descriptors mencionados en la lista del postulado:

dlist(dlx,1) = depara(descriptor dlx)

dlist(dlx,2) = L = I- nostrg(descriptor dlx)

L es inicializada en 47= LTRF

Consideremos la Base de Datos creada por el ejemplo 1 y supongamos la siguiente pregunta:

LISTA: nombre,apellidopat,apellidomat PARA CON nombre,carlos*

entonces la tabla "dlist" que se crea es:

		D'IST			
		dpern	1		
		(1)	(2)		
nombre	1	5	42	=	47-5
apellidopat	2	2	36	=	42-6
apellidomat	3	1	30	=	36-5

Se crea la cuerda con la condición de la expresión booleana de igual forma que sucede con el postulado CUANTOS y posteriormente es transformada a notación polaca.

Esta cuerda es manejada también con un stack para su evaluación. El resultado de ésta evaluación indicará, como ya se dijo, qué items cumplen con la condición determinada por la expresión booleana.

Con éste el Sistema sabe qué items cumplen con la condición y con "dlist" sabe qué descriptores le interesan al usuario para ser imprimidos, conoce la localidad en el registro de descriptores de lo de interés y en éste registro está la información para el archivo com

primido. Sobre el archivo comprimido hace operaciones como "r" y "o", de tal forma que obtiene la información final que son apuntes al registro de los datos para así recuperar el subconjunto formado por la expresión booleana e imprimirlo.

4.3. ORDENA

La palabra ORDENA se puede utilizar con el postulado LISTA.

La forma de usarla es:

CON

ORDENA Y LISTA ruido: lista de descriptores PARA ruido expresión booleana*

TIENE

En el caso de que la lista de descriptores tenga un solo miembro la impresión consistirá de una lista ordenada de estados para ese descriptor omitiendo los estados duplicados.

En una lista de descriptores con más de un elemento la impresión sería ordenada jerárquicamente e indentada.

CAPITULO V

ELIMINA

ELIMINA

bCOMb

ELIMINA REGISTROS ruido

expresión booleana*

bTITENb

En este caso todo ítem del conjunto especificado por la expresión booleana es eliminado del banco de datos. Es una instrucción muy poderosa y debe ser usada con sumo cuidado. Es muy fácil borrar todo el banco de datos con una sola instrucción. Este tipo de instrucción es muy útil, especialmente para bancos de datos con gran cantidad superflua de información.

A este tipo de instrucción el Sistema responde con tres renglones de impresión dando el número de ítems anterior, el número de ítems eliminado y el número de ítems actual.

Con ésta instrucción, al igual que en todos los casos anteriores, se va a crear la cuerda "string" con la decodificación de la expresión booleana, la cual es evaluada para así obtenerse el sub conjunto de ítems de la base de datos deseado. Dichos ítems, como la instrucción lo dice, van a ser eliminados o excluidos de la base de datos.

Aunque ésta eliminación es lógica pero no física.

Los estados pertenecientes a los ítems que se borraron permanecen almacenados en el vocabulario de control, pero al usuario no puede accederlos (para el usuario ya no existen) a menos de que los

vuelva a declarar.

Con la instrucción "DELETE" solamente es modificado el archivo comprimido.

Después de que "string" fué evaluada el Sistema conoce cuáles son los ítems que cumplen con la expresión booleana y por lo tanto los registros a ser eliminados.

A todos los ítems según en el orden en que fueron declarados se les asignó un lugar específico en el archivo comprimido.

Cuando el Sistema debe eliminar algunos ítems lo que hace es recorrer los otros registros, de tal modo que el lugar asignado a los ítems que deben de ser eliminados va a ser ocupado por los registros restantes y así no deja lugares vacíos.

Observemos por medio de un ejemplo las modificaciones que recibe el archivo comprimido después de una instrucción de elimina.

Consideremos la base de datos creada con el ejemplo 1 y supongamos la siguiente pregunta:

ELIMINAR PERSONAS CON nombre,victor y edad,30 o especialidad,otro*

La expresión booleana decodificada y en notación polaca ("string") es:

2 -5 3 -4 4 -5 -4 5 -5 -4 6 -5 -4 19 -5 -4 20 -5 -4 21 -5 -4 22 -5
-4 23 -4 24 -5 -4 25 -5 -4 26 -5 27 -5 -4 28 -4 -3

Evaluando éste cuerno obtenemos como resultado final 010111, o sea, los ítems: segundo, cuarto, quinto y sexto hacen verdadera la

expresión:

victor, guerra, ortiz, 30, no lista*

begona, albisuri, romero, 21, otros*

carlos, santamaría, ortiz, 21, otros*

miguel, cota, colis, 20, otros*

Por lo que el archivo comprimido sólo va a contener información de dos items, mientras que los otros registros permanecen sin alteraciones.

ARCHIVO COMPILADO

ANTES DE LA INSTRUCCION

DESPUES DE LA INSTRUCCION

000000
101011
011000
000101 1,2,3,4,1,5
000000
000000
111110
010101
001100
000011 1,3,5,7,9,10
000000
000000
000001
101001
011010
000101 2,4,6,8,4,11
000000
000000
100010
000011
101011
001100 21,16,12,8,7,6
100000
000000
000000
100000
100000 3,2,1,4,4,4
010111

010111
110000
010000
000000 1,3
000000
000000
110000
000000
010000
000000 1,5
000000
000000
000000
110000
010000
000000 2,6
000000
000000
100000
000000
110000
010000 21,12
100000
000000
000000
110000
100000 3,1
000000

CAPITULO VI

CORRECCION

CORRECCION

Esta instrucción proporciona al usuario una poderosa herramienta para actualizar información obsoleta de su banco de datos.

El formato general tiene la forma:

bCORb

CORRECCION (d,ds) (d,ds)...(d,ds) ruido expresión booleana

bTTTTTb

En este caso d es el nombre de un descriptor escogido del vocabulario de control y ds es el nombre de un estado de descripción para ese descriptor.

