

Te
819a
1985

**BIBLIOTECA
DE LA
UNIVERSIDAD DEL VALLE DE GUATEMALA**

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades

AUTOMATIZACION DE LOS PROCESOS DE
NORMALIZACION DE VOCABULARIOS Y TESAUROS

HECTOR ALFREDO ECHEVERRIA AYALA

Guatemala

1,985

AUTOMATIZACION DE LOS PROCESOS DE
NORMALIZACION DE VOCABULARIOS Y TESAUROS

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades

AUTOMATIZACION DE LOS PROCESOS DE
NORMALIZACION DE VOCABULARIOS Y TESAURUS

HECTOR ALFREDO ECHEVERRIA AYALA

Modelo de Trabajo Profesional presentado para optar al
grado académico de Licenciado en Ciencias de la Computación

Guatemala

1,985

Vo. Bo. :

(f) _____
Ingeniero Luis Furlán
Asesor

(f) _____
Licenciada Rocío Marbán

(f) _____
Ingeniero Luis Obiols

Fecha de aprobación: 25 de octubre de 1985.

PREFACIO

El presente trabajo se realizó dentro del ambiente de la División de Documentación e Información del Instituto Centroamericano de Investigación y Tecnología Industrial -ICAITI-, con la finalidad de obtener

Un Sistema para el Manejo de Tesoros.

El sistema para manejo de tesoros es un grupo de programas que permiten crear y estructurar tesoros con la facilidad de la automatización. En la sección del trabajo que se detalla ese sistema se presenta el manual de usuario del sistema, la estructura que tiene un tesoro dentro del computador y la documentación de los programas que componen el sistema.

DEDICATORIA

A DIOS

A MIS PADRES

Jacob Echeverría Barrera
Elida G. Ayala de Echeverría

A MI ABUELITA

Ana Dolores Recinos
vda. de Mantanico

A MIS TIOS Y TIAS

A MIS HERMANOS

Magda, Bessie, Vilia,
Rosa, Griselda, Silvia,
Jacob, Hugo y Federico.

Y A MIS AMIGOS

AGRADECIMIENTO

AL INSTITUTO CENTROAMERICANO DE INVESTIGACION Y
TECNOLOGIA INDUSTRIAL -ICAITI-, por su colaboración en
el desarrollo del presente trabajo.

AL PROGRAMA DE ORIENTACION Y BECAS PARA ESTUDIOS
SUPERIORES -FOES-, DE LA UNIVERSIDAD DEL VALLE DE
GUATEMALA, por ayudarme a financiar mi carrera.

CONTENIDO

	Página
Introducción	1
Antes de cualquier cosa	2
Programas y archivos que forman el sistema	2
Equipo necesario	3
El sistema asume que	3
Tomar en cuenta que	4
Preparando el sistema para trabajar	5
Cómo empezar	6
Procesos que se pueden realizar	6
Estructura de la base de datos de un tesoro	9
Cantidad de términos que se pueden trabajar	15
Instrucciones de trabajo	
Determinar qué tesoros están en el sistema	16
Cargar los programas	16
Códigos de acceso	17
Crear un tesoro	17
Cambiar el tesoro en uso	17
Ingresar términos individuales	17
Ingresar términos agrupados	18
Crear una relación	18
Borrar una relación	18
Borrar un término	18
Modificar un término	19
Agregar códigos de clasificación	19
Eliminar códigos de clasificación	19
Poner una nota de alcance	19
Quitar una nota de alcance	19
Mostrar un término	20
Listar los términos	20
Imprimir la sección alfabética	20
Listar códigos de clasificación y encabezados	20
Ingresar códigos de clasificación y encabezados	20
Modificar códigos de clasificación y encabezados	20
Eliminar códigos de clasificación y encabezados	21
Imprimir la sección clasificada	21
Eliminar un tesoro	21
Verificar número de registros en uso	21
Finalizar	21

Programas:

INSTAL	23
TESAURO	25
TESA	30
TESB	32
TESC	35
TESD	41
TESE	46
TESF	50
TESG	52
TESH	54
TESI	57
TESJ	59
TESK	61
TESK1	63
TESL	65
TESL1	69
TESM	71
TESN	77
TESN1	87
TESN2	89
TESO	91
TESP	96
TESQ	98
TESR	100
TESS	102
TEST	105
TESU	107
TESV	109
TESV1	116
TESV2	118
Rutina LINEAS	121
Rutina LEERFILE	123
Rutina CEROS	125
Rutina CLASIF	127
Rutina ORDEN	130
Rutina ORDEN1	132

Muestras de reportes (formatos de salida)

Listado de términos	75
Sección alfabética del tesaurus	83
Códigos de la sección clasificada	95
Sección clasificada del tesaurus	114

INTRODUCCION

=====

El Sistema para Manejo de Tesauros es un grupo de programas preparados para apoyar el trabajo que se realiza en las unidades que desarrollan actividades de Normalización de Vocabularios y Tesauros.

El sistema no está diseñado para preparar tesauros jerarquizados en todo detalle, pero permite estructurar y darle mantenimiento a un tesauro o simplemente preparar tablas de términos utilizados en determinada rama del conocimiento.

El usuario de este sistema dispondrá de las siguientes facilidades:

- Almacenar términos en orden alfabético.
- Definir términos que van a ser empleados en vez de otros.
- Crear relaciones entre los términos ingresados, con el fin de identificar qué términos son más amplios o más específicos que otros.
- Hacer anotaciones para especificar el contexto dentro del cual será empleado un término (notas de alcance).
- Asignar códigos a los descriptores para ubicarlos bajo determinados encabezados de la sección clasificada del tesauro.
- Consultar al programa para saber cuántos términos se tienen en el tesauro.
- Listar la información existente en los archivos, total o parcialmente.
- Imprimir la sección clasificada del tesauro.
- Imprimir la sección alfabética del tesauro.

Los programas están diseñados para realizar, en forma automática, aquellos procesos que se deriven de una instrucción dada por el usuario, por ejemplo, si el usuario define la relación:

ORDENADOR
Use: COMPUTADOR

el programa crea la relación complementaria, que para este caso sería:

COMPUTADOR
Usado por: ORDENADOR

Estas facilidades permitirán al usuario optimizar el tiempo y realizar en mejor forma su trabajo.

MANUAL DE USUARIO DEL SISTEMA DE MANEJO DE TESAUROS

ANTES DE CUALQUIER COSA

Antes de iniciar el trabajo con este sistema usted se deberá asegurar de que tiene el sistema completo, así como el equipo que este programa necesita.

PROGRAMAS Y ARCHIVOS QUE FORMAN EL SISTEMA:

Programas:

Tienen las instrucciones que servirán para realizar cada una de las tareas necesarias en el manejo de tesauros.

La extensión .CMD (en sistema operativo CP/M) o .PRG (en sistema operativo MS-DOS) significa Archivo de Comandos.

INSTAL.CMD	TESAURO.CMD	TESA.CMD	TESB.CMD
TESC.CMD	TESD.CMD	TESE.CMD	TESF.CMD
TESG.CMD	TESH.CMD	TESI.CMD	TESJ.CMD
TESK.CMD	TESK1.CMD	TESL.CMD	TESL1.CMD
TESM.CMD	TESN.CMD	TESN1.CMD	TESN2.CMD
TESO.CMD	TESP.CMD	TESQ.CMD	TESR.CMD
TESS.CMD	TEST.CMD	TESU.CMD	TESV.CMD
TESV1.CMD	TESV2.CMD	LINEAS.CMD	CLASIF.CMD
CEROS.CMD	ORDEN.CMD	ORDEN1.CMD	LEERFILE.CMD

Archivos de datos:

Tienen la estructura que se toma como base para crear nuevos tesauros, también tienen el tesauro que se toma como inicial al empezar el programa.

La extensión .DBF significa Archivo de Base de Datos.

A.DBF	B.DBF	C.DBF	E.DBF
-------	-------	-------	-------

Archivos de índices:

Estos archivos tienen índices que permiten que la recuperación de la información que está en los archivos de datos sea en orden alfabético. Corresponde un archivo de índices a cada uno de los archivos de datos.

La extensión .NDX significa Archivo de Índices.

A.NDX	B.NDX	C.NDX	D.NDX
-------	-------	-------	-------

Formatos para reportes:

Tienen instrucciones que definen la forma en la que van a ser presentados los reportes en la pantalla o en la impresora. El nombre de los formatos es el mismo que el del programa que los utiliza.

La extensión .FRM significa Archivo de Formato.

TESM.FRM TESO.FRM

Archivos de memoria:

Estos archivos son utilizados para almacenar datos que se usan dentro de los programas y que no corresponden a información propia del tesauero.

La extensión .MEM significa Archivo de Memoria.

TESAURO.MEM TEMP.MEM

EQUIPO NECESARIO:

Para usar el Sistema de Manejo de Tesauros se requiere una microcomputadora con las siguientes características:

- Sistema operativo que permita el uso del DBASE II.
- Memoria de trabajo mínima de 64 Kbytes, recomendable 128.
- Diskettes de 360 Kbytes mínimo, recomendable disco duro.
- Impresor EPSON MX-80, o que reciba las mismas instrucciones para el control del tipo de letra (si no se tiene será necesario modificar los programas TESN.COM y TESV.COM para dar las instrucciones correspondientes al impresor que se tenga).

EL SISTEMA ASUME QUE:

- En el disco B: se tienen los programas para manejo de tesauros. Si se tienen en otro disco será necesario modificar los programas TESAURO (línea 7) e INSTAL (línea 15).
- Se tiene una impresora EPSON MX-80.
- El papel para la impresora tiene 66 líneas contadas de arriba para abajo (tamaño carta), en letra normal.

TOMAR EN CUENTA QUE:

- El nombre de un tesoro puede tener como máximo siete (7) caracteres. Debe principiar con una letra y continuar con letras o dígitos, sin espacios en blanco intermedios.
- Los términos y los encabezados de la sección clasificada pueden tener hasta cuarenta (40) caracteres.
- Los códigos de clasificación pueden tener hasta 4 caracteres.
- Los términos y códigos se pueden ingresar en letra mayúscula o minúscula pero el programa los trabaja y deja grabados en letra mayúscula.
- Los encabezados de la sección clasificada y las Notas de Alcance se graban tal y como sean ingresadas por el usuario.
- Las Notas de Alcance pueden tener un máximo de 120 caracteres separados en cuatro líneas de 30 caracteres cada una.

PREPARANDO EL SISTEMA PARA TRABAJAR

Para preparar el computador y los programas para trabajar es necesario seguir las siguientes instrucciones.

- Si tiene los programas grabados en diskette y va a utilizar disco duro, será necesario copiarlos al disco duro. Si no los tiene en diskette, deberá escribirlos usando un programa EDITOR o PROCESADOR DE PALABRAS o directamente desde dBase II utilizando la instrucción "Modify Command". Asegúrese de copiarlos cuidadosamente para no cometer errores que interfieran con la correcta ejecución.

- El primer programa que necesitará es el `INSTAL.CMD`, este programa sirve para dar un código o clave de uso al sistema. El código identificará a las personas que pueden modificar los datos que se tienen en los tesauros que se trabajan.

Para llamar a este programa se usa la instrucción:

```
A>DBASE B:INSTAL
```

(Suponiendo que el DBASE se encuentra en el disco A:)

El programa le pedirá que ingrese el código, o palabra clave. Este código debe ser una combinación de letras o signos cuales quiera. Esta clave puede tener desde 1 a 256 caracteres, pero se recomienda usar de 6 a 10 caracteres ya que este código deberá ser ingresado 'exactamente igual' en el momento de llamar al programa para manejar tesauros.

- Borrar el programa `INSTAL` del disco (teniendo una copia en diskette), ya que cualquier persona podría cambiar el código del sistema si este programa quedara disponible en el disco.

La instrucción para borrar el programa es:

En sistema operativo CP/M A>ERA B:INSTAL.CMD

En sistema operativo MS-DOS A>ERASE B:INSTAL.PRG

El programa `INSTAL` también puede ser usado para modificar el código asignado al sistema, siguiendo el mismo proceso como si fuera la primera vez que se da la palabra clave.

En este momento su sistema ya está listo para trabajar.

COMO EMPEZAR

La instrucción para llamar al Sistema de Manejo de Tesoros es :

A>DBASE B:TESAURO
(suponiendo que el DBASE está en el disco A:)

El programa preguntará por el nivel de acceso (password), esto es el código del sistema, si la persona que utiliza el programa conoce este código deberá ingresarlo para que el programa sepa que esa persona si podrá modificar la información que está en el tesoro. Si no lo sabe sólo deberá presionar la tecla RETURN.

Es importante que el usuario sepa que cuando ingrese el código del sistema éste no se va a ver en la pantalla, pero el programa si estará recibiendo la información que se escriba.

Después de que el programa lea la respuesta del usuario presentará el menú inicial, en el que se muestran las opciones que se tienen para trabajar.

PROCESOS QUE SE PUEDEN REALIZAR:

En el menú inicial se muestran las tareas que el programa puede realizar, además muestra el nombre del tesoro que se encuentra en uso. Al comienzo del programa se tiene siempre el tesoro EJEMPLO. Si se cambiara de tesoro en uso no se podrá regresar al tesoro EJEMPLO ya que éste se emplea únicamente como estructura base para trabajar otros tesoros.

A continuación se describe la operación a la que se refiere cada una de las opciones que presenta el menú. Entre paréntesis se muestra la letra que se debe presionar para solicitar esa opción.

La letra que se presione puede ser mayúscula o minúscula.

CREAR UN TESAURO (S): Esta opción permite preparar archivos para un nuevo tesoro, desde el tesoro en uso.

BORRAR UN TESAURO (T): Para borrar los archivos que se asignaron a un tesoro. Los archivos no se podrán recuperar después.

CAMBIAR DE TESAURO EN USO (U): Para cerrar los archivos del tesoro que se trabaja y abrir los archivos de otro tesoro para poder revisar y/o modificar la información existente.

INGRESAR TERMINOS INDIVIDUALES (A): Para ingresar al tesoro uno o varios términos en cualquier momento. Incluye el ingreso de sus códigos de clasificación, si se tienen disponibles.

INGRESAR TERMINOS AGRUPADOS (B): Para ingresar términos que estén ubicados bajo el mismo encabezado en la sección clasificada, evitando tener que escribir varias veces un mismo código.

MODIFICAR UN TERMINO (F): Para modificar un término que se hubiera ingresado incorrectamente.

BORRAR UN TERMINO (E): Esta opción permite eliminar de los archivos cualquier término e información relacionada él.

CREAR UNA RELACION (C): Para un término que ya se ingresó se puede definir una relación con otro término, que puede no haber sido ingresado antes. El programa define la relación que resulta complementaria, por ejemplo

Si se crea la relación:

AZUCARES es MAS AMPLIO que LACTOSA

el programa define la relación:

LACTOSA es MAS ESPECIFICO que AZUCARES

por lo que no habría necesidad de definir la relación complementaria, es más, si se define el programa indicará que esa relación ya fue definida.

BORRAR UNA RELACION (D): Para un término al que se le asignó cierta relación con otro término se borra la relación que se solicite y también borra la relación complementaria.

AGREGAR CODIGOS A UN TERMINO (G): Para un término ya existente en el archivo de datos. Se refiere a los códigos de clasificación que se pueden asignar a un descriptor. Esta opción permite adicionar uno o más códigos a la clasificación de un descriptor.

QUITAR UN CODIGO (H): Para quitar uno o más códigos de clasificación a un descriptor.

PONER NOTA DE ALCANCE (I): Para asignar una Nota de Alcance a un descriptor que ya se ingresó al archivo de datos.

QUITAR NOTA DE ALCANCE (J): Borra del archivo la Nota de Alcance asignada a un descriptor.

Si se desea MODIFICAR una nota de alcance se deberá QUITAR la NOTA de ALCANCE y luego PONERle la nota ya modificada.

NUMERO DE REGISTROS EN USO ? (K): Este proceso muestra en la pantalla la cantidad de registros que tiene en uso cada uno de los archivos que forman parte del tesoro que se trabaja. El número de términos que se tienen en el tesoro es igual al número de registros que están en uso en el archivo, dato que se muestra en la pantalla.

MOstrar UN TERMINO (L): Escribe en la pantalla el término, sus códigos de clasificación y todas las relaciones que tenga asignadas con otros términos.

LISTAR LOS TERMINOS (M): Para listar en pantalla o por impresora un grupo, o la totalidad, de los términos que estén en el tesoro.

IMPRIMIR LA SECCION ALFABETICA (N): Imprime la sección alfabética del tesoro.

IMPRIMIR LA SECCION CLASIFICADA (V): Imprime la sección clasificada del tesoro, total o parcialmente.

LISTAR CODIGOS Y ENCABEZADOS (O): Para listar un grupo, o la totalidad, de los códigos de clasificación con el encabezado que les corresponde en la sección clasificada.

INGRESAR NUEVOS CODIGOS (P): Se refiere a los códigos de clasificación con el encabezado que les corresponde.

MODIFICAR UN CODIGO (Q): Para modificar un código o el encabezado que le corresponde.

BORRAR UN CODIGO (R): Elimina el código y el encabezado asignado a ese código.

Quando se concluye el proceso solicitado el programa retorna a mostrar nuevamente el menú inicial, por lo que se podrá pedir otra o la misma opción.

Para finalizar el programa se utiliza la opción 'Z'.

ESTRUCTURA DE LA BASE DE DATOS DE UN TESAURO

=====

La información de cada tesoro está separada en cuatro archivos de datos, con un archivo de índices para cada archivo de datos, es decir que un tesoro está formado por ocho archivos.

El nombre de los archivos se forma en base al nombre del tesoro. Suponiendo que el tesoro se llame BIÓLOGO, el nombre de cada archivo se forma como sigue:

Archivo1: ABIÓLOGO.DBF	Archivo1 (índices): ABIÓLOGO.NDX
Archivo2: BBIÓLOGO.DBF	Archivo2 (índices): BBIÓLOGO.NDX
Archivo3: CBIÓLOGO.DBF	Archivo3 (índices): CBIÓLOGO.NDX
Archivo4: DBIÓLOGO.DBF	Archivo4 (índices): DBIÓLOGO.NDX

Como se puede notar lo que se hace es anteponer al nombre del tesoro las letras 'A', 'B', 'C' y 'D' para formar el nombre de cada uno de los archivos, de datos y de índices.

En las siguientes páginas se muestran los campos que forman cada uno de los archivos.

ARCHIVO1

Archivo1.DBF (archivo de datos)
Tiene los campos:

DESCRIPTOR	40	caracteres
NOTA	1	caracter
CUMPLE	1	caracter

Archivo1.NDX (índices para el archivo1.DBF)
Ordenado por: DESCRIPTOR

El campo DESCRIPTOR se emplea para almacenar los nombres de los términos, van a estar grabados en letra mayúscula sin importar la forma en la que fueran ingresados.

El campo NOTA puede estar en blanco o tener un asterisco (*). Si tiene el asterisco significa que el descriptor que está en ese registro tiene Nota de Alcance (en otro archivo que se explica más adelante). Si este campo está en blanco implica que el término no tiene Nota de Alcance.

El campo CUMPLE es usado temporalmente para marcar los descriptores que se imprimen en la sección clasificada, lo que se hace es asignar un asterisco (*) a este campo en los registros que deben ser escritos bajo un mismo encabezado, este asterisco se borra después de escribir cada descriptor.

Ejemplos de un registro de este archivo.

Registro: 0126
DOSAJE *

Registro: 0271
AZUCARES

Registro: 0402
LACTOSA

En el registro # 126 se tiene un descriptor con Nota de Alcance.

En los registros 271 y 402 se tienen términos sin Nota de Alcance.

ARCHIVO2

Archivo2.DBF (archivo de datos)

Tiene los campos:	DESDE	4	caracteres
	RELAC	1	caracter
	HASTA	4	caracteres

Archivo2.NDX (índices para el archivo2.DBF)

Ordenado por: DESDE+RELAC

Este archivo se usa para definir las relaciones entre los términos, así como para asignar a cada descriptor códigos de clasificación.

El campo DESDE tiene un número que corresponde al número de registro (en el archivo1) del término al que corresponde la información que está en cada registro. Este término se denomina término FUENTE de la relación.

El campo RELAC tiene un código que significa el tipo de relación que se está definiendo. Puede ser uno de los siguientes:

Si está en blanco se trata de un código de clasificación.

- '1' para definir la relación USE.
- '2' para definir la relación USADO POR.
- '4' para un término MAS AMPLIO.
- '6' para un término MAS ESPECIFICO.
- '8' para TERMINOS RELACIONADOS.

El campo HASTA tiene un número que corresponde al número de registro (en el archivo1) del término OBJETO de la relación, es decir el término con el cual se tiene la relación definida por el campo RELAC.

En el caso de que el campo RELAC fuera un espacio en blanco el campo HASTA no tendría el número de registro del término OBJETO sino que un código de clasificación.

Por ejemplo:

Registro: 0018 DOSAJE (0126) tiene el código JAF
0126 JAF (el campo RELAC está en blanco)

Registro: 0125 AZUCARES (0271) tiene como MAS ESPECIFICO(6)
027160402 LACTOSA (0402)

Registro: 0126 LACTOSA (0402) tiene como MAS AMPLIO (4)
040240271 AZUCARES (0271)

ARCHIVO3

Archivo3.DBF (archivo de datos)

Tiene los campos:	CODIGO	4	caracteres
	DETALLE	40	caracteres
	ES:O:NO	1	caracter

Archivo3.NDX (índices para el archivo3.DBF)

Ordenado por: CODIGO

Este archivo tiene los encabezados de la sección clasificada, pero no tiene en sí la sección clasificada.

El campo CODIGO tiene cada uno de los códigos que se emplean en la sección clasificada.

El campo DETALLE tiene el encabezado que corresponderá a los descriptores que tengan el mismo CODIGO.

El campo ES:O:NO identifica con un asterisco (***) a los encabezados de la sección clasificada que también se usan como descriptores.

Por ejemplo:

Registro: 007
CJ ENERGIA

Código: CJ
Encabezado: ENERGIA

Registro: 023
CMA TRANSPORTE

Código: CMA
Encabezado: TRANSPORTE

Registro: 026
EMAFPLANTAS DE ESPECIAS

Código: EMAF
Encabezado: PLANTAS DE ESPECIAS

Registro: 071
EMB PLANTAS

Código: EMB
Encabezado: PLANTAS

Registro: 145
EQ NUTRICION

Código: EQ
Encabezado: NUTRICION

Registro: 621
JADAPLANIFICACION

Código: JADA
* Encabezado: PLANIFICACION
Aparece también como descriptor.

ARCHIVO4

Archivo4.DBF (archivo de datos)

Tiene los campos:	DESDE	4	caracteres
	LINEA1	30	caracteres
	LINEA2	30	caracteres
	LINEA3	30	caracteres
	LINEA4	30	caracteres

Archivo4.NDX (índices para el archivo4.DBF)

Ordenado por: DESDE

Este archivo se emplea para almacenar las Notas de Alcance que se necesiten para los descriptores.

El campo DESDE tiene el número de registro (en el archivo1) del descriptor al que corresponde la nota de alcance.

Los campos Linea1, 2, 3 y 4 tienen en sí la nota de alcance.

Por ejemplo:

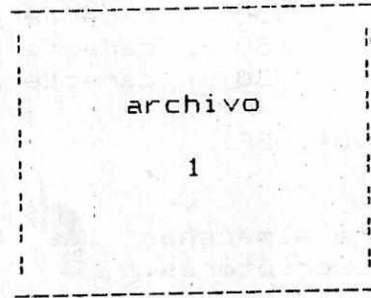
Registro: 004

0126LA OPERACION DE AGREGAR CIERTA
CANTIDAD DE UNA SUBSTANCIA.

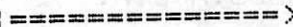
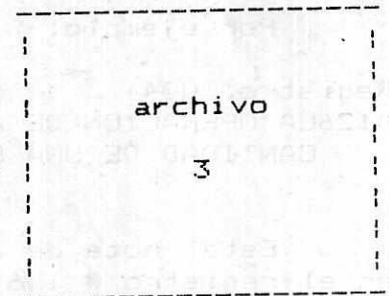
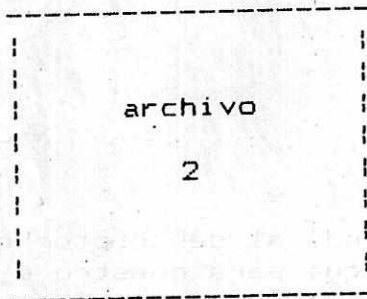
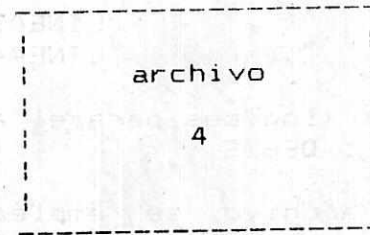
Esta nota de alcance corresponde al descriptor que está en el registro # 126 del archivo1, que para nuestro ejemplo es el término DOSAJE.

RELACION EXISTENTE ENTRE LOS ARCHIVOS

TERMINOS



NOTAS DE ALCANCE



RELACIONES ENTRE
TERMINOS y CODIGOS
DE CLASIFICACION POR
DESCRIPTOR

CODIGOS Y
ENCABEZADOS PARA
CLASIFICACION

CANTIDAD DE TERMINOS QUE SE PUEDEN TRABAJAR CON ESTE SISTEMA
=====

- 1) For el diseño del programa se pueden trabajar hasta 9999 términos. (Se podría cambiar modificando el tamaño de algunos campos.)
- 2) For el espacio disponible para almacenamiento:
 - a) Con diskette de 360 Kbytes, aprox. 1118 términos.
 - b) con disco de 10 Megabytes, aprox. 31055 términos.
(Podría cambiar si se reduce o aumenta el número de caracteres que puede tener un término)
- 3) For número de registros que permite trabajar el dBASEII: Permite 65000 registros en cada archivo. El archivo2 tiene aprox. 8 registros por cada descriptor por lo que se podría calcular el número de términos en 8125. (No se puede modificar, pero si se tratara de una lista de términos sin relaciones entre ellos se podrían llegar a tener 65000 términos.

PARA ESTE CALCULO SE SUPONE QUE:

a) El espacio que ocupa un archivo de índices es aproximadamente 1,25 veces el espacio ocupado por el archivo de datos, por lo que se tiene para cada archivo,

archivo1.dbf longitud	43,	archivo1.ndx longitud	54
archivo2.dbf longitud	10,	archivo2.ndx longitud	13
archivo3.dbf longitud	46,	archivo3.ndx longitud	58
archivo4.dbf longitud	125,	archivo4.ndx longitud	157

b) Que para cada término (un registro del archivo1) se tienen ocho registros en el archivo2, incluyendo códigos de clasificación y relaciones con otros términos.

c) Que uno de cada veinte términos tiene Nota de Alcance.

CONCLUSION

For los datos anteriores, incluyendo las suposiciones, se estima que contando con un disco duro de 10 megabytes se puede trabajar un tesoro de 8000 términos y aún tener espacio de almacenamiento para otros archivos y tareas.

CODIGO DE ACCESO

Al entrar, el sistema solicita el código de acceso o "password". Si el usuario conoce y da este código, tendrá acceso a todas las opciones. En caso contrario - da un código incorrecto o no ingresa código alguno - solamente tendrá capacidad de lectura (ver la sección "Preparando el sistema para trabajar", p. 5).

SELECCION DE OPCIONES

Aparece en pantalla un menú para seleccionar la acción deseada. Inicialmente será necesario crear un tesoro.

CREAR UN TESAURO

Se selecciona la opción "S" del menú. El sistema pide el nombre del tesoro (por ejemplo BIOLOG, POBLAC) que debe ser diferente a los nombres de tesauros ya existentes, tener un máximo de 7 caracteres, empezar con una letra y estar conformado de solo letras o de letras y dígitos pero sin espacios en blanco ni caracteres especiales.

El sistema pide también la identificación del impulsor (drive) en que se almacenará el tesoro a lo largo de la sesión de trabajo.

Puesto que el sistema puede demorar unos minutos en crear los archivos necesarios, aparece un mensaje. Ya creado el tesoro aparece de nuevo el menú.

Ya sea al arrancar o después de crear un tesoro estamos aún en el tesoro inicial o de arranque (EJEMPLO) y se debe entonces

CAMBIAR EL TESAURO EN USO

Se pide desde el menú con la letra "U". El sistema pide identificar el nombre del tesoro deseado y el impulsor en el cual estará ubicado. Vuelve al menú pero ya con el tesoro de trabajo.

INGRESAR TERMINOS INDIVIDUALES

Se selecciona con la letra "A". El sistema pide el término el cual puede ser ingresado en mayúsculas o en minúsculas pero será siempre convertido a mayúsculas y con un máximo de 40 caracteres. También se pide la o las clasificaciones del término (máximo de 4 caracteres) aunque si no se desea ingresarlas en este punto es posible hacerlo posteriormente. Si no se desea dar clasificación o cuando se han ingresado ya todas, se deja en blanco enviando con "return". Cuando ya no se desea ingresar términos se oprime "return" sin especificar término.

Para no tener que ingresar cada vez la clasificación cuando es la misma para varios términos, es posible

INGRESAR TERMINOS AGRUPADOS

Esta opción se pide con la letra "B" en el menú. El sistema pide primero el número de clasificación; ya escrito éste pide el término; en cada ciclo, si no se cambia la clasificación y se oprime "return", se está ingresando un nuevo término bajo esa clasificación. Al cambiar la clasificación se inicia un nuevo ciclo para poder ingresar otro conjunto de términos agrupados. Para finalizar, dejar en blanco el campo de código de clasificación oprimiendo la barra espaciadora y presionar "return" hasta volver al menú.

Cuando ya se tienen términos ingresados se establecen sus relaciones entre sí y para ello se debe

CREAR UNA RELACION

Esta opción se pide con la letra "C". El sistema pide el descriptor fuente. Este debe ser un término ya ingresado; si no lo es el sistema pide que se ingrese con la opción "A". A continuación se especifica qué tipo de relación se desea crear (USE, UP = use por, MA = más amplio, ME = más específico, TR = término relacionado). Luego solicita el término objeto. Si éste aún no existe, el sistema puede agregarlo si así lo pide el usuario o bien puede ignorarlo y no crear la relación. Si se pide agregar el término, el sistema pide también su número de clasificación que puede dejarse en blanco. El sistema sigue pidiendo relaciones para el descriptor fuente en tanto no se envíe en blanco el campo de relaciones es decir, se borre con la barra espaciadora y se oprima "return".

BORRAR UNA RELACION.

Para ello se usa la letra "D". Para verificar antes de borrar, usar la opción "L" para listar el término con todos sus datos. Así es posible dar una relación y un término objeto ya existentes ya que en caso contrario se tendrá un mensaje de que no se encontró la relación.

BORRAR UN TERMINO

Antes de efectuar esta operación es conveniente verificar el término a eliminar con la opción "L". La opción de borrar se activa con la letra "E". Pide el término a eliminar y una vez enviado el sistema lo borra sin pedir confirmación por lo que es importante la verificación visual. Si no existe el término a borrar se recibe el mensaje correspondiente.

En el caso de un error mecanográfico o si se decide cambiar la escritura de un término, se usará la opción

MODIFICAR UN TERMINO

También en este caso es recomendable verificar el término con la opción "L" antes de hacer el cambio. La opción de modificación se llama con "F". Pide el término a modificar y si éste no existe lo indica. Cuando el término sí existe, pide que se escriba en forma correcta.

QUIDADADO! si decide no hacer cambios cuando el sistema ya le está pidiendo el término en forma correcta, reescriba el término pero no oprima <return> con el campo vacío o incompleto ya que entonces estaría almacenando información incorrecta.

AGREGAR CODIGOS DE CLASIFICACION A UN TERMINO

Si, por ejemplo, ingresamos términos individuales con la opción "A" sin dar la clasificación o no dimos la clasificación del término objeto al crear una relación, podemos agregar los códigos con la opción "G". La pantalla muestra las clasificaciones ya existentes para el término así que en este caso no es necesaria verificación previa.

Si nos hemos equivocado de código es necesario borrarlo y escribirlo de nuevo ya que no hay opción para modificarlo directamente.

ELIMINAR CODIGOS DE CLASIFICACION

Se pide la opción "H". Al dar el término, el sistema muestra todos los códigos asignados y se pueden eliminar uno o varios. Si se da un código que no existe, un mensaje pedirá que se revisen los códigos. Cuando ya no se desee eliminar más códigos se oprime "return".

PONER UNA NOTA DE ALCANCE

Esto se hace con la opción "I" y se cuenta con 4 líneas de 30 caracteres cada una para escribir la nota de alcance. Cuando se haya terminado de escribir y no se hayan utilizado todas las líneas, se oprime "return" hasta volver al menú. La nota de alcance quedará escrita con mayúsculas y minúsculas, tal y como se ingresó.

QUITAR UNA NOTA DE ALCANCE

Aquí también puede ser conveniente revisar el término con la opción "L" previo a pedir la opción "J" de eliminar notas de alcance. Si al dar el término, éste no tiene nota de alcance, el sistema lo indica. Si tiene nota de alcance, la borra y vuelve al menú.

MOSTRAR UN TERMINO

La opción "L" del menú es la que se usa para verificaciones previas a cambios de cualquier tipo. El sistema da la opción de ordenar la información antes de presentarla y luego muestra el término con todos sus códigos de clasificación, notas de alcance y relaciones existentes.

LISTAR LOS TERMINOS

Con esta opción, "M", es posible listar - en pantalla o en impresora - todos o un grupo de términos para lo cual se especifica desde qué letra hasta qué letra se desea el listado y si se quiere o no en impresora. Se obtiene un listado con el número de registro y el término.

IMPRIMIR LA SECCION ALFABETICA

Opción "N". Pueden ordenarse los términos antes de imprimirlos. Si solamente se desea una parte de la sección alfabética se especifica a partir de qué término y hasta qué término se desea la impresión. La impresión es completa, es decir que cada término se imprime con sus clasificaciones, notas de alcance y relaciones.

LISTAR CODIGOS DE CLASIFICACION Y SUS ENCABEZADOS

Opción "O". Puede listar en pantalla o en impresora, todos los códigos y sus respectivos encabezamientos o solamente una sección desde una hasta otra letra especificadas.

La sección clasificada no se crea automáticamente con el ingreso de los términos individuales y es necesario usar la opción "P" para establecer los códigos y sus encabezados. Para ver la sección clasificada completa (código, encabezado, términos) se usa la opción "V" de impresión.

INGRESAR CODIGOS DE CLASIFICACION Y SUS ENCABEZADOS

Opción "P". Pide el código, si ya existe lo indica y pide otro. Al dar un código nuevo, pide el encabezado y pide de nuevo código hasta que le demos un ingreso vacío con "return". Cuando el encabezado es a la vez un descriptor, esto se indica con un asterisco. El encabezado quedará exactamente como se haya ingresado, con mayúsculas y minúsculas.

MODIFICAR CODIGOS DE CLASIFICACION Y ENCABEZADOS

Opción "Q". Pide el código y si no existe lo indica y pide otro. Al ingresar un código existente, da la opción de modificarlo así como el encabezado y de asignar asterisco; luego pide otro código.

BORRAR CODIGOS DE CLASIFICACION Y ENCABEZADOS

Antes de llevar a cabo esta operación es conveniente una verificación de códigos con la opción "O". La opción "R" permite borrar un código y su encabezado. Si el código solicitado no existe, lo indica y pide otro. Debe verificarse visualmente antes de oprimir "return" ya que el sistema no pide confirmación antes de eliminar el código y sus encabezados.

IMPRIMIR LA SECCION CLASIFICADA

Opción "V". Da la posibilidad de imprimir toda la sección clasificada o solamente una porción desde una letra dada a otra. La impresión da los datos completos: código de clasificación, encabezado, asterisco si el encabezado es a su vez un descriptor, y todos los términos que responden a esa clasificación y la tienen asignada en la sección alfabética.

Finalmente, para eliminar totalmente un tesoro, por ejemplo uno de prueba empleado para familiarizarse con el sistema, se usa

BORRAR UN TESAURO

Opción "T". Si el tesoro a borrar es el tesoro en uso, el sistema pide cambio de tesoro (opción "U"); esto evita que accidentalmente pueda borrarse el tesoro en el que se está trabajando. Si no existen otros tesoros y puesto que no se puede volver al de arranque (EJEMPLO), será necesario salir del sistema con la opción "Z", reingresar al sistema y borrar desde el tesoro de arranque (EJEMPLO). Si se tienen otros tesoros, la operación de borrar el deseado se puede hacer desde cualquier otro. El sistema pide confirmación y al recibirla borra el tesoro. Si no existe el tesoro pedido, el sistema lo indica y para verificar se puede hacer un directorio (ver DETERMINAR QUE TESAUROS EXISTEN YA EN EL SISTEMA, p. 16).

NUMERO DE REGISTROS EN USO

Para determinar la situación de todos los archivos del tesoro en uso. Se llama con la opción "K".

FINALIZAR

Al dar la letra "Z" se da por finalizada la sesión y la ejecución vuelve al sistema operativo.

DOCUMENTACION DE LOS PROGRAMAS

=====

En esta sección se describe la función que realiza cada instrucción, o grupo de instrucciones, dentro de los diferentes programas que forman el sistema de manejo de tesauros. Para los programas que imprimen reportes se acompañan pruebas de esos reportes.

Las líneas que dentro de los programas aparecen con un asterisco ('*') no serán mencionadas, estas no son instrucciones del dBASE II sino comentarios que ayudan a identificar cada programa.

Estos programas fueron preparados en una microcomputadora APPLE II+ con sistema operativo CP/M, memoria principal de 64 Kbytes, impresora EPSON MX-80, dos unidades de diskette de 126 Kbytes cada una y una unidad de almacenamiento CORVUS de .20 Megabytes, utilizando como lenguaje de programación para manejo de archivos el DBASE II. En este sistema se tomó como default el disco B:.

Los "mismos programas" se escribieron en una microcomputadora IBM-XT, con sistema operativo DOS, memoria principal de 256 Kbytes, impresor IBM, una unidad de diskette de 360 Kbytes y una unidad de almacenamiento de 10 Megabytes. Se trabajó como default el disco C:.

Programa INSTAL

Es empleado para dejar grabado en el archivo TESAURO.MEM el código, o palabra clave, que identificará a las personas que pueden modificar la información que esta almacenada en un tesoro.

Este programa es llamado en forma individual, por lo que da la instrucción SET TALK OFF (línea 2) para que el DBASE no escriba en la pantalla mensajes de control.

La instrucción ERASE (línea 3) borra la pantalla.

El programa presenta un mensaje al usuario para informarle que se va a modificar, o a ingresar por primera vez, el código, o palabra clave, del sistema de tesoros (líneas 4 a 11).

La instrucción CLEAR (línea 12) borra cualquier variable que se tuviera en uso.

El programa lee la respuesta del usuario almacenándola en la variable MPASS (línea 13). Luego lo pasa a letras mayúsculas (línea 14).

Se graba esta variable en el archivo TESAURO.MEM en el disco B: (línea 15). La instrucción SAVE crea un archivo con el nombre que se le indique, pero le agrega la extensión .MEM para identificarlo como un archivo de memoria.

Ya que se grabó la información se presenta de nuevo el código, tal como fue grabado (líneas 16 y 17).

Fin del programa (línea 18).

El código puede ser modificado cuantas veces sea necesario.

```
1 * PROGRAMA PARA COLOCAR EL PASSWORD AL SISTEMA DE TESAUROS
2 SET TALK OFF
3 ERASE
4 ? *PROGRAMA PARA CAMBIAR, O INGRESAR POR PRIMERA VEZ, EL CODIGO*
5 ? *QUE IDENTIFICA A LOS USUARIOS QUE PUEDEN MODIFICAR LA INFORMACION*
6 ? *DE LOS TESAUROS.*
7 ? *EL CODIGO DEBE ESTAR FORMADO POR UNA SERIE DE CARACTERES CUALES QUIER
8 ? *(256) COMO MAXIMO. LAS LETRAS MINUSCULAS SE CAMBIARAN POR MAYUSCULAS
9 ?
10 ? *INGRESE EL CODIGO*
11 ?
12 CLEAR
13 ACCEPT TO MPASS
14 STORE ! (MPASS) TO MPASS
15 SAVE TO B:TESAURO
16 ?
17 ? *EL CODIGO ES: *+MPASS
18 QUIT
19 *
```

**BIBLIOTECA
DE LA
UNIVERSIDAD DEL VALLE DE GUATEMALA**

Programa TESAURO

Este es el programa principal, presenta el MENU inicial y coordina la actividad de todo el sistema. Lo primero que hace es dar instrucciones al dBASE, que definen la forma en que se va a trabajar:

CLEAR: Cierra las bases de datos que se encontraran en uso, y borra cualquier información previa sobre otros procesos.

SET TALK OFF: Indica que no se desean mensajes de control. Se da esta instrucción porque el programa verifica cada una de las etapas en las cuales podría darse un mensaje de éstos, y toma acciones correspondientes para cada proceso. Con esto se logra que en la pantalla solo se presenten mensajes relacionados con el trabajo que se desarrolla.

SET EJECT OFF: Para que al imprimir un reporte no se deje la primera hoja en blanco, lo cual sucedería si no se diera esta instrucción.

SET EXACT ON: Para que en el momento de comparar dos términos con el fin de saber si son iguales o no, se tomen en cuenta todos los caracteres que forman cada término, y no sólo lo del que tenga menos caracteres, por ejemplo: CONTROL no es lo mismo que CONTROL DE MALEZAS, pero si no se diera esta instrucción el programa los tomaría como iguales.

SET INTENSITY OFF: Para que utilice, en la pantalla, letras verdes sobre fondo negro. **SET DEFAULT TO B:** Para indicar que se utilizarán programas que están grabados en el disco B:.

Se da la instrucción de verificar en el directorio del diskette si existe el archivo TESAURO.MEM (línea 8), ya que en ese archivo está almacenado el código del sistema de tesauros, el que se emplea para identificar a los usuarios que pueden modificar la información existente en los archivos. Si no se encuentra este archivo se da un mensaje al usuario informándole que es necesario que ese archivo esté en el diskette (o disco), para identificar a los usuarios, por lo que el programa se termina con la instrucción QUIT (líneas 9 a 11).

Si se encuentra el archivo se recupera la información que contiene por medio de la instrucción RESTORE (línea 13). Este archivo, es decir TESAURO.MEM, tiene el código del sistema almacenado en la variable MPASS.

Se borra la pantalla (línea 14) y se pide al usuario su nivel de acceso, o palabra clave (línea 15). La instrucción SET CONSOLE OFF (línea 16) se da para que no se muestre en la pantalla lo que el usuario escriba como respuesta, lo cual quedará almacenado en la variable TEMP (línea 17). Después de leer la respuesta del usuario se retorna la pantalla a su estado normal (línea 18).

El programa compara la variable TEMP con la variable MPASS, que tiene el código del sistema, (línea 19) para verificar si el usuario ingresó el código correcto; si así fuera almacena el valor 'Falso' en la variable PASSN, si no almacena un valor 'Verdadero' (líneas 20 a 23). Si la variable PASSN tiene valor 'Verdadero' el usuario no podrá modificar la información existente en los archivos, ni crear nuevos tesauros.

Línea 24: Instrucción de almacenar 40 espacios en blanco en la variable BLANCO. Esto servirá para indicar, al usuario, cual es el máximo de caracteres que puede utilizar al escribir un descriptor.

Línea 25: Se almacena en la variable TRABAJO el nombre del tesoro que se toma como inicial. El tesoro EJEMPLO es utilizado sólo para el inicio del programa, esto significa que si se cambia de tesoro en uso no se podrá retornar al tesoro EJEMPLO, a menos de que se finalice el programa y se vuelva a iniciar. Este tesoro tiene la estructura que se toma como base para la creación de otros tesauros, por lo que el programa tampoco permite que se borre.