Si d fue definido por medio de la opción AIFA, entonces ds puede escogerse del vocabulario de control, o ser el estado DESCONOCIDO o ser un nuevo nombre para ser agregado al vocabulario de control, durante el proceso de corrección. Si d fué definido por medio de la opción CODIGO, entonces ds debe escogerse del vocabulario de control o ser el estado DESCONOCIDO.

Todo ítem del conjunto especificado por la expresión booleana es reasignado al estado de descripción ds para el descriptor d apareado con él. Esto se realiza para todas las parejas d,ds incluidas en la instrucción, las cuales pueden ser desde una hasta el número total de descriptores en el banco de datos.

Cuando el comando CORRECCION es recibido por el Sistema, éste analiza que los descriptores d hayan sido declarados anteriormente por el usuario y obtiene la localidad que ocupan en el regis-

tro de descriptores.

Busca los estados ds en el registro de estados de descriptor, si éstos no han sido declarados los añade en ésta lista (vocabulario de control) en orden alfabético, haciendo corrimientos de esta dos si se requiere.

El Sistema utiliza en ésta instrucción el arreglo "dist", en el cual almacena la localidad del descriptor en "dd" y el número de estados en el vocabulario de control referentes al descriptor en cuestión.

Al igual que en las instrucciones de consulta anteriormente vistas, ENVIR crea la cuerda "string" en donde se va a encontrar la decodificación de la expresión booleana, dicha decodificación se lleva a cabo igual que en los casos anteriores.

Es evaluada ésta expresión, también, utilizando un stack, al terminar la evaluación de string el Sistema conoce cuántos y cuáles son los items que van a ser corregidos. Entonces, los items que han pasado a formar parte del subconjunto creado por la expresión booleana van a ser modificados, para ésto el archivo comprimido será alterado, ya que se pondrá el valor de "dscode" del nuevo o ya existente estado ds en el correspondiente lugar de dichos items.

Observemos por medio de un ejemplo las modificaciones de los archivos con la instrucción CORRECCION.

Supongamos la base creada en el ejemplo 1 y la instrucción:

CORRECCION (nombre,juan) (apellido,mat,ruiz) con especialidad,programador o
apellido,mat,cota*

La tabla "dlist" que se forma es:

		DLIST		
		1	2	
nombre	1 5	5 +1	DSCODE	para Juan
apellidom+t	2 2	11 +1	DSCODE	para Ruiz

La cuerda "string" con la decodificación de la expresión en notación posca es:

26 27 -5 -4 28 -5 -4 7 -5 8 -4 9 -5 -4 10 -4 11 -5 -4 12 -5 -4 -3

El resultado de la evaluación de ésta es 001001, lo cual implica que el tercer y sexto ítem que se declararon forman parte del subconjunto que satisface la expresión booleana, éstos son:

boris,dubin,rusenthall,26,programador

miguel,cota,solis,20,otro

Una vez hecha la corrección resultan alterados el registro de estados de descriptors y el que contiene la información referente a los descriptors. También recibe modificaciones el archivo comprimido.

ANÁLISIS COMPARATIVO DE LOS NOMBRES PROPIOS

- Registro de estados de descriptor

antes de la instrucción

DGS	DSOVAR	DSHOT	DETODE	DSOONE
1 begona	0	0	3	4
2 boris	0	0	5	3
3 carlos	0	0	2	1
4 miguel	0	0	1	5
5 victor	0	0	4	2
.				
.				
.				
31 albizu	9	0	41	7
32 cota	0	0	34	10
33 dubin	0	0	35	5
34 frias	0	0	36	2
35 guerra	0	0	33	3
36 ortiz	0	0	38	4
37 romero	0	0	31	8
38 rusent	8	0	37	6
39 santam	10	0	39	9
40 solis	0	0	32	11
41 stress	7	0	40	1
.				
.				
.				
81 anclia	5	0	84	2
82 operad	6	0	81	3
83 otro	0	0	82	4
84 progr	4	0	83	1

después de la instrucción

DSS	DSOVAL	DSHOW	DSNODE	DSCODE	
1	begona	0	0	3	4
2	boris	0	0	6	3
3	carlos	0	0	2	1
4	juan	0	0	1	6
5	miguel	0	0	5	5
6	victor	0	0	4	2
.					
.					
31	albizu	9	0	42	7
32	cota	0	0	34	10
33	dubin	0	0	35	5
34	frias	0	0	36	2
35	guerra	0	0	33	3
36	ortiz	0	0	39	4
37	romero	0	0	31	8
38	ruiz	0	0	37	12
39	ruvent	8	0	40	6
40	santam	10	0	32	9
41	solis	0	0	41	11
42	strass	7	0	38	1
.					
.					
81	analis	5	0	84	2
82	operad	6	0	81	3
83	otro	0	0	82	4
84	progra	4	0	83	1

Las modificaciones que ésta instrucción originó en el registro de estados de descriptores son:

a) Puesto que "juan", que es el estado del descriptor nombre que se quiere corregir, no se encontraba en el vocabulario de control se debe añadir. La organización de éste registro es alfabética y secuencial, por lo que juan debe ser almacenado después de carlos, lo que implica el corrimiento de los dos últimos registros del descriptor nombre.

Para el almacenamiento de "ruiz" también se requiere del moviniento de algunos registros, en éste caso se mueven 4 registros.

b) Algunos valores de "dexode" fueron modificados.

Como anteriormente se dijo, dexode(i) indica la localidad en la cual se encuentra el i-ésimo ítem declarado, puesto que los registros sufrieron modificaciones en cuanto al lugar que ocupan, es obvio que éste valor también varíe ya que se refiere directamente a las localidades.