De la línea 26 a la 29 se dan valores iniciales a las variables que indicarán cuáles son los nombres de los archivos que componen el tesoro que se trabaja.

En la línea 30 se indica cual es el archivo primario del tesoro, y en la 32 se dice cual es el secundario. Solamente se pueden trabajar dos, por eso cuando se necesite uno de los otros se tendrá que cerrar momentaneamente el archivo primario.

Las siguientes instrucciones que se describen constituyen un ciclo que se repite tantas veces como sea necesario, a fin de satisfacer las necesidades del usuario. Cada repetición de estas instrucciones indica una operación.

En la línea 33 se tiene la instrucción de REPETIR las siguientes instrucciones, hasta la línea 84, tantas veces como el usuario lo solicite. La forma de salir de este ciclo es que el usuario ingrese una letra 'Z'.

Primero se da la instrucción de borrar otras lecturas (línea 34), esto es para que el usuario no pueda escribir fuera del campo que se muestra en la pantalla. Luego se borra la pantalla (línea 35). Se almacena un espacio en blanco en la variable TEMP (línea 36) para indicar que la opción que de el usuario deberá ser de un caracter.

De la línea 37 a la 62 se escribe un mensaje en la pantalla, para mostrar las opciones que se tienen en el programa, y cuál es la forma de llegar a cada una de ellas. En la línea 63 se da la instrucción de leer la opción que quiera el usuario. Debido a que se permite el ingreso de letras

mayúsculas o minúsculas se pasa todo a mayúsculas dentro del programa por medio de la instrucción en la línea 64.

Si la opción es 'Z' se termina el programa (líneas 65 a 67).

Si no fuera el caso de terminar, se evaluará la opción ingresada, por el usuario, con el fin de saber si ésta es una opción válida (línea 68), si la opción no está entre las letras 'A' y 'V' se da un mensaje en la pantalla (líneas 69 a 73) y se presenta de nuevo el menú. Si es una opción válida el programa se asegura de que el usuario esté solicitando una opción a la que tenga acceso. Si la opción se refiere a modificación de los datos se revisa si el usuario ingresó el código del sistema (línea 76). Si el usuario no puede realizar esa operación se le escribe un mensaje en la pantalla (líneas 77 a 81) y se presenta de nuevo el menú. Si la opción es permitida se llamará al programa que corresponda. La forma de lograr esto es la siguiente:

En la variable TEMP (temporal) se tiene la letra que indica cuál es la opción que el usuario desea, cada una de las opciones presentadas corresponde a un programa. Los nombres de todos los programas tienen las primeras tres letras del nombre iguales (TES), varían de la cuarta en adelante. El programa de ingreso de nuevos términos se llama TESA, el de reportes se llama TESM. En la línea 84 se tiene la instrucción para llamar a otro programa: DO TES&TEMP. Al utilizar el signo '&' se indica que se deberá sustituir esa parte de la instrucción por el valor que tenga la variable TEMP en ese momento, por ejemplo: si se deseara la opción 'A' se tendría DO TESA, que corresponde al programa de ingreso de datos. Si se tuviera la letra 'M' se formaría el nombre del programa de reportes, es decir: TESM.

```

1  * MENU PARA EL MANEJO DEL TESAURO
2  CLEAR
3  SET TALK OFF
4  SET EJECT OFF
5  SET EXACT ON
6  SET INTENSITY OFF
7  *SET DEFAULT TO B
8  IF .NOT. FILE('TESAURO.MEM')
9      ? 'NO ENCUENTRO EL ARCHIVO TESAURO.MEM, ES NECESARIO PARA'
10     ? 'LA IDENTIFICACION DE LOS USUARIOS. USE EL PROGRAMA INSTAL.PRG'
11     QUIT
12 ENDIF
13 RESTORE FROM TESAURO
14 ERASE
15 ? 'INGRESE EL PASSWORD DEL SISTEMA. (O <RETURN> PARA SOLO CONSULTA)'
16 SET CONSOLE OFF
17 ACCEPT TO TEMP
18 SET CONSOLE ON
19 IF '(TEMP)=MPASS
20     STORE F TO PASSN
21 ELSE
22     STORE T TO PASSN
23 ENDIF
24 STORE ' ' TO BLANCO
25 STORE 'EJEMPLO' TO TRABAJO
26 STORE 'A' TO ARCHIVO1
27 STORE 'B' TO ARCHIVO2
28 STORE 'C' TO ARCHIVO3
29 STORE 'D' TO ARCHIVO4
30 USE &ARCHIVO1 INDEX &ARCHIVO1
31 SELECT SECONDARY
32 USE &ARCHIVO2 INDEX &ARCHIVO2
33 DO WHILE T
34     CLEAR GETS
35     ERASE
36     STORE ' ' TO TEMP
37
@ 1,10 SAY 'INSTITUTO CENTROAMERICANO DE INVESTIGACION Y TECNOLOGIA INDUSTRIAL'
38 @ 3,10 SAY ' I C A I T I '
39 @ 04,00 SAY 'Tesauro: '+TRABAJO
40 @ 06,00 SAY 'Presione la letra correspondiente a la opcion que desea'
41 @ 08,00 SAY 'A. Ingreso de terminos individuales'
42 @ 09,00 SAY 'B. Ingreso de terminos agrupados'
43 @ 10,00 SAY 'C. Crear una relacion'
44 @ 11,00 SAY 'D. Borrar una relacion'
45 @ 12,00 SAY 'E. Borrar un termino'
46 @ 13,00 SAY 'F. Modificar un termino'
47 @ 14,00 SAY 'G. Agregar codigos a un termino'
48 @ 15,00 SAY 'H. Quitar un codigo'
49 @ 16,00 SAY 'I. Poner nota de alcance'
50 @ 17,00 SAY 'J. Quitar nota de alcance'

```

```

51 @ 18,00 SAY 'K. Numero de registros en uso ?'
52 @ 08,40 SAY 'L. Mostrar un termino'
53 @ 09,40 SAY 'M. Listar los terminos'
54 @ 10,40 SAY 'N. Imprimir la seccion alfabetica'
55 @ 11,40 SAY 'O. Listar los codigos y encabezados'
56 @ 12,40 SAY 'P. Ingresar nuevos codigos'
57 @ 13,40 SAY 'Q. Modificar un codigo'
58 @ 14,40 SAY 'R. Borrar un codigo'
59 @ 15,40 SAY 'S. Crear un Tesauro'
60 @ 16,40 SAY 'T. Borrar un Tesauro'
61 @ 17,40 SAY 'U. Cambiar de Tesauro en uso'
62 @ 18,40 SAY 'V. Imprimir la seccion clasificada'
63 @ 19,39 SAY '<Z> PARA FINALIZAR ' GET TEMP
64 READ
65 STORE !(TEMP) TO TEMP
66 IF TEMP='Z'
67     QUIT
68 ENDIF
69 IF TEMP < 'A' .OR. TEMP > 'V'
70     @ 23,10 SAY 'ESTA OPCION NO ES VALIDA. PRESIONE <RETURN>'
71     SET CONSOLE OFF
72     WAIT
73     SET CONSOLE ON
74     LOOP
75 ENDIF
76 IF PASSN .AND. .NOT. TEMP $ 'LMNOUVK'
77     @ 22,10 SAY 'PARA ESTA OPCION NECESITA EL CODIGO DEL SISTEMA'
78     @ 23,10 SAY 'EL CUAL NO FUE INGRESADO. PRESIONE <RETURN>'
79     SET CONSOLE OFF
80     WAIT
81     SET CONSOLE ON
82     LOOP
83 ENDIF
84 DO TES&TEMP
85 ENDDO
86 *
```

Programa TESA

Este programa ingresa nuevos términos al tesoro que se este trabajando. Un nuevo término indica un nuevo registro en el archivo.

Se selecciona el archivo (línea 3), a continuación se tiene la instrucción para repetir lo que se presenta entre las líneas 5 y 29. Se borra la pantalla (línea 5). Se prepara la variable TEMP para la lectura de un descriptor (línea 6). Se escribe en la pantalla el mensaje que indica que se debe ingresar un nuevo término (líneas 7 a 9). El programa lee el término (línea 10). Si el usuario dejó en blanco la variable TEMP significa que no desea ingresar más términos (líneas 11 a 13) por lo que se retorna al menú inicial.

Los términos pueden ser ingresados en letra mayúscula o minúscula pero el programa lo trabajará y dejará grabado en el diskette en letra mayúscula, esta es una limitación del sistema pero que es indispensable para poder realizar las comparaciones entre términos.

Se pasa a mayúscula el término (línea 14). El programa busca este término en el archivo con el fin de no tener duplicados (línea 15). Si ya existiera este término se escribe un mensaje en la pantalla (líneas 17 a 22) y se pide otro término. Si el término no existe en el archivo se busca un registro disponible para agregarlo, si no hay se agrega un registro en blanco (líneas 24 a 27). Se coloca en el registro en blanco la información ingresada por el usuario (línea 28) y se llama a la rutina CLASIF que pedirá la clasificación de este término (línea 29).

archivo: DBASENTESAURONTESA.PRG
pagina: 1

```
1  * TESA.PRG
2  * PROGRAMA PAR INGRESAR NUEVOS DESCRIPTORES A LA BASE DE DATOS
3  SELECT PRIMARY
4  DO WHILE T
5      ERASE
6      STORE BLANCO TO TEMP
7      @ 05,20 SAY '** INGRESO DE TERMINOS NUEVOS **'
8      @ 10,10 SAY 'PARA RETORNAR AL MENU SOLO PRESIONE <RETURN>'
9      @ 12,10 SAY 'INGRESE EL TERMINO ' GET TEMP
10     READ
11     IF TEMP = ' '
12         RETURN
13     ENDIF
14     STORE !(TEMP) TO TEMP
15     FIND &TEMP
16     IF # > 0
17         @ 20,10 SAY 'ESTE DESCRIPTOR YA EXISTE EN EL ARCHIVO'
18         @ 21,10 SAY 'PRESIONE <RETURN> PARA CONTINUAR'
19         SET CONSOLE OFF
20         WAIT
21         SET CONSOLE ON
22     LOOP
23     ENDIF
24     GO TOP
25     IF DESCRIPTOR <> ' ' .OR. EOF
26         APPEND BLANK
27     ENDIF
28     REPLACE DESCRIPTOR WITH TEMP
29     DO CLASIF
30 ENDDO
31 *
```

Programa TESB

Este programa permite el ingreso de nuevos descriptores que tengan en común su clasificación. Es decir que se ingresara un código de clasificación y luego un grupo de descriptores, con el fin de no estar ingresando el mismo código de clasificación para cada uno de los descriptores.

Se borra la pantalla (línea 3), se prepara la variable MCODEIGO para recibir cuatro caracteres (línea 4) y se escribe el mensaje de que se van a ingresar nuevos descriptores al archivo (línea 5).

En la línea 6 se da la instrucción de repetir las siguientes instrucciones hasta que el usuario desee terminar.

Se prepara la variable TEMP para recibir el nombre de un descriptor (línea 7).

Se da el mensaje para que el usuario ingrese el código de clasificación y el descriptor (líneas 8 a 10). Una vez ingresado el código permanece, en la pantalla hasta que el usuario lo modifique.

El programa lee los datos (línea 11). Si el código o el descriptor fueran dejados en blanco significa que ya no se desea continuar en esta etapa, por lo que se retornará al menú inicial (líneas 11 a 14).

Se pasa a mayúsculas la información ingresada por el usuario (líneas 15 y 16).

Se selecciona el archivo1 y se busca si existe el descriptor ingresado por el usuario (líneas 17 y 18). Si ya existe se llama a la rutina CEROS para almacenar en una variable el número de registro en el cual se encontró (línea 20). Se prepara la variable BUSCAR para identificar la información de este descriptor en el archivo2 (línea 21). Se selecciona el Archivo2 y se busca si ya tiene esta clasificación (líneas 23 a 30), si ya la tiene se da un mensaje en la pantalla y se piden nuevos datos (líneas 31 a 38).

Si el descriptor no existe se busca un registro en blanco para ubicarlo, si no hay se agrega uno en blanco (líneas 40 a 43). Se ubica en este registro la información dada por el usuario (línea 44) y se llama a la rutina CEROS (línea 45).

Se selecciona el Archivo2 y se busca un registro disponible para ubicar el nuevo código para este descriptor, si no hay se agrega uno en blanco (líneas 48 a 51) y se graba la información especificada por el usuario (línea 52).

```
1 * TESB.PRG
2 * PROGRAMA PARA INGRESAR TERMINOS AGRUPADOS POR CLASIFICACION
3 ERASE
4 STORE ' ' TO MCODIGO
5 @ 05,10 SAY '** INGRESO DE NUEVOS TERMINOS AGRUPADOS POR CLASIFICACION **'
6 DO WHILE T
7   STORE BLANCO TO TEMP
8   @ 08,10 SAY 'PARA VOLVER AL MENU PRESIONE <RETURN>'
9   @ 10,10 SAY 'DE EL CODIGO CLASIFICACION ' GET MCODIGO
10  @ 12,10 SAY 'INGRESE UN TERMINO ' GET TEMP
11  READ
12  IF MCODIGO=' ' .OR. TEMP=' '
13    RETURN
14  ENDIF
15  STORE !(TEMP) TO TEMP
16  STORE '(MCODIGO) TO MCODIGO
17  SELECT PRIMARY
18  FIND &TEMP
19  IF #>0
20    DO CEROS
21    STORE REGO+' ' TO BUSCAR
22    SELECT SECONDARY
23    FIND &BUSCAR
24    STORE F TO EXISTE
25    DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF .AND. .NOT. EXISTE
26      IF HASTA=MCODIGO
27        STORE T TO EXISTE
28      ENDIF
29      SKIP
30    ENDDO
31    IF EXISTE
32      @ 23,10 SAY 'YA FUE INGRESADO ESE TERMINO. PRESIONE <RETURN>'
33      SET CONSOLE OFF
34      WAIT
35      SET CONSOLE ON
36      @ 23,10 SAY '
37      LOOP
38    ENDIF
39  ELSE
40    GO TOP
41    IF DESCRIPTOR <> ' ' .OR. EOF
42      APPEND BLANK
43    ENDIF
44    REPLACE DESCRIPTOR WITH TEMP
45    DO CEROS
46  ENDIF
47  SELECT SECONDARY
48  GO TOP
49  IF DESDE <> ' ' .OR. EOF
50    APPEND BLANK
```

archivo: DBASE\TESAURO\TESB.PRG
pagina: 2

```
51      ENDIF  
52      REPLACE DESDE WITH REGO, HASTA WITH MCODIGO  
53  ENDDO  
54  *
```

Programa TESC

Empleado para agregar nuevas relaciones entre términos.

Selecciona el archivo1 (línea 3), luego prepara la variable TEMP para leer un descriptor (línea 4). Se borra la pantalla (línea 5), se da el mensaje para que ingresen el descriptor al que se le asignará la relación (descriptor FUENTE) (líneas 6 a 8), el programa lee la opción (línea 9).

Si se deja la variable en blanco significa que no se desea continuar, por lo que se retorna al menú (líneas 10 a 12).

Se pasan los datos a mayúsculas (línea 13) y se busca el descriptor en el archivo (línea 14). En el caso de que este término no esté en el archivo se dará un mensaje en la pantalla y se retornará al menú principal (líneas 16 a 22). Para esta rutina es necesario que exista el descriptor FUENTE.

Si el descriptor existe en el archivo se llama a la rutina CEROS que retorna la variable REGO que proporciona el número de registro del descriptor (línea 24).

El número de registro del descriptor que leímos va a ser utilizado para reemplazar uno de los campos en el archivo de relaciones, es por eso que lo almacenamos en otra variable llamada REG1 (línea 25).

Ya con el número de registro del descriptor FUENTE almacenamos un valor 'verdadero' en la variable VARIOS (línea 26), además utilizamos la variable RELACION con tres espacios en blanco (línea 27) para leer el tipo de relación que el usuario desea anotar.

En la línea 28 se inicia la repetición de instrucciones para ingresar cada una de las relaciones de el término leído anteriormente; la repetición llega hasta la línea 125.

Se selecciona el archivo1 (línea 29), se escribe el mensaje indicando que se deberá escribir el tipo de relación que se desea anotar (líneas 30 y 31). Las relaciones se deberán escribir de la siguiente forma:

Relación

- USE Use otro descriptor en vez de éste.
- UP Este es usado en vez de otro término.
- MA Para poner un término más amplio.
- ME Para un término más específico.
- TR Para un término relacionado.

En la línea 32 el programa lee el tipo de relación y la pasa a mayúsculas (línea 33).

Se evalúa la respuesta ingresada por el usuario (líneas 35 a 59), en donde se asigna un código a la variable MREL y otro a la variable NREL dependiendo del tipo de relación que se desee. MREL tiene la relación solicitada por el usuario y NREL tiene la relación complementaria.

Si se deja en blanco esta variable se retorna al menú principal (líneas 49 a 51). Si es un código distinto de los permitidos se da el mensaje correspondiente y se piden nuevos datos (líneas 52 a 58).

Ya sabiendo la relación que se desea, se pregunta por el descriptor OBJETO, o sea el descriptor con el cual se tiene la relación, el cual estará en la variable TEMP1 (líneas 61 a 63). Por ejemplo: Si se deseara crear la relación

FRUTAS

ME: MANZANA

Descriptor FUENTE: FRUTAS,
Relación: MAS ESPECIFICO y
Descriptor OBJETO: MANZANA.

Se pasa el descriptor OBJETO a mayúsculas (línea 64). Si dejan la respuesta en blanco se toma como indicación de que no desean crear más relaciones y se retorna al menú principal (líneas 65 a 68).

Si el descriptor OBJETO es igual al descriptor FUENTE se da el mensaje de que esa relación no puede ser creada, y se piden nuevos datos (líneas 69 a 76).

Si el descriptor FUENTE es distinto del descriptor OBJETO se buscará en el archivo el descriptor OBJETO para saber si ya fue ingresado (línea 77). En el caso de que no exista se presentará la opción de agregarlo (líneas 79 y 80), el programa lee la respuesta (línea 81) y la pasa a mayúsculas (línea 82). Se borra el mensaje escrito (línea 83). Si no desean agregar este término se regresa al menú principal (líneas 84 a 86). Si se desea agregar este término se busca un registro en blanco en el archivo, si no hay se agraga un registro en blanco (líneas 87 a 90) para colocar allí la información dada por el usuario (línea 91).

Si la relación que se ingresó es USADO POR, código '2', implica que el descriptor OBJETO no va a tener clasificación, por eso se llamará a la rutina que pide la clasificación sólo cuando la relación tenga código distinto a '2' (líneas 92 a 94).

Al estar en la línea 96 ya se tienen los tres datos que se necesitan para crear la relación: dos términos y una relación entre ellos. Antes de crear la relación se revisará el archivo para saber si esta relación ha sido ingresada con anterioridad.

Se llama a la rutina CEROS estando ubicados en el descriptor OBJETO para saber el número de registro que le corresponde (línea 96), se selecciona el archivo2 (línea 97). Se da valor inicial de "falso" a la variable EXISTE, la cual indicará si fue encontrada esta relación en el archivo (línea 98). Se prepara la variable de búsqueda BUSCAR (línea 99) y se busca en el archivo (línea 100). Se revisan todas las relaciones de este término y si en alguna se encuentra la información buscada da valor "verdadero" a la variable EXISTE (líneas 101 a 106). Si ya existe esta relación se hace saber al usuario y se retorna a pedir más datos (líneas 107 a 114).

En este momento se va a crear la relación solicitada por el usuario y la relación complementaria.

Para nuestro ejemplo anterior la relación complementaria sería:

MANZANA

MA: FRUTAS

Descriptor FUENTE: MANZANA,

Relación: MAS AMPLIO y

Descriptor OBJETO: FRUTAS.

Se busca un registro disponible para ubicar la primera relación, si no hay se agrega uno en blanco (líneas 115 a 118) y se reemplaza con los datos correspondientes (línea 119). Luego se busca un registro en blanco para la segunda relación, si no se encuentra se agrega uno (líneas 120 a 123) y se le graba su información (línea 124).