- Registro de los descriptores

antes de la instrucción

DD	NOBERG	IFSA	DSSA	DSFA	LENGTH	RESERV	OPTION	HOW	OVADD
apelli	6	13	31	42	12	50	1	0	2
apelli	6	7	31	42	12	50	1	0	1
edad	7	19	0	1	3	0	3	0	0
especi	3	26	81	85	11	0	3	0	3
nombre	5	2	1	6	6	30	1	0	0

después de la instrucción

DD	NOBERG	IFSA	DSSA	DSFA	LENGTH	RESERV	OPTION	HOW	OVADD
apelli	6	13	31	43	12	50	1	0	2
apelli	6	7	31	43	12	50	1	0	1
edad	7	19	0	1	3	0	3	0	0
especi	3	26	81	85	11	0	3	0	3
nombre	5	2	1	7	6	30	1	0	0

Es este como la única modificación fué hecha en el valor de "dsea", el cual fué incrementado en uno para los descriptores apellido pat, apellido mat y nombre.

Se recordará que "dsea" indica la localidad del último estado del descriptor, de tal forma que al restarle a "dsea" el valor de "dssa" se obtiene el número de estados diferentes en el vocabulario de control.

Resulta normal que "dsea" de nombre haya sido incrementado ya que se añadió el estado juv, al igual que también resulta normal el incremento de "dsea" de apellido mat por la introducción del estado ruiz.

Pero puede resultar no muy claro el hecho de que hay sido incrementado el valor de apellido pat ya que a éste descriptor no se le añadió ningún estado, sin embargo éste incremento es perfectamente correcto.

Al crearse la base (ejemplo 1) apellido mat se definió idén tico al segundo descriptor el cual es apellido pat, esto significa, como ya se mencionó, no sólo que ambos descriptores tienen la misma opción y el mismo número de estados reservados sino que además comparten ambos descriptores el área de almacenamiento en el registro de estados, por ésto es por lo que es perfecto y correcto el incremento de "dsea" para apellido pat.

El archivo comprimido también fué alterado, se modificaron los lugares correspondientes a los items que satisfacen la expresión booleana pero sólo la parte correspondiente al descriptor

nombre y el descriptor apellidoat.

Los números que se encuentran en el archivo comprimido son los valores de "dscode" o el valor numérico menos la base y aunque éstos valores no fueron modificados, a los nuevos estados les fueron asignados unos valores no existentes (se recordará que "dscode" indica el orden de entrada de los estados) y puesto que a algunos items (el tercero y el sexto) se les desea corregir la información de nombre y apellidoat por los nuevos estados, entonces es claro que el valor anterior de "dscode" de éstos items debe de ser modificado.

nombre y el description apellidoomat.

Los números que se encuentran en el archivo comprimido son los valores de "dscode" o el valor numérico menos la base y aunque éstos valores no fueron modificados, a los nuevos estados les fueron asignados unos valores no existentes (se recordará que "dscode" indica el orden de entrada de los estados) y puesto que a algunos items (el tercero y el sexto) se les desea corregir la información de nombre y apellidoomat por los nuevos estados, entonces es claro que el valor anterior de "dscode" de éstos items debe de ser modificado.

- Archivo comprimido

antes de la instrucción

después de la instrucción

000000 _____
101011
011000
000101 1,2,3,4,1,5
000000
000000 _____
111110
010101
001100 1,3,5,7,9,10
000011
000000
000000 _____
000001
101001
011010 2,4,6,8,4,11
000101
000000
000000 _____
100010
000011
101011
001100 21,16,12,8,7,6
110000
000000
000000 _____
101000
110000 3,2,1,4,4,4
000111

001001 _____
100010
011001
001101 1,2,6,4,1,6
000000
000000 _____
111110
010101
001100 1,3,5,7,9,10
000011
000000
000000 _____
000000
100000
011011 2,4,12,8,4,12
001101
000000
000000 _____
100010
000011
101011
001100 21,16,12,8,7,6
110000
000000
000000 _____
101000
110000 3,2,1,1,1,4
000111

Obsérvese que los estados boris, miguel, rusehall y solis ya no pertenecen a ningún ítem y sin embargo no fueron borrados del registro de estados.

En el Apéndice A se muestra el Manual de ENVIR en español para el usuario y en el Apéndice B un ejemplo creado interactivamente para ilustrar el manejo.

CAPITULO VII

RESUMEN

RESUMEN

El objetivo de esta tesis fue hacer la documentación técnica de IVIR y el acoplamiento del Sistema para ser utilizado en español, procediendo a investigar su programa fuente (el cual se encuentra tanto en el C.S.C. como en el I.I.M.A.S.). De esta investigación se señalan las siguientes conclusiones.

Las aplicaciones programadas con archivos de datos individuales se están viendo desplazadas por Sistemas Administrativos de Bases de Datos (SABD) porque coordinan archivos integrados con precisión y utilidad en tiempo; y permiten utilizar fácil y económicamente los datos para satisfacer los requerimientos de información que demandan problemas de toma de decisiones. Además la tendencia de los SABD es que los datos sean utilizados por personas que desconozcan completamente programación de sistemas.

Los SABD que tratan de cumplir totalmente con estas características son los pertenecientes al modelo relacional, y en la Universidad Nacional Autónoma de México se tiene el SISTEMA IVIR que puede crear, actualizar, corregir y recuperar datos en computadora sin necesidad de conocimientos de programación, sin embargo este sistema era una "caja negra" y no se tenía la confianza de utilizarlo porque no se sabía como funcionaba y con los primeros problemas que enfrentaban los usuarios lo desechaban.

Aunque no existen especificaciones definitivas para el modelo relacional y para SABD, una especificación muy importante es que

el sublenguaje para operar la Base de Datos puede ser convertido a cualquier lenguaje natural (español, alemán, ruso, francés, chino, etc) mediante construcción de tablas simples.

El reconocedor sintáctico de ENVIR permite esto, sin embargo deja algunas comparaciones (en el flujo de la programación) de cuerdas en inglés, esto deberá ser incluido en tablas para evitar conversiones costosas a otros lenguajes naturales.

Por otra parte, los mensajes que envía el sistema eran codificados y mas bien deben ser autoexplicativos.

ENVIR fue convertido para operar totalmente en español y ahora fácilmente se puede convertir a cualquier otro lenguaje natural.

El reconocedor sintáctico no es de los mas eficientes, pero es simple; y se está buscando mejorarlo utilizando expresiones de apuntes y rastreo de máscaras.

El sistema usa parámetros fijos para alojar a la Base de Datos, esto ocasiona desperdicio de espacio o el sistema pide que se amplien los parámetros; si al comando "Selecciona Dominios n" se le agrega "con registros r", se pueden asignar dinámicamente los parámetros y limitarse a un uso óptimo de disco y memoria según la estructura de los datos.

La reservación de áreas para los dominios al estar analizando los dominios es la óptima porque ocupa el mínimo espacio, pero se

critican los procedimientos de información para ordenar la información para la búsqueda binaria, dichos procedimientos de información consumen mucho tiempo; por esto se propone utilizar listas ligadas auxiliadas con directorios y técnicas de dispersión (aunque esto ocupa aproximadamente 25% del espacio que la organización secuencial) en dominios de tipo alfanumérico y textos, las tablas deben permanecer con la misma tecnología.

En esta tesis se concibe como compactación de información evitar la redundancia de los estados de los descriptores y al uso de áreas de sobreflujo, y como compresión de información representar el código del descriptor dentro de la mínima configuración binaria para representar su universo.

El sistema ENVIR utiliza fronteras de palabras, pero se puede mejorar utilizando caracteres o bits.

El sistema carece de medios para expandir la Base de Datos por lo que necesita un conversor, que también podría servir para formar sub-bases u otras relaciones; aunque puede sacar solamente los códigos de los descriptores a un archivo para ser procesados por otros programas. El conversor se sugiere iniciarlo vaciando a dicho archivo también los estados de los descriptores para retroalimentar la nueva base de datos reducida o expandida.

El uso de expresiones lógicas para selección, corrección y eliminación de información es muy necesario, pero la comparación de

máscaras resultantes de la evaluación de la expresión contra el barrido del archivo se puede evitar utilizando ciertos apuntadores, cuando menos marcar donde se encuentran los primeros y últimos de un mismo estado de descriptor, y por la frecuencia de unas consultas se recomienda cuantificar cada estado de descriptor.

Las limitaciones de RIVIR para un modelo relacional es que su sublenguaje carece de operadores aritméticos, no genera variables y le faltan facilidades de impresión, por lo que se necesita también adaptarle un generador de reportes potente.

Actualmente solo se trabaja una relación con varios dominios pero la estructura de RIVIR aplicándose recursivamente manejará varias relaciones.

APENDICE A

MANUAL DE ENVIR EN ESPAÑOL
PARA EL USUARIO.

MANUAL DE USUARIO PARA EL SISTEMA

El formato de los comandos está ilustrado con una breve explicación de sus funciones. Las palabras escritas en mayúsculas son las requeridas por el sistema. No hay restricción alguna en cuanto a las columnas (formato de entrada); en este sentido, podemos decir que se puede usar formato libre.

Algunos de los comandos deben terminar con asterisco y otros usan separadores.

La coma generalmente es usada como separador, cuando se utiliza el comando LITERAL el usuario puede escoger otros símbolos como separadores.

Las palabras PARA, CON, TIENE y los dos puntos significan el fin de ciertos componentes operativos en algunos comandos, como se muestra enseguida.

Marcas el final del "ruido" y el principio de la lista de descriptores.	Indica el fin de la lista de descriptores.	Fin del "ruido" y principio de la expresión booleana.	Indica el fin del comando.
---	--	---	----------------------------

CON

TIENE ruido: lista de descriptores PARA ruido expresión booleana *

TIENE

MANUAL DE USUARIO PARA EL USUARIO

El formato de los comandos está ilustrado con una breve explicación de sus funciones. Las palabras escritas en mayúsculas son las requeridas por el sistema. No hay restricción alguna en cuanto a las columnas (formato de entrada); en este sentido, podemos decir que se puede usar formato libre.

Algunos de los comandos deben terminar con asterisco y otros usan separadores.

La coma generalmente es usada como separador, cuando se utiliza el comando LITERAL el usuario puede escoger otros símbolos como separadores.

Las palabras PARA, CON, TIENE y los dos puntos significan el fin de ciertos componentes operativos en algunos comandos, como se muestra enseguida.

Marcan el final del "ruido" y el principio de la lista de descriptores.	Indica el fin de la lista de descriptores.	Fin del "ruido" y principio de la expresión booleana.	Indica el fin del comando.
---	--	---	----------------------------

CON

TIENE ruido: lista de descriptores PARA ruido expresión booleana *

TIENE

En el "ruido" el usuario puede poner cualquier cuerda de caracteres (excepto asteriscos) tan larga como le quiere, esto puede ayudarle a hacer la instrucción más inteligible. El "ruido" es ignorado por el sistema.

ANJO

Cancela el comando VERIB para poner el título al final de cada página (ver TITULO).

AGREGA DOMINIOS n nombre de descriptor (especificación).....*

Este comando es utilizado para añadir descriptores aún no especificados. Su forma es idéntica a la del comando SELECCION DOMINIOS.

El parámetro n no puede ser menor que el número total de descriptores definidos en el banco.

Los nombres y los números de secuencia definidos en las especificaciones de los descriptores no pueden ser duplicados de descriptores anteriores.

Todos los registros que fueron almacenados en la base antes de esta instrucción tendrán el estado DESCONOCIDO en estos descriptores.

TARJS

AGREGA REGISTROS DE CINTA

DISCO

ENVIR lee datos como imágenes de tarjetas del medio de entrada que se especifique, construye el archivo comprimido y lista sólo las imágenes de tarjetas que contengan algún error reconocible.