La línea 125 es el final de la repetición. Cuando el usuario no desee continuar se seleccionará el archivo1 y se retornará al menú principal.

```
1 * TESC.PRG
2 * PROGRAMA PARA AGREGAR UNA RELACION ENTRE TERMINOS
3 SELECT PRIMARY
4 STORE BLANCO TO TEMP
5 ERASE
6 @ 05,15 SAY ' ** CREAR UNA RELACION ENTRE TERMINOS ** '
7 @ 08,10 SAY ' PARA VOLVER AL MENU PRESIONE <RETURN> '
8 @ 10,05 SAY ' INGRESE EL DESCRIPTOR FUENTE ' GET TEMP
9 READ
10 IF TEMP=' '
11     RETURN
12 ENDIF
13 STORE !(TEMP) TO TEMP
14 FIND &TEMP
15 IF #=0
16     @ 15,05 SAY ' ESTE DESCRIPTOR NO EXISTE EN EL ARCHIVO, DEBE INGRESAR
17     @ 16,05 SAY ' CON LA OPCION "A" DEL MENU '
18     @ 17,05 SAY ' PRESIONE <RETURN> '
19     SET CONSOLE OFF
20     WAIT
21     SET CONSOLE ON
22     RETURN
23 ENDIF
24 DO CEROS
25 STORE REGO TO REG1
26 STORE T TO VARIOS
27 STORE ' ' TO RELACION
28 DO WHILE VARIOS
29     SELECT PRIMARY
30     @ 14,05 SAY BLANCO+BLANCO
31     @ 12,05 SAY ' QUE RELACION ? (USE, UP, MA, ME o TR) ' GET RELACION
32     READ
33     STORE !(RELACION) TO RELACION
34     DO CASE
35     CASE RELACION='USE'
36         STORE '1' TO MREL
37         STORE '2' TO NREL
38     CASE RELACION='UP'
39         STORE '2' TO MREL
40         STORE '1' TO NREL
41     CASE RELACION='MA'
42         STORE '4' TO MREL
43         STORE '6' TO NREL
44     CASE RELACION='ME'
45         STORE '6' TO MREL
46         STORE '4' TO NREL
47     CASE RELACION='TR'
48         STORE '8' TO MREL,NREL
49     CASE RELACION=' '
50         STORE F TO VARIOS
```

Titulo: DBASENTESAURONTESC.PRG
Pagina: 2

```
LOOP  
OTHERWISE  
  @ 23,10 SAY 'RELACION NO PERMITIDA. PRESIONE <RETURN>'  
  SET CONSOLE OFF  
  WAIT  
  SET CONSOLE ON  
  @ 23,10 SAY '  
  LOOP  
ENDCASE  
CLEAR GETS  
STORE BLANCO TO TEMP1  
@ 14,05 SAY 'INGRESE EL TERMINO OBJETO ' GET TEMP1  
READ  
STORE !(TEMP1) TO TEMP1  
IF TEMP1=' '  
  STORE F TO VARIOS  
  LOOP  
ENDIF  
IF TEMP1=TEMP  
@ 21,05 SAY 'LOS DESCRIPTORES DEBEN SER DISTINTOS. PRESIONE <RETURN>'  
  SET CONSOLE OFF  
  WAIT  
  SET CONSOLE ON  
@ 21,05 SAY '  
  LOOP  
ENDIF  
FIND &TEMP1  
IF #=0  
  STORE ' ' TO TEMP  
  @ 22,10 SAY 'ESTE TERMINO NO EXISTE. LO AGREGO (S/N) ' GET TEMP  
  READ  
  STORE !(TEMP) TO TEMP  
  @ 22,10 SAY '  
  IF TEMP<>'S'  
    RETURN  
  ENDIF  
  GO TOP  
  IF DESCRIPTOR <> ' '  
    APPEND BLANK  
  ENDIF  
  REPLACE DESCRIPTOR WITH TEMP1  
  IF MREL <>'2'  
    DO CLASIF  
  ENDIF  
ENDIF  
DO CEROS  
SELECT SECONDARY  
STORE F TO EXISTE  
STORE REG1+MREL TO BUSCAR  
FIND &BUSCAR
```

archivo: DBASE\TESAURO\TESC.PRG
pagina: 3

```
101 DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF .AND. .NOT. EXISTE
102     IF HASTA=REGO
103         STORE T TO EXISTE
104     ENDIF
105     SKIP
106 ENDDO
107 IF EXISTE
108
109 @ 19,10 SAY 'ESTA RELACION YA EXISTE. PRESIONE <RETURN> PARA CONTINUAR'
110     SET CONSOLE OFF
111     WAIT
112     SET CONSOLE ON
113
114 @ 19,10 SAY '
115     LOOP
116     ENDF
117     GO TOP
118     IF DESDE <> ' ' .OR. EOF
119         APPEND BLANK
120     ENDF
121     REPLACE DESDE WITH REG1, RELAC WITH MREL, HASTA WITH REGO
122     GO TOP
123     IF DESDE <> ' '
124         APPEND BLANK
125     ENDF
126     REPLACE DESDE WITH REGO, RELAC WITH NREL, HASTA WITH REG1
127 ENDDO
128 SELECT PRIMARY
129 RETURN
130 *
```

Programa TESD

Para eliminar una relación existente entre dos términos.

Selecciona el archivo1 (línea 3), prepara la variable TEMP para leer un término (línea 4). Se borra la pantalla (línea 5), el programa escribe un mensaje para que el usuario sepa que debe ingresar los datos de la relación que desea eliminar (líneas 6 a 8), lee el término (línea 9). Si dejan la variable en blanco se retorna al menú principal (líneas 10 a 12). Se pasa la variable TEMP a mayúscula (línea 13). Se busca el término en el archivo (línea 14). Si el término no existe se da aviso al usuario y se retorna al menú principal (líneas 15 a 22).

Si el descriptor FUENTE existe se llama a la rutina CEROS (línea 23) que retorna el número de registro en la variable REGO. Este número se guarda en la variable REG1 ya que más adelante se llamará de nuevo a esta rutina (línea 24).

Se prepara la variable REL para leer el tipo de relación que se desea borrar (línea 25). El programa pide el tipo de relación y lo lee (líneas 26 y 27) pasándolo a mayúsculas. Se evalúa la respuesta del usuario y se asigna un código a la variable MREL dependiendo de la relación de que se trate, y la relación complementaria a la variable NREL (líneas 29 a 43). Si no es una relación válida se notifica al usuario y se retorna al menú principal (líneas 44 a 50).

Se prepara la variable TEMP1 para leer un término y se pide al usuario que lo ingrese, en este caso se trata del descriptor OBJETO de la relación (líneas 51 a 53). Se pasa a mayúsculas el dato leído (línea 54). Si el usuario dejó la variable en blanco se retorna al menú principal (líneas 55 a 57). Se busca el término en el archivo1 (línea 58), si no existe se informa al usuario y se retorna al menú principal (líneas 59 a 65).

Si el descriptor OBJETO existe se llama a la rutina CEROS (línea 66), que retorna la variable REGO con el número de registro del descriptor OBJETO. Si el descriptor FUENTE es el mismo que el OBJETO se informa al usuario que tal relación no pudo ser grabada en el archivo, por lo cual no se puede borrar, teniendo que regresar al menú principal (líneas 67 a 73).

Se selecciona el Archivo2 (línea 74). Para buscar esta relación se unen los datos REG1 y MREL en la variable BUSCAR y se localizan los registros que cumplan con este valor (línea 77). Para verificar si se encontró esta información se da valor inicial de "falso" a la variable EXISTE (línea 76). Se revisan cada uno de los registros que cumplan esta condición para ver si esa es la relación buscada (líneas 78 a 84). Si se encuentra el registro se le reemplazan sus tres campos: DESDE, RELAC y HASTA con espacios en blanco para indicar que este registro está disponible (línea 80).

Si no se encontrara esta relación en el archivo se avisa al usuario y se retorna al menú inicial (lineas 85 a 91).

Se preparan las variables EXISTE y BUSCAR para localizar la relación complementaria (lineas 92 y 93), luego se buscan los registros que cumplan con esta condición (línea 94).

Se revisan todos los registros para ver si entre ellos se encuentra el que contiene los datos buscados (lineas 95 a 101). Si se encuentra se le reemplazan sus tres campos con espacios en blanco para dejar disponible ese registro (línea 97). Luego se selecciona el archivo y se retorna al menú inicial.

Archivo: DBASE\TESAURO\TESD.PRG
Pagina: 1

```
1 * TESD.PRG
2 * Programa para eliminar una relacion entre descriptores
3 SELECT PRIMARY
4 STORE BLANCO TO TEMP
5 ERASE
6 @ 05,15 SAY '** ELIMINAR UNA RELACION ENTRE DESCRIPTORES **'
7 @ 08,10 SAY 'SI DESEA VOLVER AL MENU PRESIONE <RETURN>'
8 @ 10,05 SAY 'INGRESE EL DESCRIPTOR FUENTE ' GET TEMP
9 READ
10 IF TEMP=' '
11     RETURN
12 ENDIF
13 STORE !(TEMP) TO TEMP
14 FIND &TEMP
15 IF #=0
16     @ 15,05 SAY 'ESTE DESCRIPTOR NO EXISTE EN EL ARCHIVO'
17     @ 16,05 SAY 'PRESIONE <RETURN>'
18     SET CONSOLE OFF
19     WAIT
20     SET CONSOLE ON
21     RETURN
22 ENDIF
23 DO CEROS
24 STORE REGO TO REG1
25 STORE ' ' TO REL
26 @ 12,05 SAY 'RELACION A ELIMINAR ? (USE, UP, MA, ME o TR) ' GET REL
27 READ
28 STORE !(REL) TO REL
29 DO CASE
30 CASE REL='USE'
31     STORE '1' TO MREL
32     STORE '2' TO NREL
33 CASE REL='UP'
34     STORE '2' TO MREL
35     STORE '3' TO NREL
36 CASE REL='MA'
37     STORE '4' TO MREL
38     STORE '6' TO NREL
39 CASE REL='ME'
40     STORE '6' TO MREL
41     STORE '4' TO NREL
42 CASE REL='TR'
43     STORE '8' TO MREL,NREL
44 OTHERWISE
45     @ 16,10 SAY 'RELACION NO VALIDA. PRESIONE <RETURN> PARA CONTINUAR'
46     SET CONSOLE OFF
47     WAIT
48     SET CONSOLE ON
49     RETURN
50 ENDCASE
```

```
51 STORE BLANCO TO TEMP1
52 @ 14,05 SAY 'INGRESE EL TERMINO OBJETO ' GET TEMP1
53 READ
54 STORE '(TEMP1) TO TEMP1
55 IF TEMP1='
56     RETURN
57 ENDIF
58 FIND &TEMP1
59 IF #=0
60     @ 16,10 SAY 'ESTE DESCRIPTOR NO HA SIDO INGRESADO. PRESIONE <RETURN>'
61     SET CONSOLE OFF
62     WAIT
63     SET CONSOLE ON
64     RETURN
65 ENDIF
66 DO CEROS
67 IF REGO=REG1
68     @ 10,10 SAY 'LOS DESCRIPTORES DEBEN SER DISTINTOS. PRESIONE <RETURN>'
69     SET CONSOLE OFF
70     WAIT
71     SET CONSOLE ON
72     RETURN
73 ENDIF
74 SELECT SECONDARY
75 STORE REG1+MREL TO BUSCAR
76 STORE F TO EXISTE
77 FIND &BUSCAR
78 DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF .AND. .NOT. EXISTE
79     IF HASTA=REGO
80         REPLACE DESDE WITH ' ', RELAC WITH ' ', HASTA WITH ' '
81         STORE T TO EXISTE
82     ENDIF
83     SKIP
84 ENDDO
85 IF .NOT. EXISTE
86     @ 23,10 SAY 'NO ENCONTRE ESA RELACION. PRESIONE <RETURN>'
87     SET CONSOLE OFF
88     WAIT
89     SET CONSOLE ON
90     RETURN
91 ENDIF
92 STORE F TO EXISTE
93 STORE REGO+NREL TO BUSCAR
94 FIND &BUSCAR
95 DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF .AND. .NOT. EXISTE
96     IF HASTA=REG1
97         REPLACE DESDE WITH ' ', RELAC WITH ' ', HASTA WITH ' '
98         STORE T TO EXISTE
99     ENDIF
100    SKIP
```

- 101 ENDDO
- 102 SELECT PRIMARY
- 103 RETURN
- 104 *

El programa de mantenimiento de la base de datos de la tesauría se divide en tres partes: la primera es el mantenimiento de la estructura de la base de datos, la segunda es el mantenimiento de los datos de la tesauría y la tercera es el mantenimiento de los índices de la tesauría.

El mantenimiento de la estructura de la base de datos se realiza mediante el uso de los comandos de mantenimiento de la base de datos de dBase III. El mantenimiento de los datos de la tesauría se realiza mediante el uso de los comandos de mantenimiento de los datos de dBase III. El mantenimiento de los índices de la tesauría se realiza mediante el uso de los comandos de mantenimiento de los índices de dBase III.

El programa de mantenimiento de la base de datos de la tesauría se divide en tres partes: la primera es el mantenimiento de la estructura de la base de datos, la segunda es el mantenimiento de los datos de la tesauría y la tercera es el mantenimiento de los índices de la tesauría.

El mantenimiento de la estructura de la base de datos se realiza mediante el uso de los comandos de mantenimiento de la base de datos de dBase III. El mantenimiento de los datos de la tesauría se realiza mediante el uso de los comandos de mantenimiento de los datos de dBase III. El mantenimiento de los índices de la tesauría se realiza mediante el uso de los comandos de mantenimiento de los índices de dBase III.

Programa TESE

Para eliminar un término y sus relaciones con otros.

Selecciona el archivo1 (línea 3), prepara la variable TEMP para una lectura de un término (línea 4), se borra la pantalla (línea 5) y se presenta un mensaje de ingreso de datos del término que se desea eliminar (líneas 6 a 8). El programa lee la respuesta del usuario (línea 9). Si no se ingresó información se retorna al menú inicial (líneas 11 a 12).

Se pasa la variable TEMP a mayúsculas (línea 13). Se busca el término en el archivo1 (línea 14). Si el término no existe se da un mensaje en la pantalla y se retorna al menú principal (líneas 15 a 21).

Si el término está en el archivo se recupera el número de registro llamando a la rutina CEROS (línea 22). Revisando si el descriptor tiene nota de alcance (línea 23), el programa selecciona el archivo4 y deja en blanco el registro que tiene la nota de alcance de este descriptor (líneas 24 a 28). Nuevamente en el archivo1 se deja en blanco el registro que tiene al descriptor que se está borrando (línea 31).

Se selecciona el archivo2 (línea 32) y se prepara la variable BUSCAR para efectuar la búsqueda de la información del término que se está borrando (línea 33).

La instrucción SET EXACT OFF (línea 34) se emplea para que en la búsqueda (línea 35) se localice cualquier registro que el campo DESDE sea igual que número de registro del término que se está eliminando, sin importar qué tipo de relación sea. Luego se retorna el tipo de búsqueda al estado normal, es decir que además de pedir un número de registro es necesario especificar el tipo de relación que se desea encontrar (línea 36).

En este momento buscan los datos de este término, principiando por sus códigos de clasificación. Para cada uno que se encuentre se dejará ese registro disponible (líneas 37 a 47).

Las siguientes instrucciones servirán para borrar las relaciones que tenga el descriptor que se está eliminando. Primero se prepara la variable BUSCAR para localizar la relación complementaria de la relación que se está borrando (líneas 50 a 61). Como se necesitará seguir buscando más relaciones se guarda el número del registro actual en la variable SAVEREG (línea 62) y se localiza la relación complementaria (línea 63). Si se diera el caso de que no se encuentra esta relación se dará aviso al usuario de que el archivo está incompleto, esto quiere decir que fue modificado sin usar el programa de manejo de tesauros, por lo cual el tesauró que se trabaja ya no es 100% confiable (líneas 64 a 68).

SELECCIONAR PARA BORRAR UN TERMINO Y SUS SINONIMOS CON UN
SELECCIONAR PARA BORRAR UN TERMINO Y SUS SINONIMOS CON UN
SELECCIONAR PARA BORRAR UN TERMINO Y SUS SINONIMOS CON UN

Si se encuentra la relación complementaria se reemplaza
ese registro con espacios en blanco para dejarlo disponible
(líneas 70 a 76). Luego se retorna al registro que había
quedado marcado (línea 77) y se reemplaza con espacios en
blanco para dejarlo disponible, ya que es una relación asignada
al descriptor que se está eliminando (líneas 78 a 86). Luego se
selecciona el archivo y se retorna al menú (líneas 89 y 90).

```

10 END
11 IF #=0
12 THEN
13   GOTO 14
14   SET CONSOLE OFF
15   WAIT
16   SET CONSOLE ON
17   RETURN
18 END
19 IF #=0
20 THEN
21   GOTO 22
22   SET CONSOLE OFF
23   WAIT
24   SET CONSOLE ON
25   RETURN
26 END
27 IF #=0
28 THEN
29   GOTO 30
30   SET CONSOLE OFF
31   WAIT
32   SET CONSOLE ON
33   RETURN
34 END
35 IF #=0
36 THEN
37   GOTO 38
38   SET CONSOLE OFF
39   WAIT
40   SET CONSOLE ON
41   RETURN
42 END
43 IF #=0
44 THEN
45   GOTO 46
46   SET CONSOLE OFF
47   WAIT
48   SET CONSOLE ON
49   RETURN
50 END
51 IF #=0
52 THEN
53   GOTO 54
54   SET CONSOLE OFF
55   WAIT
56   SET CONSOLE ON
57   RETURN
58 END
59 IF #=0
60 THEN
61   GOTO 62
62   SET CONSOLE OFF
63   WAIT
64   SET CONSOLE ON
65   RETURN
66 END
67 IF #=0
68 THEN
69   GOTO 70
70   SET CONSOLE OFF
71   WAIT
72   SET CONSOLE ON
73   RETURN
74 END
75 IF #=0
76 THEN
77   GOTO 78
78   SET CONSOLE OFF
79   WAIT
80   SET CONSOLE ON
81   RETURN
82 END
83 IF #=0
84 THEN
85   GOTO 86
86   SET CONSOLE OFF
87   WAIT
88   SET CONSOLE ON
89   RETURN
90 END

```

```

1  * TESE.PRG
2  * PROGRAMA PARA BORRAR UN TERMINO Y SUS RELACIONES CON OTROS
3  SELECT PRIMARY
4  STORE BLANCO TO TEMP
5  ERASE
6  @ 05,15 SAY '** ELIMINACION DE UN TERMINO **'
7  @ 08,10 SAY 'PARA RETORNAR AL MENU PRESIONE <RETURN>'
8  @ 10,10 SAY 'INGRESE EL TERMINO ' GET TEMP
9  READ
10 IF TEMP=' '
11     RETURN
12 ENDIF
13 STORE !(TEMP) TO TEMP
14 FIND &TEMP
15 IF #=0
16     @ 15,10 SAY 'ESTE TERMINO NO EXISTE EN EL ARCHIVO. PRESIONE <
17     SET CONSOLE OFF
18     WAIT
19     SET CONSOLE ON
20     RETURN
21 ENDIF
22 DO CEROS
23 IF NOTA='*'
24     USE &ARCHIVO4 INDEX &ARCHIVO4
25     FIND &REGO
26     REPLACE DESDE WITH ' ', LINEA1 WITH ' ', LINEA2 WITH ' ',
27         LINEA3 WITH ' ', LINEA4 WITH ' ',
28     USE &ARCHIVO1 INDEX &ARCHIVO1
29     GO &REGO
30 ENDIF
31 REPLACE DESCRIPTOR WITH ' ', NOTA WITH ' '
32 SELECT SECONDARY
33 STORE REGO+' ' TO BUSCAR
34 SET EXACT OFF
35 FIND &REGO
36 SET EXACT ON
37 DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
38     SKIP
39     IF EOF
40         REPLACE DESDE WITH ' ', RELAC WITH ' ', HASTA WITH ' '
41         LOOP
42     ENDIF
43     STORE # TO SAVEREG
44     SKIP -1
45     REPLACE DESDE WITH ' ', RELAC WITH ' ', HASTA WITH ' '
46     GO SAVEREG
47 ENDDO
48 * Vamos a eliminar todas las relaciones que tenga este descriptor
49 DO WHILE DESDE=REGO .AND. .NOT. EOF
50     DO CASE

```

1: DBASE\TESAURO\TESE.PRG
2

```
CASE RELAC='1'  
  STORE HASTA+'2' TO BUSCAR  
CASE RELAC='2'  
  STORE HASTA+'1' TO BUSCAR  
CASE RELAC='4'  
  STORE HASTA+'6' TO BUSCAR  
CASE RELAC='6'  
  STORE HASTA+'4' TO BUSCAR  
CASE RELAC='8'  
  STORE HASTA+'8' TO BUSCAR  
ENDCASE  
STORE # TO SAVEREG  
FIND &BUSCAR  
IF #=0  
  @ 20,10 SAY 'ERROR EN EL ARCHIVO. POSIBLEMENTE POR QUE SE MODIFICO  
  @ 21,10 SAY 'SIN USAR ESTE PROGRAMA. EL TESAURO YA NO ES CONFIABLE  
  
  *QUIT  
ENDIF  
STORE F TO EXISTE  
DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF .AND. .NOT EXISTE  
  IF HASTA=REGO  
    REPLACE DESDE WITH ' ', RELAC WITH ' ', HASTA WITH ' '  
    STORE T TO EXISTE  
  ENDIF  
  SKIP  
ENDDO  
GO SAVEREG  
SKIP  
IF EOF  
  REPLACE DESDE WITH ' ', RELAC WITH ' ', HASTA WITH ' '  
  LOOP  
ENDIF  
STORE # TO SAVEREG  
SKIP -1  
REPLACE DESDE WITH ' ', RELAC WITH ' ', HASTA WITH ' '  
GO SAVEREG  
DDO  
LECT PRIMARY  
TURN
```

Programa TESH

Programa para modificar el nombre de un término.

Se selecciona el archivo1 (línea 3) y se prepara la variable TEMP para leer un término (línea 4).

Se borra la pantalla (línea 5) y se presenta el mensaje para indicar que se debe ingresar el término que se desea modificar (líneas 6 a 8), el programa lee el término (línea 9). Si el usuario dejó en blanco la variable se retorna al menú inicial (líneas 10 a 12).

Se cambia la variable TEMP a mayúsculas (línea 13) y se busca en el archivo (línea 14). Si el término no está se da un mensaje en la pantalla indicando que no fue encontrado tal término y se retorna al menú inicial (líneas 15 a 21).

En la línea 22 da la instrucción CLEAR GETS, para que el usuario no pueda escribir fuera del campo especificado para ello en la pantalla.

Se prepara la variable TEMP para la lectura del término ya modificado (línea 23). Luego el programa lee el término y lo pasa a mayúsculas (líneas 26 y 27), por último regresa al menú inicial (línea 28).

Archivo: DBASE\TESAURO\TESF.PRG
Página: 1

```
* TESH.PRG
* ESTE PROGRAMA PERMITE LA MODIFICACION DE UN TERMINO
SELECT PRIMARY
STORE BLANCO TO TEMP
ERASE
@ 05,15 SAY '** MODIFICACION DE UN TERMINO **'
@ 08,10 SAY 'PARA REGRESAR AL MENU PRESIONE <RETURN>'
@ 10,10 SAY 'INGRESE EL TERMINO ' GET TEMP
READ
IF TEMP=
    RETURN
ENDIF
STORE !(TEMP) TO TEMP
FIND &TEMP
IF #=0
    @ 15,10 SAY 'ESTE TERMINO NO EXISTE EN EL ARCHIVO. PRESIONE <RETURN>'
    SET CONSOLE OFF
    WAIT
    SET CONSOLE ON
    RETURN
ENDIF
CLEAR GETS
STORE BLANCO TO TEMP
@ 14,10 SAY 'ESCRIBA EL TERMINO EN FORMA CORRECTA'
@ 16,10 SAY '->' GET TEMP
READ
REPLACE DESCRIPTOR WITH !(TEMP)
RETURN
*
```

Programa TESSG

Para coordinar el ingreso de nuevos códigos de clasificación a un descriptor que ya se encuentre en el archivo.

Se selecciona el archivo1 (línea 3), se borra la pantalla (línea 4) y se prepara la variable TEMP para la lectura de un descriptor (línea 5). Se pide el descriptor al cual se le ingresarán nuevos códigos de clasificación (líneas 6 a 8). Luego el programa lee el descriptor (línea 9). Si la variable está en blanco se retorna al menú inicial (líneas 10 a 12). Se pasa la variable TEMP a mayúsculas (línea 13) y se busca el descriptor en el archivo (línea 14). Si no existe se da aviso al usuario y se retorna al menú inicial (líneas 15 a 22). Si el descriptor existe se llama a la rutina CLASIF para que pida los códigos para la clasificación de este descriptor (línea 23). Por último se retorna al menú inicial (línea 24).

```
1 * TESG.PRG  
2 * PROGRAMA PARA COORDINAR EL INGRESO DE CODIGOS ADICIONALES A UN TERMINO  
3 SELECT PRIMARY  
4 ERASE  
5 STORE BLANCO TO TEMP  
6 @ 05,10 SAY ' ** ADICION DE CODIGOS A LA CLASIFICACION DE UN TERMINO ** '  
7 @ 08,10 SAY ' PARA VOLVER AL MENU PRESIONE <RETURN> '  
8 @ 10,10 SAY ' INGRESE EL TERMINO ' GET TEMP  
9 READ  
10 IF TEMP=""  
11 RETURN  
12 ENDIF  
13 STORE !(TEMP) TO TEMP  
14 FIND &TEMP  
15 IF #=0  
16 @ 20,10 SAY ' ESTE TERMINO NO EXISTE EN EL ARCHIVO. PRESIONE <RETURN> '  
17 @ 21,10 SAY ' (VER OPCION "A" DEL MENU) '  
18 SET CONSOLE OFF  
19 WAIT  
20 SET CONSOLE ON  
21 RETURN  
22 ENDIF  
23 DO CLASIF.  
24 RETURN  
25 *
```

Programa TESH

Para quitar un código de clasificación a un descriptor.

Se selecciona el archivo1 (línea 3), se borra la pantalla (línea 4), se prepara la variable TEMP para la lectura de un descriptor (línea 5). El programa presenta el mensaje de ingreso de datos y lee el descriptor (líneas 6 a 9).

Si el usuario dejó en blanco la variable TEMP se retorna al menú inicial (líneas 10 a 12).

Se pasa la variable TEMP a mayúsculas (línea 13) y se busca en el archivo (línea 14). Si no se encuentra en el archivo se da el mensaje en la pantalla y se retorna al menú inicial (líneas 15 a 21).

Se llama a la rutina CEROS para recuperar el número de registro del descriptor (línea 22) y se prepara la variable BUSCAR para localizar los registros que correspondan al descriptor que se trabaja (línea 23). Se selecciona el archivo2 (línea 24), ya que en él se encuentran los códigos de clasificación que han sido asignados a este descriptor.

Las líneas 25 y 56 encierran un ciclo de instrucciones que serán repetidas tantas veces como el usuario lo solicite. Cada repetición implica la eliminación de un código o el ingreso de datos equivocados.

Lo primero que se hará es presentar en la pantalla los códigos que han sido asignados al descriptor que se trabaja, para ello se prepara la variable LINEA con un asterisco ('*') (línea 26). Se localiza el primer registro con información relacionada a este descriptor (línea 27) y para los que correspondan se repite tomar el código, que está en el campo HASTA y agregarlo a la variable LINEA, separados por un signo '+' (líneas 28 a 31). Con esto se logra que la variable LINEA quede de la siguiente forma: * código1 + código2 ... y así se pueda presentar en la pantalla (línea 32).

Se prepara la variable TEMP para la lectura de un código de cuatro caracteres (línea 33). Se presenta un mensaje de ingreso de datos en la pantalla y el programa lee la respuesta del usuario (líneas 34 y 35). Si la variable fue dejada en blanco se retorna al menú inicial (líneas 36 a 38).

Se pasa la variable TEMP a mayúsculas (línea 39) y se busca en el archivo el código ingresado por el usuario (línea 40). Se revisan todos los registros de este descriptor (líneas 42 a 48) y si alguno cumple con tener el código ingresado por el usuario se reemplazan sus tres campos: DESDE, RELAC y HASTA con espacios en blanco para eliminar su información y dejar esta área de memoria disponible para otros datos (línea 44). Si el código no existe se informa al usuario y se piden nuevos datos (líneas 49 a 55).

SET CONSOLE OFF
WAIT
SET CONSOLE ON

Archivo: DBASE\TESAURO\TESH.PRG
Línea: 1

```

* TESH.PRG
* PROGRAMA PARA QUITAR UNA CLASIFICACION A UN DESCRIPTOR
SELECT PRIMARY
ERASE
STORE BLANCO TO TEMP
@ 05,10 SAY '** QUITAR UN CODIGO DE CLASIFICACION A UN TERMINO **'
@ 08,10 SAY 'PARA VOLVER AL MENU PRESIONE <RETURN>'
@ 10,10 SAY 'INGRESE EL TERMINO ' GET TEMP
READ
IF TEMP='
  RETURN
ENDIF
STORE !(TEMP) TO TEMP
FIND &TEMP
IF #=0
  @ 20,10 SAY 'ESTE TERMINO NO EXISTE EN EL ARCHIVO. PRESIONE <RETURN>'
  SET CONSOLE OFF
  WAIT
  SET CONSOLE ON
  RETURN
ENDIF
DO CEROS
STORE REGO+' ' TO BUSCAR
SELECT SECONDARY
DO WHILE T
  STORE '* ' TO LINEA
  FIND &BUSCAR
  DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
    STORE LINEA+TRIM(HASTA)+' + ' TO LINEA
    SKIP
  ENDDO
  @ 20,10 SAY 'CLASIFICACION ACTUAL '+ LINEA + BLANCO
  STORE ' ' TO TEMP
  @ 15,10 SAY 'CODIGO A ELIMINAR ' GET TEMP
  READ
  IF TEMP='
    RETURN
  ENENDIF
  STORE !(TEMP) TO TEMP
  FIND &BUSCAR
  STORE F TO EXISTE
  DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF .AND. .NOT. EXISTE
    IF HASTA=TEMP
      REPLACE DESDE WITH ' ', RELAC WITH ' ', HASTA WITH ' '
      STORE T TO EXISTE
    ENENDIF
    SKIP
  ENDDO
  IF .NOT. EXISTE
    @ 20,00 SAY 'REVISE LOS CODIGOS DE ESTE DESCRIPTOR (PRESIONE <RETURN>)'

```

```
51          SET CONSOLE OFF  
52          WAIT  
53          SET CONSOLE ON  
54  
          @ 20,00 SAY *  
55          ENDIF  
56          ENDDO  
57          *
```

Programa TESI

Se utiliza para poner una nota de alcance a un descriptor.

Selecciona el archivo1 (línea 3), borra la pantalla (línea 4), y prepara la variable TEMP para la lectura de un descriptor (línea 5). Presenta el mensaje de ingreso de datos y lee la respuesta del usuario (líneas 6 a 9). Si la variable fue dejada en blanco se retorna al menú inicial (líneas 10 a 12).

Se pasa la variable TEMP a mayúsculas y se busca el descriptor en el archivo (líneas 13 y 14). Si el descriptor no existe se informa al usuario y se retorna al menú inicial (líneas 15 a 22).

Si el descriptor ya tiene nota de alcance, es decir si el campo NOTA del registro correspondiente al descriptor actual tiene un asterisco (*), se informa al usuario de que a este descriptor ya se le asignó una nota de alcance y se retorna al menú inicial (líneas 23 a 30).

No se tiene la opción para modificar una nota de alcance, pero si el usuario desea hacerlo puede 'borrarle' la actual y 'asignarle' otra.

Para asignarle la nota de alcance al descriptor primero se le marca el campo NOTA con un asterisco (línea 31) y se llama a la rutina CEROS (línea 32).

En este momento se tiene como archivo primario el archivo1 y como secundario el archivo2. Las notas de alcance están en el archivo4, por lo que es necesario abrir el archivo4 (línea 33), con lo cual, automáticamente, se cierra el archivo1 que se encontraba como primario.

Ya en el archivo4 se busca un registro disponible para ubicar la nota de alcance, si no se encuentra uno se agrega uno en blanco (líneas 34 a 37).

Se pone el número de registro del descriptor al cual corresponde la nota de alcance en el campo DESDE (línea 38) y se presentan cuatro líneas de treinta caracteres para que el usuario escriba la nota de alcance (líneas 39 a 42), luego el programa lee los datos y abre nuevamente el archivo1 (líneas 43 y 44).

```
1 * TESI.PRG
2 * PROGRAMA PARA PONER NOTA DE ALCANCE A UN DESCRIPTOR
3 SELECT PRIMARY
4 ERASE
5 STORE BLANCO TO TEMP
6 @ 05,10 SAY '** PONER NOTA DE ALCANCE A UN TERMINO **'
7 @ 08,10 SAY 'PARA VOLVER AL MENU PRESIONE <RETURN>'
8 @ 10,10 SAY 'INGRESE EL DESCRIPTOR ' GET TEMP
9 READ
10 IF TEMP=' '
11 RETURN
12 ENDIF
13 STORE !(TEMP) TO TEMP
14 FIND &TEMP
15 IF #=0
16 @ 20,10 SAY 'ESTE DESCRIPTOR NO EXISTE. ES NECESARIO INGRESAR
17 @ 21,10 SAY 'USANDO LA OPCION "A" DEL MENU. PRESIONE <RETURN>'
18 SET CONSOLE OFF
19 WAIT
20 SET CONSOLE ON
21 RETURN
22 ENDIF
23 IF NOTA='*'
24 @ 20,10 SAY 'ESTE DESCRIPTOR YA TIENE NOTA DE ALCANCE, PARA MODIFIC
25 @ 21,10 SAY 'DEBE BORRARLA Y ASIGNARLA DE NUEVO. PRESIONE <RE
26 SET CONSOLE OFF
27 WAIT
28 SET CONSOLE ON
29 RETURN
30 ENDIF
31 REPLACE NOTA WITH '*'
32 DO CEROS
33 USE &ARCHIVO4 INDEX &ARCHIVO4
34 GO TOP
35 IF DESDE <> ' ' .OR. EOF
36 APPEND BLANK
37 ENDIF
38 REPLACE DESDE WITH REGO
39 @ 16,10 SAY 'NOTA DE ALCANCE ' GET LINEA1
40 @ 17,10 SAY ' ' GET LINEA2
41 @ 18,10 SAY ' ' GET LINEA3
42 @ 19,10 SAY ' ' GET LINEA4
43 READ
44 USE &ARCHIVO1 INDEX &ARCHIVO1
45 RETURN
46 *
```

Programa TESJ

Para quitar una nota de alcance asignada a un descriptor.

Se selecciona el archivo1 (línea 3), se borra la pantalla (línea 4) y se prepara la variable TEMP para la lectura de un descriptor (línea 5). Luego se presenta en la pantalla el mensaje de ingreso de datos (líneas 6 a 8) para que el programa lea el nombre del descriptor al que se le va a quitar la nota de alcance (línea 9).

Si se dejó en blanco la variable TEMP se retorna al menú inicial (líneas 10 a 12). Si esta variable tiene información se pasa a letras mayúsculas (línea 13).

El programa busca el descriptor en el archivo (línea 14), si éste no existe se informa al usuario y se retorna al menú inicial (líneas 15 a 21).

Si el descriptor existe se revisa el campo NOTA, el cual debe tener un asterisco como indicación de que tiene nota de alcance. Si no la tuviera se informa al usuario y se retorna al menú inicial (líneas 22 a 28).

Para quitarle la nota de alcance primero se reemplaza el asterisco que debe estar en el campo NOTA por un espacio en blanco (línea 29), luego se llama a la rutina CEROS (línea 30).

En este momento es necesario abrir el archivo4, en donde están las notas de alcance (línea 31), operación que deja como archivo primario el archivo4 y cierra el archivo1.

Se localiza el registro que tenga la nota de alcance del descriptor que se trabaja (línea 32) y se reemplazan sus cuatro líneas con espacios en blanco, para que ese registro quede borrado pero disponible para almacenar otros datos (líneas 33 y 34).

Antes de retornar al menú inicial se abre nuevamente el archivo1, para dejarlo como primario (línea 35).

```
1 * TESJ.FRG
2 * PROGRAMA PARA QUITAR LA NOTA DE ALCANCE DE UN TERMINO
3 SELECT PRIMARY
4 ERASE
5 STORE BLANCO TO TEMP
6 @ 05,10 SAY '** QUITAR NOTA DE ALCANCE A UN TERMINO **'
7 @ 08,10 SAY 'PARA VOLVER AL MENU PRESIONE <RETURN>'
8 @ 10,10 SAY 'INGRESE EL TERMINO ' GET TEMP
9 READ
10 IF TEMP=' '
11     RETURN
12 ENDIF
13 STORE !(TEMP) TO TEMP
14 FIND &TEMP
15 IF #=0
16
17 @ 20,10 SAY 'ESTE DESCRIPTOR NO EXISTE EN EL ARCHIVO. PRESIONE <R
18     SET CONSOLE OFF
19     WAIT
20     SET CONSOLE ON
21     RETURN
22 ENDIF
23 IF NOTA<>' '*
24 @ 20,10 SAY 'ESTE DESCRIPTOR NO TIENE Nota de Alcance. PRESIONE <
25     SET CONSOLE OFF
26     WAIT
27     SET CONSOLE ON
28     RETURN
29 ENDIF
30 REPLACE NOTA WITH ' '
31 DO CEROS
32 USE &ARCHIVO4 INDEX &ARCHIVO4
33 FIND &REGO
34 REPLACE DESDE WITH ' ', LINEA1 WITH ' ', LINEA2 WITH ' ',
35     LINEA3 WITH ' ', LINEA4 WITH ' '
36 USE &ARCHIVO1 INDEX &ARCHIVO1
37 RETURN
38 *
```

Programa TESK

Para reportar en pantalla el número de registros que utiliza cada uno de los archivos del tesoro en uso.

Este programa principia guardando las variables temporales que están en uso, en el archivo TEMP (línea 3). Luego cierra las bases de datos y borra todas las variables temporales que estuvieran en uso (línea 4) y recupera las variables que se grabaron en el archivo TEMP (línea 5). Esto se hace con el fin de cerrar los archivos que se trabajan, pero sin perder los datos que están en las variables de trabajo, por ejemplo el nombre del tesoro en uso.

Borra la pantalla (línea 6) y escribe un mensaje en la pantalla, indicando el significado de los datos que se van a escribir (líneas 7 a 16).

La forma en que se presentará la información de cada archivo es la siguiente: Se preparan tres variables, BUSCAR, CAMPO y LARGO que indicarán cuál es el archivo que se desea presentar en la pantalla. BUSCAR es el número del archivo = 1, 2, 3 o 4. CAMPO es el nombre del primer campo en ese archivo, el cual se revisará para ver si el registro está en blanco. LARGO es la longitud de registro del archivo que se presenta.

Cuando se completa la información de cada archivo se llama a la rutina TESK1 para que revise ese archivo y presente los datos en la pantalla.

La información es preparada de la siguiente forma:

ARCHIVO1: líneas 17 a 20.

ARCHIVO2: líneas 21 a 23.

ARCHIVO3: líneas 24 a 26.

ARCHIVO4: líneas 27 a 29.

La variable BUSCAR se prepara sólo para el ARCHIVO1, ya que en la rutina TESK1 se le incrementa el valor, por lo cual la siguiente vez que se le llame sabrá cuál es el archivo a trabajar.

Después de presentar la información el programa presenta un mensaje y espera que el usuario indique que se puede continuar (líneas 30 a 33). El programa cierra el archivo4 que fue el último en ser presentado (línea 34) y recupera los valores iniciales de las variables que fueron almacenados en el archivo TEMP al inicio de este programa (línea 35).

Se indica que el archivo2 se usará como secundario y el archivo1 como primario (líneas 36 a 39) y se retorna al menú inicial (línea 40).

archivo: DBASE\TESAURO\TESK.PRG

pagina: 1

```
1 * TESK.PRG
2 * PROGRAMA PARA REPORTAR EN PANTALLA EL NUMERO DE REGISTROS EN
3 SAVE TO TEMP
4 CLEAR
5 RESTORE FROM TEMP
6 ERASE
7 @ 0,5 SAY 'TESAURO: '+TRABAJO
8 @ 5,5 SAY ' REGISTROS LIBRES:'
9 @ 6,5 SAY ' REGISTROS EN USO:'
10 @ 8,5 SAY ' TOTAL DE REGISTROS:'
11 @ 10,5 SAY ' LONGITUD DEL REGISTRO:'
12 @ 12,5 SAY ' TOTAL DE CARACTERES:'
13 @ 14,19 SAY 'ARCHIVO1: TERMINOS'
14 @ 16,19 SAY 'ARCHIVO2: RELACIONES ENTRE TERMINOS'
15 @ 18,19 SAY 'ARCHIVO3: CODIGOS Y ENCABEZADOS DE CLASIFICACION'
16 @ 20,19 SAY 'ARCHIVO4: NOTAS DE ALCANCE'
17 STORE 1 TO BUSCAR
18 STORE 'DESCRIPTOR' TO CAMPO
19 STORE 42 TO LARGO
20 DO TESK1
21 STORE 'DESDE' TO CAMPO
22 STORE 10 TO LARGO
23 DO TESK1
24 STORE 'CODIGO' TO CAMPO
25 STORE 36 TO LARGO
26 DO TESK1
27 STORE 'DESDE' TO CAMPO
28 STORE 125 TO LARGO
29 DO TESK1
30 @ 23,20 SAY 'PRESIONE <RETURN> PARA VOLVER AL MENU'
31 SET CONSOLE OFF
32 WAIT
33 SET CONSOLE ON
34 CLEAR
35 RESTORE FROM TEMP
36 SELECT SECONDARY
37 USE &ARCHIVO2 INDEX &ARCHIVO2
38 SELECT PRIMARY
39 USE &ARCHIVO1 INDEX &ARCHIVO1
40 RETURN
41 *
```

Rutina TESK1

Escribe en la pantalla la información de un archivo del tesoro en uso.

Esta rutina prepara la variable ARCHIVO para indicar el archivo que se utiliza (línea 3).

Dependiendo del valor que tenga la variable BUSCAR se abrirá el archivo 1, 2, 3 o 4 en forma indexada, es decir con los datos en orden alfabético (líneas 4 a 13).

La variable TEMP acumulará el número de registros en blanco que utiliza el archivo que se presenta, por lo que se le da valor inicial cero (línea 14).

Cada vez que se encuentre un registro en blanco se incrementará el valor de TEMP en 1 (líneas 14 a 18).

Esta rutina coloca la información de los cuatro archivos en la pantalla, por lo que calcula la posición que le corresponde al archivo, dependiendo del valor de la variable BUSCAR, almacenándolo en la variable LINEA (línea 19).

Escribe el nombre del archivo (línea 20) y el número de registros en blanco, o sea la variable TEMP (línea 21).

Para saber el número total de registros en este archivo, esta rutina abre nuevamente el archivo pero ahora no en forma indexada, sino en forma secuencial, con el fin de que el último registro ingresado esté al final del archivo (líneas 22 a 31).

Ya con el archivo abierto en forma secuencial se da la instrucción de 'ir al final del archivo' (línea 32). El número de ese registro se obtiene con la función '#', por lo que el número de registros en uso se obtiene restandole a # el valor de TEMP (línea 33) el cual se almacena en la variable TEMP y se escribe en la pantalla (línea 34).

Se emplea nuevamente la función '#' para almacenar en la variable TEMP el número total de registros que tiene este archivo (línea 35), el que se escribe en la pantalla (línea 36).

El número de caracteres que tiene el registro de este archivo viene en la variable LARGO, por lo que el total de caracteres en el archivo se calcula multiplicando ese valor por el número de registros en el archivo (línea 37) el que se almacena en la variable TEMP. Luego se escribe en pantalla la longitud del registro (línea 38) y el total de caracteres en el archivo (línea 39).

Se le incrementa el valor a la variable BUSCAR, para que cuando se retorne a esta rutina se escriban los datos del archivo siguiente (línea 40). Se retorna al programa TESK (línea 41).

```
1 * TESK1.PRG
2 * ESCRIBE EN LA PANTALLA LOS DATOS DE UN ARCHIVO DEL TESAURO EN US
3 STORE 'ARCHIVO'+STR(BUSCAR,1) TO ARCHIVO
4 DO CASE
5 CASE BUSCAR=1
6 USE &ARCHIVO1 INDEX &ARCHIVO1
7 CASE BUSCAR=2
8 USE &ARCHIVO2 INDEX &ARCHIVO2
9 CASE BUSCAR=3
10 USE &ARCHIVO3 INDEX &ARCHIVO3
11 CASE BUSCAR=4
12 USE &ARCHIVO4 INDEX &ARCHIVO4
13 ENDCASE
14 STORE 0 TO TEMP
15 DO WHILE &CAMPO=' ' .AND. .NOT. EOF
16 STORE TEMP+1 TO TEMP
17 SKIP
18 ENDDO
19 STORE BUSCAR*10+20 TO LINEA
20 @ 3,LINEA SAY ARCHIVO
21 @ 5,LINEA SAY TEMP USING '#,###,###'
22 DO CASE
23 CASE BUSCAR=1
24 USE &ARCHIVO1
25 CASE BUSCAR=2
26 USE &ARCHIVO2
27 CASE BUSCAR=3
28 USE &ARCHIVO3
29 CASE BUSCAR=4
30 USE &ARCHIVO4
31 ENDCASE
32 GO BOTTOM
33 STORE # - TEMP TO TEMP
34 @ 6,LINEA SAY TEMP USING '#,###,###'
35 STORE # TO TEMP
36 @ 8,LINEA SAY TEMP USING '#,###,###'
37 STORE TEMP*LARGO TO TEMP
38 @ 10,LINEA SAY LARGO USING '#,###,###'
39 @ 12,LINEA SAY TEMP USING '#,###,###'
40 STORE BUSCAR+1 TO BUSCAR
41 RETURN
42 *
```

Programa TESL

Para escribir en la pantalla todos los datos de un descriptor.