TARJE

ASEGURA Y LISTA REGISTROS DE CINEA

DISCO

Lee los datos como imágenes de tarjetas del medio que se especifique, construye el archivo comprimido e imprime la lista de todos los registros.

Si TIVIN detecta un error en algún registro, el sistema envía un mensaje y este registro no es introducido en el archivo comprimido.

ALTO

Cancela el comando USE OCHANDOS DE CINEA.

ARRIBA

Prende una bandera tal que la impresión del TIVINO estará al principio de una nueva página en la impresión.

COMIGO

Cuando se usa este comando los estados de todos los descriptores COMIGO y LISTA, impresos por ENVIR como respuesta a los comandos LISTA y ORDENA Y LISTA, serán representados por su código numérico en lugar de su nombre alfabético.

COMA

Cancela el comando LITERRA.

COM

CORRECCION (d,ds) (d,ds)...(d,ds) ruido expresión booleana *

EDITAR

Este comando le permite al usuario acudir a cualquier registro en el archivo comprimido y alterar su contenido.

En cada paréntesis "d" representa el nombre del descriptor y "ds" es el estado del descriptor que debe de ser substituido por el estado existente en todos los registros del archivo comprimido que satisfacen la expresión booleana. Esta instrucción debe de

ser usada con mucho cuidado, ya que un error de lógica en la expresión booleana puede destruir la base de datos.

CON

CUANTOS ruido expresión booleana *

TIEMPO

Como resultado da el número de registros en el banco que satisfacen la expresión booleana.

DECIMALES LIBRE

Cuando es usado esta comando ENVIER acepta en los registros de entrada números reales con cualquier número de decimales (máximo 9) y normaliza cualquier entrada al número de lugares especificado en el comando SELECCIONA DOMINIOS, redondeando el número o aumentando tantos ceros como sea necesario.

DIPIURA

Para uno de los administradores del sistema.

CON

ELIMINA REGISTROS ruido expresión booleana *
FINIR

Todos los registros en el archivo comprimido que satisfacen la expresión booleana son borrados de la base.

FINIR modifica el archivo comprimido para ocupar el espacio dejado por los registros borrados.

Con esta instrucción se puede destruir la base al poner una expresión booleana incorrecta.

ENVIA A LA SALIDA

CON

ruido:lista de descriptores PARA ruido exp.bo

ORDENA Y ENVIA A LA SALIDA

FINIR

Este comando genera un subconjunto de datos, en la forma de registros binarios legibles, a varios módulos externos asociados con FINIR.

El subconjunto de datos generado consiste de todos los registros en el banco que satisfacen la expresión booleana.

La lista de descriptores y la expresión booleana están gobernados por las mismas reglas explicadas en el comando LISTA, excepto que cualquier par de paréntesis en la lista de descriptores será

ignorado.

ENVIA A LA SALIDA escribe los registros seleccionados en el medio de salida en el orden en el cual ocurren en el archivo comprimido.

ORDENA Y ENVIA A LA SALIDA los ordena jerárquicamente de acuerdo a la lista de descriptores.

Los estados de descriptores ALFA son ordenados alfabéticamente, mientras que los de DESDE-A y CODIGO son ordenados en forma ascendente.

ESCRIBIR BANCO

La última versión del banco es actualizada según los últimos cambios hechos con AGREGA, CORRIGE y ELIMINA, si no se usa ESCRIBIR BANCO cualquier incidencia queda sin efecto.

ESTRUCTURA DE LA FUNCIÓN

Imprimo el nombre y las especificaciones de todos los descriptores del banco, originalmente especificados por el usuario en los comandos SELECCIONAR DESCRITORES o AGREGAR DESCRITORES.

En la impresión los descriptores son dispuestos en la se-

cuencia ascendente que corresponde a la declarada. En los descriptores AFA y CODIGO, los estados y los códigos numéricos asignados por ENVIR a cada estado son impresos.

Para los descriptores AFA los estados son listados alfabéticamente y para los descriptores CODIGO los estados son listados en la misma secuencia en que fueron declarados por el usuario en los comandos SINTACCIONA DOMINIOS o AGRICA DOMINIOS.

FIN

Termina la ejecución de ENVIR.

La ocurrencia de este comando, como último en la ejecución de ENVIR, es un requerimiento para una normal terminación del programa.

IDEM=FAISO

Cancela el comando IDEM=VERADERO.

IDEM=VERDADERO

Hace que ENVIR retenga el subconjunto de datos definido en cada pregunta, para referirse a él en la siguiente pregunta. El conjunto de datos retenido es referenciado por la palabra IDEM. Esta palabra puede ser usada para obtener este efecto, sin haber usado antes IDEM=VERDADERO, cuando el archivo comprimido consiste sólo de un buffer.

INTERACTIVO

Este comando, que puede usarse sólo cuando se ejecuta ENVIR por terminal, ocasiona la impresión del siguiente mensaje:

ENVIR ESPERA POR DATOS

cuando espera un "input".

USE BANCO

En cualquier ejecución de ENVIR que haga uso de un banco definido en una ejecución anterior, se requiere este comando una vez, antes de cualquier instrucción que envuelva este banco.

En la ejecución en la cual se define por primera vez el banco, no debe de ser usado este comando, a menos de que haya sido usado antes el comando ESCRIBE BANCO.

LEER COMANDOS DE CINTA

Le permite al usuario que ENVIR ejecute una serie de instrucciones (de ENVIR) almacenadas en cinta.

LISTA

CON

ruido:lista de descriptores PARA ruido expresión booleana*

ORDENA Y LISTA

TIEMPO

Este comando le permite al usuario definir un subconjunto del banco el cual será impreso. Tiene algunas variaciones en el formato de salida, como se verá enseguida. Identifica los registros que satisfacen la expresión e imprime los estados especificados en la lista de descriptores.

Lista de descriptores y reglas de formato

1.- La lista de descriptores es una lista de nombres de des-

criptores separados por coma (o algún otro separador).