Se selecciona el archivo1 (línea 3), se prepara la variable TEMP para la lectura de un término (línea 4), se prepara la variable OPCION para indicar si se desea la información ordenada (línea 5). Se borra la pantalla (línea 6). Se presenta un mensaje que pide el término a buscar, así como la opción de reordenar los datos en caso de que se les haya modificado alguna relación (líneas 7 a 9). El programa lee la información (línea 10). Si la variable TEMP fue dejada en blanco se retorna al menú inicial (líneas 11 a 13). Si no, se pasa el contenido de la variable TEMP a mayúsculas (línea 14).

El programa busca el término en el archivo (línea 15), si el término no existe se notifica al usuario y se retorna al menú inicial (líneas 16 a 22).

Si el término existe se llama a la rutina CEROS, la que retorna el número de registro del término en la variable REGO (línea 23). El programa pregunta al usuario si desea que se ordene la información de este término (líneas 24 y 25), lee la respuesta y borra la pantalla (líneas 26 y 27).

Se escribe en la pantalla el término (línea 28), luego el programa prepara la variable BUSCAR para localizar la información relacionada con este término (línea 29), si el usuario pidió la opción de reordenar la información se llama a la rutina ORDEN1 (líneas 30 a 32). Se selecciona el ARCHIVO2 (línea 33).

Esta rutina presentará, primero, los códigos de clasificación, para ello se irán agregando a la variable LINEA, la cual se prepara asignándole un espacio en blanco (línea 34).

El programa busca el primer registro que contenga un código de clasificación (línea 35).

Los códigos que se encuentren serán agregados a la variable LINEA (líneas 36 a 39). Cada código estará separado del siguiente por una diagonal ('/').

Se selecciona el archivo1 (línea 40). Se escribe en la primera línea y columna 50 el contenido de la variable LINEA, es decir los códigos de clasificación de este descriptor.

Si el descriptor tiene nota de alcance, es decir, si el campo NOTA tiene un asterisco ('*') (línea 42), se abrirá el ARCHIVO4 (línea 43), se localizará el registro en el cual está la nota correspondiente al descriptor actual (línea 44) y se escribirán los campos que dentro de la nota tengan información (líneas 47 a 54). Luego se abre el ARCHIVO1 (línea 55).

Después de escribir la nota de alcance, si la tuviera, se selecciona el archivo2 (línea 57).

El código que identifica la relación que se tiene en cada uno de los registros del archivo2 está dado por el valor de la variable RELAC.

Se buscará un registro que tenga la relación USE, cuyo código es el '1' (líneas 58 y 59). Si se encuentra esta relación se llamará a la rutina TESL1 que escribe en la pantalla la relación encontrada (líneas 60 a 62).

Se busca si existe, para este descriptor, la relación USADO POR, es decir el código '2' (líneas 63 y 64), si se encuentra se llama a la rutina TESL1 para que la escriba en la pantalla (líneas 65 a 67).

El programa busca un registro con código '4', para localizar términos MAS AMPLIOS (líneas 68 y 69). Si encuentra registros que cumplan llama a la rutina TESL1 (líneas 70 a 72).

Luego busca registros con código '6', que corresponde a términos MAS ESPECIFICOS (líneas 73 y 74), si los encuentra llama a la rutina TESL1 (líneas 75 a 77).

Por último busca registros con el código '8' de TERMINOS RELACIONADOS (líneas 78 y 79), si encuentra información llamará a la rutina TESL1 (líneas 80 a 82).

Como esta información fue escrita en la pantalla se indica al usuario que presione cualquier tecla después de que lea lo que aparece escrito (líneas 83 a 86).

Se selecciona el archivo1 y se retorna al menú inicial (líneas 87 y 88).

Archivo: DBASE\TESAURO\TESL.PRG
Línea: 1.

```
* TESL.PRG
* PROGRAMA PARA BUSCAR UN TERMINO Y ESCRIBIR SUS RELACIONES
SELECT PRIMARY
STORE BLANCO TO TEMP
STORE ' ' TO OPCION
ERASE
@ 05,10 SAY '** MOSTRAR UN TERMINO Y SUS RELACIONES CON OTROS **'
@ 08,10 SAY 'SI DESEA RETORNAR AL MENU PRESIONE <RETURN>'
@ 10,10 SAY 'INGRESE EL TERMINO ' GET TEMP
READ
IF TEMP=' '
    RETURN
ENDIF
STORE !(TEMP) TO TEMP
FIND &TEMP
IF #=0
@ 12,10 SAY 'ESTE DESCRIPTOR NO EXISTE EN EL ARCHIVO. PRESIONE <RETURN>'
    SET CONSOLE OFF
    WAIT
    SET CONSOLE ON
    RETURN
ENDIF
DO CEROS
@ 14,10 SAY 'LA INFORMACION DE ESTE DESCRIPTOR PUEDE NO ESTAR EN ORDEN'
@ 16,10 SAY 'DESEA QUE LA ORDENE PARA ASEGURARNOS (S/N) ' GET OPCION
READ
ERASE
@ 01,00 SAY DESCRIPTOR
STORE REGO+' ' TO BUSCAR
IF !(OPCION)='S'
    DO ORDEN1
ENDIF
SELECT SECONDARY
STORE ' ' TO LINEA
FIND &BUSCAR
DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
    STORE LINEA+TRIM(HASTA)+'/' TO LINEA
    SKIP
ENDDO
SELECT PRIMARY
@ 1,50 SAY LINEA
IF NOTA='*'
    USE &ARCHIVO4 INDEX &ARCHIVO4
    FIND &REGO
    STORE 1 TO I
    STORE 'NA: ' TO TEMP
    DO WHILE I < 5
        STORE 'LINEA'+STR(I,1) TO LINEA
        IF &LINEA > ' '
            ? ' ' +TEMP+&LINEA
```

```
51          ENDIF
52          STORE ' ' TO TEMP
53          STORE I+1 TO I
54      ENDDO
55      USE &ARCHIVO1 INDEX &ARCHIVO1
56  ENDIF
57  SELECT SECONDARY
58  STORE REGO+'1' TO BUSCAR
59  FIND &BUSCAR
60  IF # > 0
61      DO TESL1
62  ENDIF
63  STORE REGO+'2' TO BUSCAR
64  FIND &BUSCAR
65  IF # > 0
66      DO TESL1
67  ENDIF
68  STORE REGO+'4' TO BUSCAR
69  FIND &BUSCAR
70  IF # > 0
71      DO TESL1
72  ENDIF
73  STORE REGO+'6' TO BUSCAR
74  FIND &BUSCAR
75  IF # > 0
76      DO TESL1
77  ENDIF
78  STORE REGO+'8' TO BUSCAR
79  FIND &BUSCAR
80  IF # > 0
81      DO TESL1
82  ENDIF
83  @ 23,40 SAY 'PRESIONE <RETURN> PARA CONTINUAR'
84  SET CONSOLE OFF
85  WAIT
86  SET CONSOLE ON
87  SELECT PRIMARY
88  RETURN
89  *
```

Rutina TESL1

Sirve para escribir en la pantalla cada grupo de términos que se relacionan con el descriptor que se está presentando. Un grupo lo forman cada categoría, es decir USE, UP, MA, ME y TR.

La variable INICIO tiene el número que identifica al primer registro, en el archivo2, en que se define una relación de la categoría que se está escribiendo (línea 3). Por ejemplo el primer registro en el que se definen términos relacionados (TR) de cierto descriptor.

Si el usuario pidió la opción de reordenar los términos se llamó a la rutina ORDEN (líneas 4 a 6).

De la línea 7 a la 18 se identifica la categoría que se está trabajando. La forma de hacerlo es la siguiente:

La variable BUSCAR fue preparada uniendo un número de registro, de cuatro dígitos, con un código, de un carácter (ver líneas 58, 63, 68, 73 y 78 del programa TESL), es decir que en la quinta posición de la variable BUSCAR se tiene el código de la relación que se presenta. En el caso de que fuera un '1' se trataría de la relación 'USE', por lo que se almacena esa información en la variable TEMP (línea 9). Si se tratara de un '2' sería 'UP' (línea 11), un '4' para 'MA' (línea 13), un '6' para 'ME' (línea 15) y un '8' para 'TR' (línea 17) que son los valores permitidos por el programa.

Esta rutina localiza el primer registro de este grupo de relaciones (línea 19) y escribe la información como sigue:

Se repetirá para todos los registros que cumplan con tener información del descriptor que se escribe y que sea el mismo tipo de relación (líneas 20 a 28).

Debido a que en el Archivo2 sólo se tienen números de registro es necesario almacenar el campo HASTA, que corresponde al número de registro del término con el cual se tiene esta relación, en la variable REG1 (línea 21). Luego se selecciona el archivo1 (línea 22) y se localiza el registro en el que está el nombre del término (línea 23). Se escribe en la pantalla un margen de dos espacios y el nombre de la categoría junto con el descriptor (línea 24). Con el fin de que solo aparezca el nombre de la categoría con el primer descriptor del grupo se cambia el valor de la variable TEMP por espacios en blanco.

Ya que se escribió este término se selecciona el Archivo2 (línea 26). Se avanza al siguiente registro (línea 27). En el caso de que ya no se tengan más términos relacionados en esta categoría se retornará al programa TESL, el que está coordinando la escritura en detalle de un descriptor.

```
1 * TESL1.PRG
2
3 * PROGRAMA PARA ESCRIBIR UN GRUPO DE TERMINOS RELACIONADOS CON UN DESCRIP
4 STORE # TO INICIO
5 IF !(OPCION)='S'
6 DO ORDEN
7 ENDIF
8 DO CASE
9 CASE $(BUSCAR,5)='1'
10 STORE 'USE: ' TO TEMP
11 CASE $(BUSCAR,5)='2'
12 STORE 'UP: ' TO TEMP
13 CASE $(BUSCAR,5)='4'
14 STORE 'MA: ' TO TEMP
15 CASE $(BUSCAR,5)='6'
16 STORE 'ME: ' TO TEMP
17 CASE $(BUSCAR,5)='8'
18 STORE 'TR: ' TO TEMP
19 ENDCASE
20 GO INICIO
21 DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
22 STORE HASTA TO REG1
23 SELECT PRIMARY
24 GO &REG1
25 ? ' '+TEMP+DESCRIPTOR
26 STORE ' ' TO TEMP
27 SELECT SECONDARY
28 SKIP
29 ENDDO
30 RETURN
31 *
```

Programa TESM

Para listar los términos, o un grupo de ellos.

La instrucción SET EXACT OFF (línea 3) es empleada para indicar que las comparaciones entre variables deben ser realizadas tomando en cuenta únicamente los caracteres de la variable que tenga menos caracteres, por ejemplo: 'GAS' no es lo mismo que 'GAS METANO', pero dando esta instrucción si son tomados como iguales.

Se selecciona el archivo1 (línea 4). Se dan valores iniciales a la variable DES y HAS para incluir por default los descriptores que comiencen con la letra 'A' hasta la 'Z' (líneas 5 y 6). Además se prepara la variable TEMP para dar la opción de imprimir el listado, o solo presentarlo en la pantalla (línea 7). Se borra la pantalla (línea 8). Se presentan estas opciones en la pantalla para que el usuario pueda modificarlas (líneas 9 a 12). El programa lee la respuesta (línea 13).

La respuesta puede estar en letra mayúscula o minúscula, por lo que son pasadas a mayúsculas (líneas 14 y 15).

Para imprimir el listado (líneas 16 a 19) se utiliza el formato TESM en la instrucción de reporte con la indicación de sacarlo por la impresora (línea 18).

El formato para impresión, así como una prueba de un reporte de este programa se presentan en las hojas siguientes.

Para el reporte por pantalla (líneas 20 a 26) se utiliza el mismo formato TESM pero sin la instrucción de imprimir (línea 21). Este reporte saldrá en la pantalla pero conforme vayan saliendo los datos en la línea inferior, los demás se irán perdiendo en la parte superior, es decir que no da tiempo suficiente para leerlos. La forma para detener el listado en este caso será manual, es decir haciendo uso de la tecla de control (CTRL) combinada con la letra 'S', lo cual suspende la actividad del computador, para continuar basta con presionar cualquier tecla. Este CONTROL también puede ser aplicado al listado por impresora ya que CONTROL-S suspende la actividad del computador sin importar cuál sea.

Se cambia el modo de hacer comparaciones para que sean tomados todos los caracteres de las variables que se comparan (línea 27) y se retorna al menú inicial.

archivo: DBASE\TESAURO\TESM.PRG

pagina: 1

```
1  * TESH.PRG
2  * listado de los terminos del tesauro
3  SET EXACT OFF
4  SELECT PRIMARY
5  STORE 'A' TO DES
6  STORE 'Z' TO HAS
7  STORE ' ' TO TEMP
8  ERASE
9  @ 05,10 SAY '** LISTADO DE LOS TERMINOS **'
10 @ 10,10 SAY 'LISTAR LOS TERMINOS INCLUIDOS DESDE LA LETRA ' GET DES
11 @ 12,10 SAY ' HASTA LA LETRA ' GET HAS
12 @ 16,10 SAY 'DESEA EL LISTADO EN LA IMPRESORA (S/N) ? ' GET TEMP
13 READ
14 STORE '(DES)' TO DES
15 STORE '(HAS)' TO HAS
16 IF '(TEMP)='S'
17     SET CONSOLE OFF
18     REPORT FORM TESH TO PRINT FOR DESCRIPTOR >= DES .AND. DESCRIPTOR
19     SET CONSOLE ON
20 ELSE
21     REPORT FORM TESH FOR DESCRIPTOR >=DES .AND. DESCRIPTOR <= HAS
22     @ 23,60 SAY 'PRESIONE <RETURN>'
23     SET CONSOLE OFF
24     WAIT
25     SET CONSOLE ON
26 ENDIF
27 SET EXACT ON
28 RETURN
29 *
```


LISTADO DE LOS TERMINOS DEL TESAURO

#	DESCRIPTOR
65	ABACA
64	ADULTERACION
36	ALTA FRECUENCIA
1	ANALISIS DE DATOS
13	ANALISIS DE REDES
18	ANALISIS DE REGRESION
17	ANALISIS ESTADISTICO
2	ANALISIS MATEMATICO
23	ANALISIS POR CAMINO CRITICO
22	C.P.M.
47	COMPRESION
19	CUESTIONARIOS
63	DATOS NUMERICOS
59	DEFECTOS
30	DEFORMACION
58	DEFORMACION PLASTICA
60	DETERIORACION
48	DIFUSION
57	DILATACION
49	ENCOGIMIENTO
6	ENCUESTAS
31	ENERGIA
32	EQUILIBRIO
33	ESFUERZO
34	ESTABILIDAD
5	ESTADISTICA
7	ESTADISTICAS
50	EXPANSION
61	EXTENSOMETROS
8	EXTRAPOLACION
35	FRECUENCIA
56	FRICCION
37	IMPACTO
38	INTERFERENCIA
21	INVESTIGACION DE OPERACIONES
12	INVESTIGACION OPERATIVA
39	MAGNETISMO
24	METODO DEL CAMINO CRITICO
14	MODELOS
15	MODELOS MATEMATICOS
9	MUESTREO
51	OSCILACION
62	PANDEO
3	PARAMETROS
25	PERT
40	POLARIDAD
41	POTENCIA
28	PREDICCION
43	PRESION
10	PROBABILIDAD

LISTADO DE LOS TERMINOS DEL TESAURO

#	DESCRIPTOR
11	PRONOSTICO
29	PROYECCIONES (FUTURO)
42	PUNTO DE EQUILIBRIO
20	RECOLECCION DE DATOS
52	RESONANCIA
53	ROTACION
44	ROZAMIENTO
26	RUJA CRITICA
14	SIMULACION
54	SINCRONIZACION
27	TECNICA PARA EVALUACION DE PROYECTOS
45	TENSION
46	TORSION
4	VARIABLES
55	VIBRACION

Programa TESN

Para imprimir la sección alfabética del tesoro.

Se selecciona el archivo1 (línea 3). El programa se ubica en el primer registro de ese archivo (línea 4).

Para dar la opción de ordenar la información del tesoro se prepara la variable OPCION con un carácter (línea 5). Se borra la pantalla (línea 6) y se presenta el mensaje de que la impresora debe tener letra tamaño normal, así como la opción de ordenar la información (líneas 7 a 11).

El programa lee la respuesta (línea 12). Si el usuario dejó en blanco la variable OPCION se retorna al menú inicial (líneas 13 a 15).

El programa permite que se listen todos los términos o una parte de ellos. Para definir qué términos se van a incluir se toma como default la letra 'A' y 39 espacios en blanco en la variable IV para empezar con el primer término del archivo (línea 16). El programa borra la pantalla (línea 17) y llama a la rutina TESN2 (línea 18) que da opción al usuario de modificar este default.

Se dan las instrucciones para escribir en la impresora, con un margen izquierdo de 10 espacios y no escribir en la pantalla (líneas 19 a 21).

Para controlar las líneas en cada página se preparan las variables: PAGINA y ROW, las que indicarán la página y la línea en la que se está escribiendo (línea 22).

El programa escribe el número de página y el nombre del tesoro, sólo en la primera página se escribe el nombre del tesoro (línea 23). Deja una línea en blanco para separar la línea escrita de la demás información (línea 24).

Se repetirán las siguientes instrucciones para todos los registros del archivo (líneas 25 a 100).

Para cada registro se llama la rutina CEROS (línea 26). Se prepara la variable BUSCAR para localizar la información de este término (línea 27). Se almacena en la variable LARGO el término que se trabaja, pero quitándole los espacios en blanco que pudiera tener a la derecha (línea 28), ya que el término tiene asignados 40 caracteres, pero no necesariamente los ocupa todos. Luego se almacena en la variable LARGO1 el número de caracteres que tiene la variable LARGO (línea 29).

Si el usuario pidió que se ordenara la información se llama a la rutina ORDEN1 (líneas 30 a 32).

Se selecciona el archivo2 (línea 33). El programa busca los registros con códigos de clasificación asignados a este

término (línea 34). Si los encuentra los une en la variable LINEA, separados por una diagonal '/' (líneas 36 a 41).

El tamaño de la línea de impresión es de 50 caracteres, por lo que se almacena en la variable TEMP el número de espacios en blanco que es necesario incluir entre el término y sus códigos para que ocupen exactamente 50 caracteres (líneas 42 y 43), luego se almacena el descriptor, los espacios necesarios y los códigos que tenga asignados en la variable LINEA (línea 44).

Si no tuviera códigos asignados se almacena sólo el término en la variable LINEA (línea 46). Esto sucedería en el caso de que este término tuviera la relación USE.

En la instrucción de impresión (línea 50) aparece la secuencia: chr(27)+'G' que indica escribir en negrita, en la impresora EPSON MX-80. Para indicar que se termine de escribir negrita se escribe la secuencia chr(27)+'H', esto significa que el descriptor y su clasificación serán impresos en letra más oscura (línea 48). Luego se llama a la rutina LINEAS que cuenta las líneas impresas por página (línea 49) y se selecciona el archivo1 (línea 50).

Si el descriptor tiene nota de alcance se abre el archivo4 (línea 52) y se localiza el registro que tiene la nota de alcance de este descriptor (línea 53). La nota de alcance tiene cuatro líneas, para contarlas se prepara la variable I (línea 54) y la variable TEMP para identificar la información que se escribe como Nota de Alcance (línea 55). Luego se imprimen las líneas que tengan información (líneas 56 a 64). En la línea 63 se le cambia el valor de 'NA: ' a la variable TEMP por espacios en blanco, para que la identificación sea escrita sólo con la primera línea. Al terminar esta sección se abre nuevamente el archivo1 (línea 65).

En la línea de impresión (línea 59) se tiene la secuencia: chr(27)+'4' y luego chr(27)+'5', que son instrucciones para la impresora EPSON MX-80 de escribir en 'itálica'.

Después de escribir la nota de alcance, si la tuviera, se se prepara la variable BUSCAR con el número de registro del descriptor en uso y el código '1' para buscar en el Archivo2 si tiene la relación USE con algún término (líneas 68 a 69), si la tuviera se llama a la rutina TESN1 para que lo escriba (líneas 70 a 72).

Para revisar si tiene la relación USADO POR se prepara la variable BUSCAR con el número de registro del descriptor en uso y el código '2' (líneas 73 y 74). Si se encuentra se llama a la rutina TESN1 para que lo escriba (líneas 75 a 77).

La relación MAS AMPLIO se busca preparando la variable BUSCAR con el número de registro del descriptor y el código '4' (líneas 78 y 79). Si se encuentra se llama a la rutina TESN1 para que lo escriba (líneas 80 a 82).

Para revisar si tiene términos MAS ESPECIFICOS se prepara la variable BUSCAR con el número de registro del descriptor y el código '6' (líneas 83 y 84). Si se encuentra se llama a la rutina TESN1 para que lo escriba (líneas 85 a 87).

Por último se buscan TERMINOS RELACIONADOS, preparando la variable BUSCAR con el número de registro del término y el código '8' (líneas 88 y 89). Si se encuentran se llama a la rutina TESN1 para que los escriba (líneas 90 a 92).

Al terminar de imprimir la información de este término se selecciona el archivo y se localiza el término siguiente (que sigue a REGO) (líneas 93 a 95).

Si en este momento el contador de líneas (ROW) es mayor que el número 1 se deja una línea en blanco para separar un término del siguiente y se llama a la rutina que cuenta las LINEAS (líneas 96 a 99). Fin de la repetición (línea 100).

Al terminar de escribir todos los términos se da la instrucciones de dejar el margen en cero (valor de default), activar la pantalla y desactivar la impresora (líneas 101 a 103). Y se retorna al menú inicial (línea 104).

En las siguientes hojas se presenta un ejemplo de la salida impresa preparada por este programa.

```

1  * TESN.PRG
2  * PROGRAMA PARA IMPRIMIR LA SECCION ALFABETICA DEL TESAURO
3  SELECT PRIMARY
4  GO TOP
5  STORE ' ' TO OPCION
6  ERASE
7  @ 01,10 SAY 'LA IMPRESORA DEBE TENER LETRA NORMAL !'
8  @ 05,10 SAY '** IMPRESION DE LA SECCION ALFABETICA DEL TESAURO **'
9  @ 10,10 SAY 'LA INFORMACION PUEDE NO ESTAR EN ORDEN.'
10 @ 12,10 SAY 'DESEA QUE LA REORDENE (S/N) ? ' GET OPCION
11 @ 13,10 SAY '** PRESIONE <RETURN> PA'A VOLVER AL MENU SIN IMPRIMIR'
12 READ
13 IF OPCION=' '
14     RETURN
15 ENDIF
16 STORE 'A'+$(BLANCO,1,39) TO IV
17 ERASE
18 DO TESN2
19 SET PRINT ON
20 SET MARGIN TO 10
21 SET CONSOLE OFF
22 STORE 1 TO PAGINA, ROW
23 ? 'PAGINA'+STR(PAGINA,3)+' DEL TESAURO '+TRABAJO+' -SECCION ALFABETI
24 ?
25 DO WHILE .NOT. EOF .AND. DESCRIPTOR <= IV
26     DO CEROS
27     STORE REGO+' ' TO BUSCAR
28     STORE TRIM(DESCRIPTOR) TO LARGO
29     STORE LEN(LARGO) TO LARGO1
30     IF !(OPCION)='S'
31         DO ORDEN1
32     ENDIF
33     SELECT SECONDARY
34     FIND &BUSCAR
35     IF # > 0
36         STORE TRIM(HASTA) TO LINEA
37         SKIP
38         DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
39             STORE LINEA+'/'+'+TRIM(HASTA) TO LINEA
40             SKIP
41         ENDDO
42         STORE LEN(LINEA) TO TEMP
43         STORE 50 - LARGO1 - TEMP TO TEMP
44         STORE LARGO+$(BLANCO+BLANCO,1,TEMP) + LINEA TO LINEA
45     ELSE
46         STORE LARGO TO LINEA
47     ENDIF
48     ? CHR(27)+'G'+LINEA+CHR(27)+'H'
49     DO LINEAS
50     SELECT PRIMARY

```

archivo: DBASE\TESAURO\TESN.PRG
pagina: 2

```
51 IF NOTA='*'
52 USE &ARCHIVO4 INDEX &ARCHIVO4
53 FIND &REGO
54 STORE 1 TO I
55 STORE 'NA: ' TO TEMP
56 DO WHILE I < 5
57 STORE 'LINEA'+STR(I,1) TO LINEA
58 IF &LINEA > ' '
59 ? ' '+TEMP+CHR(27)+'4'+&LINEA+CHR(27)+'5'
60 DO LINEAS
61 ENDIF
62 STORE I+1 TO I
63 STORE ' ' TO TEMP
64 ENDDO
65 USE &ARCHIVO1 INDEX &ARCHIVO1
66 ENDIF
67 SELECT SECONDARY
68 STORE REGO+'1' TO BUSCAR
69 FIND &BUSCAR
70 IF # > 0
71 DO TESN1
72 ENDIF
73 STORE REGO+'2' TO BUSCAR
74 FIND &BUSCAR
75 IF # > 0
76 DO TESN1
77 ENDIF
78 STORE REGO+'4' TO BUSCAR
79 FIND &BUSCAR
80 IF # > 0
81 DO TESN1
82 ENDIF
83 STORE REGO+'6' TO BUSCAR
84 FIND &BUSCAR
85 IF # > 0
86 DO TESN1
87 ENDIF
88 STORE REGO+'8' TO BUSCAR
89 FIND &BUSCAR
90 IF # > 0
91 DO TESN1
92 ENDIF
93 SELECT PRIMARY
94 GO &REGO
95 SKIP
96 IF ROW > 1
97 ?
98 DO LINEAS
99 ENDIF
100 ENDDO
```

101 SET MARGIN TO 0
102 SET CONSOLE ON
103 SET PRINT OFF
104 RETURN
105 *

PAGINA 1 DEL TESAURO C: ICAITI - SECCION ALFABETICA

ABACA

ADULTERACION CAL

ALTA FRECUENCIA EDA

ANALISIS DE DATOS
UP: ANALISIS DE REGRESION EBA/EBD
ANALISIS ESTADISTICO

ANALISIS DE REDES EBK

ANALISIS DE REGRESION
USE: ANALISIS DE DATOS

ANALISIS ESTADISTICO
USE: ANALISIS DE DATOS

ANALISIS MATEMATICO EBA

ANALISIS POR CAMINO CRITICO
USE: C.P.M. EBK

C.P.M.
UP: ANALISIS POR CAMINO CRITICO
METODO DEL CAMINO CRITICO
PERT
RUTA CRITICA
TECNICA PARA EVALUACION DE PROYECTOS

COMPRESION EDC

CUESTIONARIOS
USE: ENCUESTAS

DATOS NUMERICOS
USE: ESTADISTICAS

DEFECTOS
TR: DEFORMACION EJG/JADA

DEFORMACION
ME: DEFORMACION PLASTICA EDA
TR: DEFECTOS
DETERIORACION
EXTENSOMETROS
PANDEO

DEFORMACION PLASTICA
MA: DEFORMACION EJG

DETERIORACION
TR: DEFORMACION CAL

PAGINA: 2

DIFUSION	EDC
DILATACION USE: EXPANSION	
ENCOGIMIENTO	EDC
ENCUESTAS UP: CUESTIONARIOS RECOLECCION DE DATOS	EBD
ENERGIA	EDA
EQUILIBRIO	EDA
ESFUERZO	EDA
ESTABILIDAD	EDA
ESTADISTICA NA: SE REFIERE AL CAMPO DE ESTUDIO	EBD
ESTADISTICAS NA: CUANDO EL DOCUMENTO LLEVA DA- TOS ESTADISTICOS, ETC. UP: DATOS NUMERICOS	EBD
EXPANSION UP: DILATACION	EDC
EXTENSOMETROS TR: DEFORMACION	GKG
EXTRAPOLACION	EBD
FRECUENCIA	EDA
FRICCION USE: ROZAMIENTO	
IMPACTO	EDA
INTERFERENCIA	EDA
INVESTIGACION DE OPERACIONES USE: INVESTIGACION OPERATIVA	
INVESTIGACION OPERATIVA UP: INVESTIGACION DE OPERACIONES	EBK
MAGNETISMO	EDA
METODO DEL CAMINO CRITICO USE: C.P.M.	

PAGINA: 3

MODELOS	EBK
MODELOS MATEMATICOS	EBK
MUESTREO	EBD
OSCILACION	EDC
PANDEO TR: DEFORMACION	EJG
PARAMETROS	EBA
PERT USE: C.F.M.	
POLARIDAD	EDA
POTENCIA	EDA
PREDICCION USE: PRONOSTICO	
PRESION	EDA
PROBABILIDAD	EBD
PRONOSTICO UP: PREDICCION PROYECCIONES (FUTURO)	EBD
PROYECCIONES (FUTURO) USE: PRONOSTICO	
PUNTO DE EQUILIBRIO	EDA
RECOLECCION DE DATOS USE: ENCUESTAS	
RESONANCIA	EDC
ROTACION	EDC
ROZAMIENTO UP: FRICCION	EDA
RUTA CRITICA USE: C.F.M.	
SIMULACION	EBK
SINCRONIZACION	EDC
TECNICA PARA EVALUACION DE PROYECTOS	

PAGINA: 4

USE: C.P.M.

TENSION

TORSION

VARIABLES

VIBRACION

EDA

EDA

EBA

EDC

PARAMETROS

PERT

USE: C.P.M.

FOLARIDAD

POTENCIA

PREDICION

USE: PROMOSTIC

PRESSION

PROBABILIDAD

PROMOSTIC

USE: PROMOSTIC

PROYECCIONES (TURBO)

USE: PROMOSTIC

PUNTO DE EQUILIBRIO

RECOLECCION DE DATOS

USE: ENCUESTAS

RESONANCIA

ROTACION

ROTAMIENTO

USE: FRICTION

SUTA CRITICA

USE: C.P.M.

SIMULACION

SINCRONIZACION

TECNICA PARA EVALUACION DE PROYECTOS

Rutina TESN1

Escribe en forma codificada, en la impresora, las relaciones de un término con otros.

Guarda en la variable INICIO el número del primer registro que tenga información relacionada con el término que se trabaja (línea 3). Si el usuario pidió la opción de ordenar la información se llama a la rutina ORDEN (líneas 4 a 6).

Se identifica la categoría que se está escribiendo, es decir se mira si es UP, ME etc. y dependiendo de ello se asigna a la variable TEMP el texto correspondiente para la impresión de los descriptores (líneas 7 a 18).

Se localiza el primer registro que corresponda a esta categoría (línea 19). La impresión se repetirá para todos los registros (línea 20). El campo HASTA del Archivo2 tiene el número de registro del término que se debe escribir, por lo que se almacena este número en la variable REG1 (línea 21). Se selecciona el Archivo1 (línea 22) y se localiza el registro que indica la variable REG1 (línea 23). Se escribe el nombre del término y se llama a la rutina que cuenta las líneas impresas (líneas 24 y 25). Se le cambia el valor que tenía la variable TEMP por espacios en blanco para que la identificación de la categoría sea escrita únicamente con el primer término (línea 26). Luego se selecciona nuevamente el archivo2 y se avanza al registro siguiente (líneas 27 y 28). Fin de la repetición (línea 29).

Al terminar de escribir esta categoría se retorna al programa TESN (línea 30).

```
1 * TESN1.PRG
2
* PROGRAMA PARA ESCRIBIR UN GRUPO DE TERMINOS RELACIONADOS CON UN DESCRIPTOR
3 STORE # TO INICIO
4 IF '(OPCION)='S'
5 DO ORDEN
6 ENDIF
7 DO CASE
8 CASE $(BUSCAR,5)='1'
9 STORE 'USE: ' TO TEMP
10 CASE $(BUSCAR,5)='2'
11 STORE 'UP: ' TO TEMP
12 CASE $(BUSCAR,5)='4'
13 STORE 'MA: ' TO TEMP
14 CASE $(BUSCAR,5)='6'
15 STORE 'ME: ' TO TEMP
16 CASE $(BUSCAR,5)='8'
17 STORE 'TR: ' TO TEMP
18 ENDCASE
19 GO INICIO
20 DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
21 STORE HASTA TO REG1
22 SELECT PRIMARY
23 GO &REG1
24 ? ' '+TEMP+DESCRIPTOR
25 DO LINEAS
26 STORE ' ' TO TEMP
27 SELECT SECONDARY
28 SKIP
29 ENDDO
30 RETURN
31 *
```

Programa TESN2

Pide los datos que se usan para definir el rango de términos que se imprimirán en la sección alfabética del tesoro.

La variable IV tiene una letra "A", que se tomará como default para iniciar el listado (línea 4).

El programa presenta el mensaje en la pantalla para que el usuario pueda modificar este default, suponiendo que desee principiar el listado con cierto término, o desde tal letra. El programa lee la respuesta del usuario y la pasa a mayúsculas (líneas 6 y 7).

Ya sabiendo cuál es el primer término que se desea incluir en el listado (está en la variable IV) se localiza en el archivo descartando los que alfabéticamente sean menores (líneas 8 a 10).

Para definir el último término del listado se da como default el valor 'ZZ' a la variable IV para que incluya todos los términos (línea 12). Luego lee la respuesta del usuario (línea 16).

En el caso de que el usuario desee listar 'hasta tal letra' será necesario agregar una letra 'Z' a la letra que ingrese el usuario (el programa la agrega en línea 19). Esto se debe a que si solo se dejara la letra ingresada por el usuario el reporte incluiría hasta un término que tuviera esa sola letra (si existiera tal término), ya que cualquier otro término que principie con esa letra y tenga otras sería alfabéticamente mayor, estando fuera del rango.

Por ejemplo: Si se pide listar hasta la letra "J"

el programa asume que se desean listar todos los términos que principien con 'J' por lo que se toma como último elemento del listado 'JZ'.

chivo: DBASE\TESAURO\TESN2.PRG
gina: 1

* TESN2.PRG

RUTINA PARA IDENTIFICAR LOS TERMINOS QUE SE INCLUYEN EN EL LISTADO (TESN.PRG)

```
@ 05,10 SAY 'PRIMER TERMINO A INCLUIR ' GET IV
@ 07,10 SAY '"A" = DESDE EL PRIMERO'
@ 09,10 SAY 'SI DESEA PRINCIPIAR CON OTRA LETRA O TERMINO ESCRIBALO'
READ
STORE !(IV) TO IV
DO WHILE DESCRIPTOR < IV .AND. .NOT. EOF
  SKIP
0 ENDDO
1 ERASE
2 STORE 'ZZ'+$(BLANCO,1,38) TO IV
3 @ 05,10 SAY 'ULTIMO TERMINO A INCLUIR ' GET IV
4 @ 07,10 SAY '"ZZ" = HASTA EL FINAL DEL ARCHIVO'
5 @ 09,10 SAY 'SI DESEA TERMINAR CON OTRA LETRA O TERMINO ESCRIBALO'
6 READ
7 STORE !(IV) TO IV
8 IF LEN(TRIM(IV))=1
9   STORE TRIM(IV)+'Z'+$(BLANCO,1,38) TO IV
0 ENDIF
1 RETURN
2 *
3
```

Programa TESO

Este programa muestra e imprime los códigos empleados en la sección clasificada y el encabezado al que corresponden.

La instrucción SET EXACT OFF (línea 3) indica que las comparaciones deben ser realizadas tomando en cuenta sólo los caracteres de la variable que tenga menos caracteres. Esto se hace con el propósito de poder reportar un grupo de códigos, por ejemplo: todos los que principien con KD.

Se selecciona el ARCHIVO1 (línea 4), que se encuentra como primario (el ARCHIVO2 está como secundario). Como sólo se pueden tener dos archivos en uso se abre el ARCHIVO3 (línea 5), que quedará como archivo primario en vez del ARCHIVO1. En el ARCHIVO3 están los códigos de clasificación y su significados.

Se preparan las variables DES y HAS para tener como opciones de default el escribir los códigos incluidos desde la letra 'A' hasta la combinación 'ZZZZ' (líneas 6 y 7). Se prepara la variable TEMP para la lectura de un carácter que indicará la opción de listar en pantalla o en impresora el reporte (línea 8). Luego se borra la pantalla (línea 9).

Se presentan las opciones en la pantalla (líneas 10 a 14) y el programa lee la respuesta del usuario (línea 15) y se pasan a mayúsculas. (líneas 16 y 17).

Estas opciones permiten el listado de de uno, varios o todos los códigos de clasificación, ya sea en la pantalla o en la impresora. Si el usuario no las modifica se tendrá el listado de todos los términos en la pantalla.

Si el usuario desea el listado en la impresora se desactiva la pantalla (línea 19). Usando el formato TESO se da la instrucción de imprimir el reporte (línea 20) y luego se activa nuevamente la pantalla (línea 21).

Si el listado se desea en la pantalla se usa el formato TESO sin la instrucción de imprimir (línea 23). Este listado será continuo, es decir que si se desea detener se deberá hacer manualmente empleando la tecla de control (CTRL) combinada con la letra 'S' lo que causa la suspensión del listado. Para continuar se presiona cualquier tecla.

Después de listar los códigos se da la instrucción de volver a la modalidad de realizar las comparaciones tomando en cuenta todos los caracteres de las variables que se comparan (línea 29).

Se abre el archivo1, como primario, y se retorna al menú inicial (líneas 30 y 31).

```

1  * TESO.PRG
2  * PROGRAMA PARA IMPRIMIR LOS CODIGOS DE LA SECCION CLASIFICADA
3  SET EXACT OFF
4  SELECT PRIMARY
5  USE &ARCHIVO3 INDEX &ARCHIVO3
6  STORE 'A ' TO DES
7  STORE 'ZZZZ' TO HAS
8  STORE ' ' TO TEMP
9  ERASE
10 @ 05,10 SAY '** LISTADO DE LOS CODIGOS DE LA SECCION CLASIFICADA **'
11 @ 08,10 SAY 'PREPARE LA IMPRESORA CON LETRA NORMAL !'
12 @ 10,10 SAY 'LISTAR CODIGOS INCLUIDOS DESDE: ' GET DES
13 @ 12,10 SAY ' HASTA: ' GET HAS
14 @ 16,10 SAY 'EL LISTADO EN IMPRESORA (S/N) ? ' GET TEMP
15 READ
16 STORE !(DES) TO DES
17 STORE !(HAS) TO HAS
18 IF !(TEMP)='S'
19     SET CONSOLE OFF
20     REPORT FORM TESO TO PRINT FOR CODIGO >= DES .AND. CODIGO <= HAS
21     SET CONSOLE ON
22 ELSE
23     REPORT FORM TESO FOR CODIGO >= DES .AND. CODIGO <= HAS
24     @ 23,60 SAY 'PRESIONE <RETURN>'
25     SET CONSOLE OFF
26     WAIT
27     SET CONSOLE ON
28 ENDIF
29 SET EXACT ON
30 USE &ARCHIVO1 INDEX &ARCHIVO1
31 RETURN
32 *

```

Formato TESO

Para el listado de los códigos de la sección clasificada.

Estas son instrucciones para el dBASE II que identifican un tipo de reporte para un archivo.

Línea #1: Si se desea Encabezado de Página.

Línea #2: Encabezado = 'CODIGOS UTILIZADOS EN LA SECCION CLASIFICADA'.

Línea #3: No se desea el reporte a Doble Espacio.

Línea #4: No se desean totales.

Línea #5: En la primera columna colocar el código.

Línea #6: Encabezado de la primera columna = 'COD.'

Línea #7: En la segunda columna colocar el significado.

Línea #8: Encabezado de la segunda columna = 'Encabezado'

Línea #9: En la tercera columna coloque la marca que lo identifica, además, como descriptor.

Línea #10: Encabezado de la tercera columna = 'Descriptor'

En hoja siguiente se presenta un reporte elaborado con este formato.

- 1 Y
- 2 CODIGOS UTILIZADOS EN LA SECCION CLASIFICADA
- 3 N
- 4 N
- 5 05, CODIGO
- 6 COD.
- 7 40, DETALLE
- 8 ENCABEZADO
- 9 10, ' ' +ES:0:NO
- 10 DESCRIPTOR

CODIGOS UTILIZADOS EN LA SECCION CLASIFICADA

COD.	ENCABEZADO	DESCRIPTOR
CAL	ACTIVIDADES DE CAMBIO	
E	CIENCIA	
EB	CIENCIAS MATEMATICAS	
EBA	MATEMATICA	
EBD	ESTADISTICA	*
EBK	INVESTIGACION OPERATIVA	*
ED	FISICA	
EDA	CONCEPTOS FUNDAMENTALES DE FISICA	
EDC	FORMAS DE MOVIMIENTO	
EDD	PROPIEDADES FISICAS	*
EDG	ELECTRICIDAD	
EDJ	TERMODINAMICA	*
EDM	MECANICA DE FLUIDOS	
EDR	FISICA DE RADIACIONES	
EJG	COMPORTAMIENTO MECANICO DE MATERIALES	
GKG	INSTRUMENTOS DE MEDICION	*
JADA	PLANIFICACION	*

Programa TESP

Para ingresar nuevos códigos de la sección clasificada y su significado.

En este momento se tiene como archivo primario el archivo1, y como secundario el archivo2. Para trabajar los códigos será necesario abrir el archivo3 y dejarlo como primario (líneas 3 y 4). Cualquier archivo puede tenerse como primario o como secundario, pero para tener mejor control sobre los archivos que están en uso solo se cambia el archivo primario.

Las siguientes instrucciones se repetirán hasta que el usuario deje la variable de lectura (TEMP) en blanco (líneas 5 a 34).

Se borra la pantalla (línea 6). Se prepara la variable TEMP para leer un código de cuatro caracteres como máximo (línea 7). Se escribe el mensaje en la pantalla para indicar que se ingrese un código (líneas 8 a 10). El programa lee la respuesta del usuario (línea 11). Si la variable TEMP fue dejada en blanco se abre el archivo1 y se retorna al menú inicial (líneas 12 a 15).

Si se ingresó un código se pasa a mayúsculas, ya que puede ser ingresado en minúsculas o mayúsculas (línea 16).

El programa busca ese código en el archivo (línea 17), si ya existe se escribe un mensaje en la pantalla para indicarlo (líneas 19 a 23), luego se pide otro código (línea 24).

Si el código no ha sido ingresado al archivo el programa busca un registro disponible para ubicarlo, si no lo encuentra agrega uno en blanco (líneas 26 a 29). Se llena este registro con el código ingresado por el usuario (línea 30) y se le pide al usuario que ingrese el encabezado al que corresponde ese código (línea 31). Además se pide al usuario que ingrese un asterisco para marcar el encabezado en caso de que sea también un descriptor (línea 32) y el programa lee los datos (línea 33).

hivo: DBASE\TESAURO\TESP.PRG
ina: 1

```
* TESP.PRG
* PROGRAMA PARA INGRESAR NUEVOS CODIGOS A LA SECCION CLASIFICADA
SELECT PRIMARY
USE &ARCHIVO3 INDEX &ARCHIVO3
DO WHILE T
  ERASE
  STORE ' ' TO TEMP
```

```
@ 05,10 SAY 'INGRESO DE CODIGOS Y ENCABEZADOS DE LA SECCION CLASIFICADA'
@ 08,10 SAY 'PARA VOLVER AL MENU PRESIONE <RETURN>'
@ 10,10 SAY 'INGRESE EL CODIGO ' GET TEMP
READ
IF TEMP=' '
  USE &ARCHIVO1 INDEX &ARCHIVO1
  RETURN
ENDIF
STORE !(TEMP) TO TEMP
FIND &TEMP
IF # > 0
```

```
@ 12,10 SAY 'ESTE CODIGO YA EXISTE EN EL ARCHIVO. PRESIONE RETURN
SET CONSOLE OFF
WAIT
SET CONSOLE ON
```

```
@ 12,10 SAY '
LOOP
ENDIF
GO TOP
IF CODIGO <> ' ' .OR. EOF
  APPEND BLANK
ENDIF
REPLACE CODIGO WITH TEMP
@ 12,10 SAY 'ENCABEZADO ' GET DETALLE
@ 14,10 SAY 'MARCA "*" SI ES DESCRIPTOR ' GET ES:0:NO
READ
```

ENDDO

*

Programa TESQ

Para modificar un código de clasificación o el encabezado al que corresponde.