2.- Si la lista de descriptores no es encerrada entre paréntesis los estados se imprimen en el orden que se declaró en la lista, cada estado en una línea diferente y cada estado indentado 5 posiciones con respecto al anterior. Estados idénticos de un descriptor en registros consecutivos serán impresos sólo una vez, a menos de que los registros difieran en el estado de un previo descriptor.

3.- Para cualquier grupo de descriptores encerrado entre paréntesis, resultarán impresos sus estados correspondientes en la misma línea, con 5 posiciones de indentación con respecto a la línea anterior.

Si una línea es idéntica a la anterior, se omite la segunda.

Para la impresión de los estados, ENVIR reserva el número de caracteres del estado más largo del descriptor más un blanco. Si la línea para imprimirse requiere de más posiciones que las permitidas, la instrucción no es ejecutada y se manda mensaje de error.

4.- Si una pregunta contiene una lista de descriptores idéntica, en todos los aspectos, a la de previas preguntas es suficiente con poner la palabra MISMO en el lugar de la lista de descriptores.

Expresión booleana

Una expresión booleana es una cuerda de descriptores y estados de descriptor separados por coma, operadores booleanos o pa-

Palabras reservadas y las siguientes reglas:

1.- d, ds

Es una expresión booleana si d es un nombre de descriptor y ds es un estado en el rango de este descriptor. Los rangos de un descriptor DESDE-A y CODIGO se especifican en el comando SUBRANGOS DOMINIOS.

El rango de un descriptor ALTA se crea con el conjunto de estados correspondientes al descriptor.

El rango de todos los descriptores incluye el estado DESCONOCIDO, el cual en la impresión se representa con tres guiones.

2.- IDEM

Es una expresión booleana que representa al subconjunto de finido bajo el comando IDEM=VERDADERO.

3.- $!b$

Es una expresión booleana, si b lo es.

4.- $d, d \supset s_1 \wedge s_2$

Es una expresión booleana si s_1 y s_2 son estados en el rango del descriptor d y si s_1 es más pequeño que s_2 . Esta forma sólo puede ser usada para descriptores DESDE-A y CODIGO, en el último caso s_1 y s_2 son substituidos por su código numérico aunque el usuario es libre para dar los nombres en lugar del código.

5.- $d, s_1 \vee s_2 \dots$

Es una expresión booleana si s_1 y s_2 son cualquier estado en el rango del descriptor d .

6.- $b_1 \vee b_2 \vee b_3 \dots$

Es una expresión booleana si b_1, b_2, b_3, \dots son expresiones booleanas y op_1, op_2, \dots son los operadores "y" o "o".

7.- $(b_1 op_1 b_2) op_2 (b_3 \dots)$

Es una expresión booleana como un corolario del anterior. No es limitado el número de paréntesis para utilizarse en la expresión booleana.

Ordenamiento

El comando ORDENA Y LISTA es idéntico al comando LISTA excepto que la lista que va a imprimirse es ordenada como se explica en ORDENA Y ENVIA A LA SALIDA, mientras que para el comando LISTA son imprimidos siguiendo el orden del archivo comprimido.

LITERAL separador

Permite usar un separador diferente a la coma en los datos y en los comandos. El carácter seleccionado debe seguir a la palabra LITERAL. El nuevo separador no puede ser ninguno de los siguientes caracteres: * > < o blanco.

Cuando se usa el comando LITERAL, las palabras DE, A, ENNE, PARA, NINGUNO, DESCONOCIDO y los operadores booleanos Y, O, NO son reemplazados para usarse en los datos, y solo son reconocidos en su

capacidad operacional si son precedidos inmediatamente por un periodo.

MENSAJE A LA SALIDA

El contenido de la primera tarjeta siguiente a este comando es escrita por ENVIR al principio del subconjunto de datos creado como respuesta a los comandos ENVIA A LA SALIDA y ORDENA Y ENVIA A LA SALIDA.

Esto le permite al usuario pasar información requerida por el módulo invocado. Esta información debe de ser menos o igual a 80 caracteres.

NOTA mensaje del usuario *

El mensaje del usuario puede contener cualquier cuerda de caracteres excepto asteriscos y ocupar cualquier número de líneas.

Un asterisco termine el mensaje.

El mensaje se imprime y continúa la ejecución de las siguientes instrucciones.

REORDENA DOMINIOS

En experimentación.

REORDENA DOMINIOS número de secuencia de descriptores separados por algún separador

Este comando sirve para modificar la correspondencia entre el descriptor en su definición y el archivo de los registros de entrada. Inmediatamente se debe proceder al comando AGREGA REGISTROS o AGREGA Y LISTA REGISTROS, siempre que un nuevo grupo de registros va a ser leído sin seguir la definición de los descriptores hecha en el banco, con los comandos SELECCIONA DOMINIOS o AGREGA REGISTROS.

El comando REORDENA DOMINIOS debe de tener tantos archivos como los nuevos registros de entrada.

La nueva correspondencia está establecida por meter un número de secuencia de descriptor en el apropiado archivo. (Los números de secuencia de los descriptores son imprimidos por ENVIR en respuesta al comando SELECCIONA DOMINIOS DE LA RELACION).

SELECCIONA DOMINIOS n nombre de descriptor (especificación).....*

El parámetro n es el número de archivos en el registro de entrada.

SELECCIONA DOMINIOS es el comando por el cual el usuario hace todas las especificaciones de los descriptores para la construcción del archivo comprimido. Este comando puede ser usado sólo una vez en la vida del banco.

Descriptores adicionales sólo pueden ser especificados por el comando AGRUPA DOMINIOS.

Todo descriptor es declarado dando su nombre y su especificación entre paréntesis.

Las especificaciones son de la forma:

(s tipo l)

Donde s es el número de posición (entero positivo), el cual indica el lugar de este archivo en los registros de datos. El tipo puede ser ALFA, CODIGO o NUMER, identificando el tipo del descriptor.

El parámetro l es:

a) Para los descriptores ALFA el máximo número de estados únicos aceptados por éste descriptor.

b) Para los descriptores CODIGO la lista de todos los nombres en la secuencia debe ser separados por algún separador. NUMER le asocia a cada uno de estos nombres un entero positivo igual a su secuencia en ésta lista, empezando con 1 para el primer nombre.

c) Para los descriptores $W(377)-1$, 1 toma la forma:

$i \dots j$

Donde i y j son los valores mínimo y máximo respectivamente del rango del descriptor, ambos pueden tener cualquier signo pero el positivo es implícito, y en todos los casos j debe de ser mayor que i . Si el número es real todo se debe especificar dentro de los paréntesis, de la siguiente forma:

DECIMAL k

Donde k es el número de decimales esperados en todos los registros. En este caso i y j deben de ser dados con todos sus dígitos sin el punto decimal.

La especificación DECIMAL puede estar seguida por una especificación opcional de UNIDAD la cual también puede ser usada para los descriptores cuyos valores van a ser enteros, sirve para especificar las unidades. Esta UNIDAD es de la forma:

EN a

Donde " a " es una cuerda de caracteres, del tamaño de una palabra de computadora o menor.

d) Cuando dos descriptores tienen especificaciones idénticas, el nombre del segundo descriptor puede estar seguido por una especificación de la forma:

$s=r$

Donde " s " es el número de secuencia del descriptor en cuestión y " r " es el número de secuencia del anterior descriptor que tiene la misma especificación.

El primer descriptor declarado puede, pero no es necesario, empezar en la misma tarjeta que contiene la palabra **SUBFUNCIONALES DOMINIOS**.

Las declaraciones de los demás descriptores pueden hacerse en éstas u otras tarjetas.

Cualquier otro caracter que un blanco o asterisco después de un paréntesis derecho será considerado como parte del nombre del siguiente descriptor.

El comando termina con un asterisco.

No es necesario especificar tantos descriptores como archivos hay en los registros de entrada, pero sólo los archivos cuyos descriptores hayan sido especificados serán almacenados en el archivo comprimido.

TITULO el resto de la tarjeta puede contener cualquier cuerda de caracteres.

La cuerda de caracteres dada por el usuario será imprimida al principio de todas las páginas hasta que sea usado un nuevo comando **TITULO**. Este comando no puede exceder a 80 columnas.

VACIA

Una vez que éste comando ha sido usado cualquier comando **ENVIA A LA SALIDA** u **ORDENA Y ENVIA A LA SALIDA** usado después producirá una copia de los datos sumados al subconjunto del archivo binario.

APENDICE B

EJEMPLO CREADO INTERACTIVAMENTE
PARA ILUSTRAR EL MANEJO DE ENVIR.

FOLIO 052279 HORA 111251



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
SECRETARÍA DE EDUCACIÓN PÚBLICA
DIRECCIÓN GENERAL DE INVESTIGACIONES CIENTÍFICAS

AGOSTO DE 1978

SELECCIÓN DE FOLIOS 7

NÚMERO ALFA 50)

DIRECCIÓN ALFA 50)

SEXO (MASCULINO FEMENINO)

(MASCULINO)

EDADES DESDE 15 A 65)

OCCUPACIONES ALFA 10)

HORAS TRABAJADAS (DESDE 200 A 400 DECIMAL 1 EN HORAS)

SUELDO SEMANAL (DESDE 100000 A 600000 DECIMAL 2 EN PESOS)

AGREGA Y LISTA REGISTROS DE CINTA

RUIZ SANCHEZ JUAN PLUTARCO ELIAS CALLES 25 MASCULINO 22 CENTAURO

35.0/2000.00



GUILFIERREZ JUAREZ MARIE, INSURGENTES 501 1284, 1, 32, SECRETARIA,
20.0, 1000.00*

DIAZ LIZIETA ELIABETH, INSURGENTES 35, FEMENINO, 22, SECRETARIA,
40.0, 2125.00*

CORONA CRUZ MARCOS, ELIABETH, VENTIZ 105, 2, 42, INGENIERO ELECTRICISTA,
32.5, 2500.00*

DIAZ SANCHEZ RUBERT, EL CHURRUSCU 12, MASQUILLO, 30, FUGROGRADISTA,
40.0, 2387.50*

RODRIGUEZ GUTIERREZ ELIABETH, PATRIOTISMO 539, FEMENINO, 25, ABOGADO,
20.0, 2120.25*

GUERRA SANCHEZ OCTAVIO, REFORMA 241, 2, 27, INGENIERO ELECTRICISTA,
40.0, 3000.00*

ZAMUDIO VAZQUEZ PAUL, REFORMA 28, 2, 35, CONTADOR,
35.0, 2500.00*

GUILFIERREZ FERNANDEZ LAILA, BAJA CALIFORNIA 12, 1, 27, SECRETARIA,
40.0, 1900.50*



ESCALANTE PERMANEZ PUBLICA AMOSIL LEGIN 015 00101100 23 PROGRAMARCP

20.00.1375.000

ESCALITE BANCO

FECHA DE ALMACENAMIENTO DE CINTA: 052279 HORA FINES 111251

BANCO DE LOCALIZACION Y VALORES ACTUALES USADOS

	I	F	G	H	I	J	K	L
400	1000	300	5000	400	60	30	1000	400
7	112	3	8A	7	53	38		

NUMERO DE ARCHIVOS = 10
 NUMERO DE BUFFERS UTILIZADOS 1

ESTRUCTURA DE LA RELACION

1. NOMBRE
 - NUMERO DE ESTADOS RESERVADOS PARA LA DESCRIPCION 50
 - NUMERO DE ESTADOS UTILIZADOS PARA LA DESCRIPCION 10
 - OPCION ALFA. NUMERO DE CARACTERES EN EL ESTADO MAYOR 26
2. LINEACION
 - NUMERO DE ESTADOS RESERVADOS PARA LA DESCRIPCION 50
 - NUMERO DE ESTADOS UTILIZADOS PARA LA DESCRIPCION 10
 - OPCION ALFA. NUMERO DE CARACTERES EN EL ESTADO MAYOR 24
3. SEXO
 - OPCION CODIGO. NUMERO DE CARACTERES EN EL ESTADO MAYOR 9
4. LAD
 - DESLE 15 A 85. OPCION ALFA. NUMERO DE CARACTERES EN EL ESTADO MAYOR 3
5. OCUPACION
 - NUMERO DE ESTADOS RESERVADOS PARA LA DESCRIPCION 10
 - NUMERO DE ESTADOS UTILIZADOS PARA LA DESCRIPCION 5
 - OPCION ALFA. NUMERO DE CARACTERES EN EL ESTADO MAYOR 22
6. HORAS TRABAJO
 - DESLE 200 A 400. OPCION ALFA. NUMERO DE CARACTERES EN EL ESTADO MAYOR 10

FECHA: 052279

PAGINA



7. SELLO SERIAL 110000 A 600000 SERIAL 2 LITONETA DESCS
DESDE 1951. NUMERO DE CARACTERES EN EL ESTADO MAYOR 13
OFICIA



CUANTOS TIENEN SUELDO SEMANAL DE 1000.00 A 2000.00 Y ELABORAN 15 A 70*

NO. DE REGISTROS QUE CUMPLE LA CONDICION = 3
 NO. DE REGISTROS EN EL BANCO DE DATOS = 10
 PORCENTAJE AL RESP. TOTAL EN EL BANCO DE DATOS = 30.00

LISTA NOMBRE, OCUPACION, HORAS TRABAJO PARA CEN IDEM*

NO. DE REGISTROS QUE CUMPLE LA CONDICION = 3
 NO. DE REGISTROS EN EL BANCO DE DATOS = 10
 PORCENTAJE AL RESP. TOTAL EN EL BANCO DE DATOS = 30.00

GUILFIERREZ JUAREZ MARIA
 SECRETARIA
 20.0 HORAS
 GUILFIERREZ FERNANDEZ LAURA
 SECRETARIA
 40.0 HORAS
 ESCALANTE FERNANDEZ MOLICA
 PROGRAMADOR
 20.0 HORAS

CORRECCION (SUELDO SEMANAL 6000.00) CON OCUPACION INGENIERO ELECTRICISTA*
 2 REGISTROS FUERON CORREGIDOS COMO SE REQUIRIO

LISTA Y ORDENAR (NOMBRE, OCUPACION, SUELDO SEMANAL) PARA CEN IDEM*

NO. DE REGISTROS QUE CUMPLE LA CONDICION = 2
 NO. DE REGISTROS EN EL BANCO DE DATOS = 10
 PORCENTAJE AL RESP. TOTAL EN EL BANCO DE DATOS = 20.00

GONZALEZ CRUZ MARCOS INGENIERO ELECTRICISTA 6000.00 PESOS
 GUERRA SANCHEZ OCTAVIO INGENIERO ELECTRICISTA 6000.00 PESOS

CUANTOS TIENEN NI LUNES, DESCUONCILL*

NO. DE REGISTROS QUE CUMPLE LA CONDICION = 10



NO. DE REGISTROS EN EL BANCO DE DATOS = 10
 PORCENTAJE DE RESP. TOTAL EN EL BANCO DE DATOS = 100.00

LISTA DE REGISTROS QUE TIENEN EDAD DE 25 A 39 Y SEXO FEMENINO*

NO. ANTERIOR DE REGISTROS EN EL BANCO = 10
 NO. DE REGISTROS ELIMINADOS = 3
 ACTUAL NO. DE REGISTROS EN EL BANCO = 7

LISTA Y ORDENACION DE REGISTROS PARA QUE SE CORRIJA DESCRIPCION*

NO. DE REGISTROS QUE CUMPLE LA CONDICION = 7
 NO. DE REGISTROS EN EL BANCO DE DATOS = 7
 PORCENTAJE DE RESP. TOTAL EN EL BANCO DE DATOS = 100.00

RUIZ SANCHEZ JUAN	MASCULINO	35
ESCALANTE FERNANDEZ ROSA	FEMENINO	23
DIAS LUIS FELIX	FEMENINO	29
GUERRA LUCAS MARCOS	MASCULINO	37
DIAS SANCHEZ FLORENTIN	MASCULINO	35
ZARAGOZA VAZQUEZ PAUL	MASCULINO	35
GUERRA SANCHEZ OCTAVIO	MASCULINO	27

NOTA CUANDO SE ELIMINAN REGISTROS SE DEBE DE DAR EL COMANDO
 "ESCRIBIR BANCO" PARA QUE LA NUEVA VERSION DEL BANCO SEA
 ALMACENADA Y ASI DEJAR LA ANTERIOR*

ESCRIBIR BANCO

FECHA DE ALMACENAMIENTO EN CINTA: HICLBY 052279 HORA: LHMSS 111251

BANCO DE LOCALIZACION Y VALORES ACTUALES USADOS

A	B	C	D	E	F	G	H	I	J	K	L
400	1000	300	500	400	80	30	1000	40	1000	400	400
7	112	3	82	7	93	30					

NO. DE ARCHIVOS = 7
 NUMERO DE BUFFERS COMPRIMIDOS = 1

BIBLIOGRAFIA

BIBLIOGRAFIA

"An Introduction to Database Systems".

C.J.DAT

Addison-Wesley Publishing Company.

2^a Edición, 1977.

"Introducción a la Informática".

José Luis Mora y Enzo Molino.

Editorial Trillas.

2^a Edición, 1975.

"Envir, The Primary Module of the ENVIRAS SYSTEM. User Manual".

Gulf Universities Research Consortium.

1973.

"Exir, User's Manual"

EPO Biology Department. University of Colorado.

Lois Abbott, editor.

Enero, 1975.