En este momento se tiene como archivo primario el archivo1, y como secundario el archivo2. Para trabajar los códigos será necesario abrir el archivo3 y dejarlo como primario (líneas 3 y 4).

Las siguientes instrucciones se repetirán hasta que el usuario deje la variable de lectura (TEMP) en blanco (líneas 5 a 32).

Se borra la pantalla (línea 6). Se prepara la variable TEMP para leer un código de cuatro caracteres como máximo (línea 7). Se escribe el mensaje en la pantalla para indicar que se ingrese un código (líneas 8 a 10). El programa lee la respuesta del usuario (línea 11). Si la variable TEMP fue dejada en blanco se abre el archivo1 y se retorna al menú inicial (líneas 12 a 15).

Si se ingresó un código se pasa a mayúsculas, ya que puede ser ingresado en minúsculas o mayúsculas (línea 16).

El programa busca ese código en el archivo (línea 17), si no existe se escribe un mensaje en la pantalla para indicarlo (líneas 19 a 23), luego se pide otro código (línea 24).

Se pide al usuario que modifique el código o el encabezado con el fin de dejarlo grabado en forma correcta (líneas 26 a 30). El programa lee la respuesta y pasa el código a mayúsculas (líneas 31 y 32).

Archivo: DBASE\TESAURO\TESQ.PRG
Línea: 1

```
* TESQ.PRG
* PROGRAMA PARA MODIFICAR LOS CODIGOS DE CLASIFICACION
SELECT PRIMARY
USE &ARCHIVO3 INDEX &ARCHIVO3
DO WHILE T
  ERASE
  STORE ' ' TO TEMP
@ 05,10 SAY '** MODIFICAR CODIGOS O ENCABEZADOS DE LA SECCION CLASIFICADA'
@ 08,10 SAY 'PARA VOLVER AL MENU PRESIONE <RETURN>'
@ 10,10 SAY 'INGRESE EL CODIGO ' GET TEMP
READ
IF TEMP=' '
  USE &ARCHIVO1 INDEX &ARCHIVO1
  RETURN
ENDIF
STORE !(TEMP) TO TEMP
FIND &TEMP
IF # = 0
@ 12,10 SAY 'ESTE CODIGO NO EXISTE EN EL ARCHIVO. PRESIONE <RETURN>'
  SET CONSOLE OFF
  WAIT
  SET CONSOLE ON
@ 12,10 SAY '
  LOOP
ENDIF
*
@ 14,10 SAY 'MODIFIQUE EL CODIGO Y/O EL ENCABEZADO'
@ 16,10 SAY 'CODIGO ' GET CODIGO
@ 18,10 SAY 'ENCABEZADO ' GET DETALLE
@ 20,10 SAY 'MARCA "*" SI ES DESCRIPTOR ' GET ES:0:NO
READ
REPLACE CODIGO WITH !(CODIGO)
ENDDO
*
```

Programa TESR

Para borrar un código y el encabezado al que corresponde, en la sección clasificada.

Se abre el archivo3 para dejarlo como primario (líneas 3 y 4). En este archivo no está la sección clasificada sino que los códigos con los encabezados de la sección clasificada.

Se borra la pantalla (línea 5). Las siguientes instrucciones se repetirán hasta que el usuario deje la variable TEMP en blanco (líneas 6 a 27).

Se prepara la variable TEMP para la lectura de un código de cuatro caracteres como máximo (línea 7). Se escribe en la pantalla un mensaje para indicar al usuario que debe ingresar el código que desea borrar (líneas 8 a 10). El programa lee la respuesta (línea 11).

Si la variable TEMP fue dejada en blanco se abre el archivo1, como primario, y se retorna al menú inicial (líneas 12 a 15).

Si el usuario ingresó un código se pasa a mayúsculas (línea 16) y se busca en el archivo (línea 17). Si el registro no existe en el archivo se informa al usuario y se piden nuevos datos (líneas 18 a 25).

Si se encontró el registro con la información que se desea borrar se dejará ese registro disponible, reemplazando sus tres campos: CODIGO, DETALLE y ES:O:NO con espacios en blanco (línea 26).

Programa 1182

chivo: DBASE\TESAURO\TESR.PRG

gina: 1

```
* TESR.PRG
* PROGRAMA PARA BORRAR UN CODIGO DE LA SECCION CLASIFICADA.
SELECT PRIMARY
USE &ARCHIVO3 INDEX &ARCHIVO3
ERASE
DO WHILE T
  STORE ' ' TO TEMP
  @ 05,10 SAY '** BORRAR UN CODIGO DE CLASIFICACION **'
  @ 08,10 SAY 'PARA VOLVER AL MENU PRESIONE <RETURN>'
  @ 10,10 SAY 'INGRESE EL CODIGO ' GET TEMP
  READ
  IF TEMP=' '
    USE &ARCHIVO1 INDEX &ARCHIVO1
    RETURN
  ENDIF
  STORE !(TEMP) TO TEMP
  FIND &TEMP
  IF #=0
    @ 12,10 SAY 'ESTE CODIGO NO EXISTE EN EL ARCHIVO. PRESIONE <RETURN>'
    SET CONSOLE OFF
    WAIT
    SET CONSOLE ON
  @ 12,10 SAY '
  LOOP
  ENDIF
  REPLACE CODIGO WITH ' ',DETALLE WITH ' ', ES:0:NO WITH ' '
ENDDO
*
```

Programa TESS

Se utiliza para preparar los archivos para un nuevo tesoro. Cada tesoro tiene cuatro archivos de base de datos y cuatro archivos de índices.

Se borra la pantalla (línea 3) y se escribe el mensaje indicando que se va a crear un nuevo tesoro (línea 4).

Se llama a la rutina LEERFILE que lee el nombre del tesoro (línea 5). Si el usuario no ingresó información se retorna al menú inicial (líneas 6 a 8).

La variable EXISTE retorna un valor 'verdadero' (de la rutina LEERFILE) en el caso de que el tesoro ya exista, si así fuera se informará al usuario de que no puede crear otro tesoro con ese nombre (líneas 10 a 13) y se retornará al menú inicial (línea 14).

Las variables ARCHIVO1, ARCHIVO2, ARCHIVO3 y ARCHIVO4 tienen los nombres de los cuatro archivos del tesoro que está en uso.

Las variables FICHA1, FICHA2, FICHA3 y FICHA4 tienen los nombres de los cuatro archivos del nuevo tesoro.

Se almacenan las variables en uso en el archivo TEMP (línea 16). Se borran todas las variables de memoria y se cierran los archivos en uso (línea 17). Se recuperan las variables grabadas en el archivo TEMP (línea 18). Esto se hace con el fin de cerrar los archivos sin perder los valores de las variables en uso.

Los archivos que tienen la estructura base para la creación de nuevos tesoros llevan los nombres 'A', 'B', 'C' y 'D', correspondiendo a los archivos 1, 2, 3 y 4 de cada tesoro.

La operación de crear archivos para un nuevo tesoro lleva aproximadamente dos minutos, por lo que se da un mensaje en la pantalla (línea 19).

Se abre el archivo 'A' (línea 20) para copiarle la estructura al archivo1 del nuevo tesoro (línea 21). Este es el archivo de términos.

Se abre el archivo 'B' (línea 22) para copiarle la estructura al archivo2 del nuevo tesoro (línea 23). Este es el archivo de relaciones entre términos.

Se abre el archivo 'C' (línea 24) para copiarle la estructura al archivo3 del nuevo tesoro (línea 25). Este es el archivo de códigos de clasificación y sus significados.

Se abre el archivo 'D' (línea 26) para copiarle la estructura al archivo4 del nuevo tesoro (línea 26). Este es el archivo de Notas de Alcance.

En este momento solo se han creado los archivos de base de datos, pero hace falta crear los archivos de índices. Como esta operación necesita otro poco de tiempo se da un nuevo mensaje en la pantalla (línea 28).

El ARCHIVO1 se indexa por DESCRIPTOR (no importa que no tenga datos aún) (líneas 29 y 30). El ARCHIVO2 se indexa por los campos DESDE y RELAC (líneas 31 y 32). El ARCHIVO3 por código (líneas 33 y 34) y el ARCHIVO4 se indexa por el campo DESDE (líneas 35 y 36).

En este programa sólo se preparan los archivos para un nuevo tesoro, pero sigue en USO el tesoro que se había seleccionado previamente. Para cambiar de tesoro es necesario pedir la opción CAMBIAR DE TESAURO EN USO, que se presenta en el menú inicial.

Antes de retornar al menú inicial se abre nuevamente el ARCHIVO1 del tesoro en uso, y como archivo secundario el ARCHIVO2 del mismo tesoro (líneas 37 a 39).

```
1 * TESS.PRG
2 * ESTE PROGRAMA CREA ARCHIVOS PARA UN NUEVO TESAURO
3 ERASE
4 @ 05,10 SAY '** CREACION DE UN NUEVO TESAURO **'
5 DO LEERFILE
6 IF .NOT. LEIDO
7     RETURN
8 ENDIF
9 IF EXISTE
10     @ 20,10 SAY 'ESTE TESAURO YA EXISTE EN EL DISCO. PRESIONE <RETURN>'
11     SET CONSOLE OFF
12     WAIT
13     SET CONSOLE ON
14     RETURN
15 ENDIF
16 SAVE TO TEMP
17 CLEAR
18 RESTORE FROM TEMP
19 @ 20,10 SAY 'UN MOMENTITO'
20 USE A
21 COPY TO &FICHA1 STRU
22 USE B
23 COPY TO &FICHA2 STRU
24 USE C
25 COPY TO &FICHA3 STRU
26 USE D
27 COPY TO &FICHA4 STRU
28 @ 20,23 SAY 'POR FAVOR !'
29 USE &FICHA1
30 INDEX ON DESCRIPTOR TO &FICHA1
31 USE &FICHA2
32 INDEX ON DESDE+RELAC TO &FICHA2
33 USE &FICHA3
34 INDEX ON CODIGO TO &FICHA3
35 USE &FICHA4
36 INDEX ON DESDE TO &FICHA4
37 USE &ARCHIVO1 INDEX &ARCHIVO1
38 SELECT SECONDARY
39 USE &ARCHIVO2 INDEX &ARCHIVO2
40 RETURN
41 *
```

Programa TEST

Para borrar los archivos de un tesoro.

Se borra la pantalla (línea 3) y se escribe el mensaje indicando que se va a borrar un tesoro (línea 4).

Se llama a la rutina LEERFILE para leer el nombre del tesoro (línea 5). Si el usuario no ingresó información se retorna al menú inicial (líneas 6 a 8).

Si el archivo no existe se informa al usuario y se retorna al menú inicial (líneas 9 a 15). Si el tesoro en uso es el que se desea borrar se indica al usuario de que debe cambiar de tesoro en uso para poder borrarlo (líneas 16 a 23). Esto significa que no se puede borrar el tesoro que está en uso.

En este momento se pide al usuario una confirmación de que desea borrar los archivos de ese tesoro (ya que después de estas instrucciones no se podrá recuperar esta información), (línea 24 y 25). El programa lee la respuesta (línea 26), si el usuario ingresa cualquier letra que sea distinta a una 'S' (de Si), se retornará al menú inicial (líneas 27 a 29).

Luego de dar oportunidad al usuario a interrumpir el proceso se da el mensaje en pantalla informando que se borra el tesoro solicitado (línea 30).

Se dan las instrucciones para borrar (DELETE FILE) los cuatro archivos de base de datos (líneas 31 a 34).

Para cada archivo de base de datos se tiene un archivo de índices, el cual tiene el mismo nombre que el archivo de base de datos al que corresponde. Es necesario borrar también los archivos de índices, para esto el programa prepara las variables que tendrán el nombre de cada uno de los archivos a borrar, esto lo hace agregando la terminación .NDX al nombre de cada archivo para que se haga referencia a los archivos de INDICES (líneas 35 a 38) y se borran estos archivos (líneas 39 a 43).

```
1 * TEST.PRG
2 * PROGRAMA PARA BORRAR LOS ARCHIVOS DE UN TESAURO
3 ERASE
4 @ 05,10 SAY '** BORRAR UN TESAURO **'
5 DO LEERFILE
6 IF .NOT. LEIDO
7     RETURN
8 ENDIF
9 IF .NOT. EXISTE
10     @ 20,10 SAY 'ESTE TESAURO NO EXISTE. PRESIONE <RETURN>'
11     SET CONSOLE OFF
12     WAIT
13     SET CONSOLE ON
14     RETURN
15 ENDIF
16 IF !(ARCHIVO1)=!(FICHA1)
17     @ 20,10 SAY 'ESTE TESAURO ESTA EN USO. PRESIONE <RETURN> PARA CONTINUAR.
18     @ 22,10 SAY 'PARA BORRARLO DEBE DE CAMBIAR DE TESAURO EN USO'
19     SET CONSOLE OFF
20     WAIT
21     SET CONSOLE ON
22     RETURN
23 ENDIF
24 STORE ' ' TO TEMP1
25 @ 20,20 SAY 'SEGURO QUE LO DESEA BORRAR (S/N) ? ' GET TEMP1
26 READ
27 IF '(TEMP)='S'
28     RETURN
29 ENDIF
30 @ 20,20 SAY 'UN MOMENTITO. BORRANDO TESAURO: '+TEMP
31 DELETE FILE &FICHA1
32 DELETE FILE &FICHA2
33 DELETE FILE &FICHA3
34 DELETE FILE &FICHA4
35 STORE FICHA1+'.NDX' TO FICHA1
36 STORE FICHA2+'.NDX' TO FICHA2
37 STORE FICHA3+'.NDX' TO FICHA3
38 STORE FICHA4+'.NDX' TO FICHA4
39 DELETE FILE &FICHA1
40 DELETE FILE &FICHA2
41 DELETE FILE &FICHA3
42 DELETE FILE &FICHA4
43 RETURN
44 *
```

Programa TESU

Para cambiar de tesoro en uso.

Borra la pantalla (línea 3) y escribe el mensaje de que se va a cambiar de tesoro en uso (línea 4).

Llama a la rutina LEERFILE que pide el nombre del tesoro (línea 5). SI el usuario dejó la variable de lectura en blanco se retorna al menú inicial (líneas 6 a 8).

Si el tesoro no existe se informa al usuario y se retorna al menú inicial (líneas 9 a 15).

Si el tesoro existe se guarda el nombre del tesoro, junto con el nombre de la unidad de disco en el que está, en la variable TRABAJO (línea 16).

Se escribe en la pantalla el mensaje de Cambio de Tesoro (línea 17).

Las variables Ficha1, 2, 3 y 4 tienen el nombre del tesoro al cual se cambia a trabajar, por lo que se dan esos valores a las variables Archivo1, 2, 3 y 4 que son las que dicen cual es el tesoro en uso (líneas 18 a 21).

Ya con las variables teniendo los nombres de los nuevos archivos en uso se selecciona el archivo2 como secundario y el archivo1 como primario (líneas 22 a 25) y se retorna al menú inicial (línea 26).

```
1 * TESU.PRG
2 * PROGRAMA PARA CAMBIAR DE TESAURO EN USO
3 ERASE
4 @ 05,10 SAY '** CAMBIAR EL TESAURO QUE SE TRABAJA **'
5 DO LEERFILE
6 IF .NOT. LEIDO
7     RETURN
8 ENDIF
9 IF .NOT. EXISTE
10     @ 20,10 SAY 'ESTE TESAURO NO EXISTE. PRESIONE <RETURN>'
11     SET CONSOLE OFF
12     WAIT
13     SET CONSOLE ON
14     RETURN
15 ENDIF
16 STORE !(DISCO)+'':'+TEMP TO TRABAJO
17 @ 20,20 SAY 'CAMBIO A TRABAJAR EN TESAURO: '+TRABAJO
18 STORE FICHA1 TO ARCHIVO1
19 STORE FICHA2 TO ARCHIVO2
20 STORE FICHA3 TO ARCHIVO3
21 STORE FICHA4 TO ARCHIVO4
22 SELECT SECONDARY
23 USE &ARCHIVO2 INDEX &ARCHIVO2
24 SELECT PRIMARY
25 USE &ARCHIVO1 INDEX &ARCHIVO1
26 RETURN
27 *
```

Programa TESV

Este programa imprime la sección clasificada del tesauero, o una parte de ella.

Primero selecciona el archivo1, que es el que tiene los nombres de los términos (línea 3). Localiza los registros que tengan la marca '*' en el campo CUMPLE para borrarlas (líneas 4 a 8). El campo CUMPLE es utilizado para seleccionar a los descriptores que se escribirán bajo cada encabezado de la sección clasificada, después de escribir cada registro se le borra la marca que lo identificó como parte de determinado encabezado. Si durante la impresión se interrumpe el programa, intencional o accidentalmente, pueden quedar algunos descriptores con la marca '*', por lo que primero se asegura de que ningún registro quedara marcado.

Borra la pantalla (línea 9) y prepara la variable REG1 para la lectura de un código (línea 10).

Escribe en la pantalla un mensaje para informar al usuario de que tiene la opción de imprimir solo una parte de la sección clasificada, es decir desde tal código hasta tal otro. Default se tiene el código 'A' para iniciar el listado con el primer código que se encuentre en el archivo. Además se da la posibilidad de retornar al menú principal sin la obtención del listado, esto es dejando en blanco el campo de lectura (líneas 11 a 16).

Si la variable de lectura REG1 está en blanco se retorna al menú principal (líneas 17 a 19).

El programa se asegura de que el código quede en letra mayúscula (línea 20).

Los índices de la sección clasificada están en el archivo3 por lo que se abre este archivo como secundario (líneas 21 y 22).

Los códigos que alfabéticamente sean menores que el ingresado por el usuario no serán tomados en cuenta, por lo que se 'saltan' los registros con código menor (líneas 23 a 25).

Para saber cuál será el último código a listar el programa prepara la variable REG1 para la lectura (línea 26). La instrucción CLEAR GETS (línea 27) se emplea para que el usuario no pueda escribir en la pantalla fuera del campo especificado para ello.

El programa pregunta al usuario cuál desea como último código a incluir en el listado, informándole que si desea todos los códigos deberá dar el código 'ZZZZ' (líneas 28 a 30). Al leer la información el programa la pasa a letras mayúsculas (línea 31).

Para controlar el número de páginas se da valor inicial 1 a las variables PAGINA y ROW, que contarán el número de páginas y líneas respectivamente (línea 32).

Se desactiva la pantalla (línea 33) y se activa la impresora con margen izquierdo de 10 espacios (líneas 34 y 35).

En la primera página del listado se escribe el nombre del tesoro (línea 36) y se deja la siguiente línea en blanco.

Los pasos siguientes (líneas 38 a 82) se repetirán para cada uno de los encabezados que se incluirán en el listado.

El programa graba en la variable REG el número del registro que se trabaja para que se pueda regresar a este registro después de utilizar los demás archivos (línea 39).

Se escribe el encabezado (línea 40), se almacena el encabezado (que está en el campo DETALLE) en la variable BUSCAR (línea 41) y en la variable DESCR se almacena el campo ES:0:NO, que servirá para saber si el encabezado que se escribe es un descriptor del tesoro (línea 42).

Se llama a la rutina LINEAS que controla el número de líneas impresas (línea 43). El código del encabezado que se está escribiendo se almacena en la variable CODE (línea 44).

Lo que se hará a continuación es buscar en el archivo2 (línea 45) los registros que tengan como código de clasificación el código que se está escribiendo en el listado.

La variable IV se usa para contar el número de términos que se encuentren con esta clasificación (línea 46).

Los registros que se buscan son los que tengan como campo HASTA el código almacenado en la variable CODE (línea 47). Luego se almacena el campo DESDE (que indica cuál es el número de registro correspondiente al descriptor que tiene esta clasificación) en la variable DES (línea 49). Se selecciona el archivo1 y se marca el campo CUMPLE con un asterisco para saber que ese registro deberá ser listado (líneas 50 a 55).

Con las instrucciones de la línea 48 a 56 se marcan todos los descriptors que se deben escribir con el encabezado que se trabaja, pero aún no se han impreso porque se fueron encontrando por orden de registro y no en orden alfabético, luego se seleccionará el archivo1 para tener los descriptors ya en orden alfabético y sólo imprimir los que tengan un asterisco en el campo CUMPLE.

Si la variable DESCR tiene un asterisco el encabezado que se escribe es, a su vez, un descriptor, por lo que se llama a la rutina TESV2 para que localice a este descriptor (líneas 58 a 60) y lo escriba sólo como encabezado.

En el archivo1 se encuentran marcados los descriptores que se deben escribir en el grupo correspondiente al encabezado que se trabaja, por lo que se da la instrucción de ir al primer registro del archivo (línea 61).

Para contar el número de descriptores que se van escribiendo se utiliza la variable JV, a la que se le da valor inicial de 0 (línea 62).

La variable IV tiene el número de descriptores que se deben escribir en este grupo, si este número es mayor que cero se localiza el primer descriptor a imprimir (líneas 63 a 65) ya que si fuera cero indicaría que no se imprimirá ningún término.

Mientras no se complete el número de términos a imprimir se repiten las instrucciones de la línea 66 a 73.

Se incrementa en 1 el número de descriptores impresos (línea 67), se llama a la rutina CEROS que retorna el número de registro que se trabaja (línea 68), se llama a la rutina TESV1 para que imprima la información de este descriptor (línea 69). Si hacen falta descriptores por imprimir se localiza el siguiente (líneas 70 a 72).

Ya que se imprimieron los descriptores de este encabezado se abre nuevamente el archivo3 y se localiza el siguiente encabezado a trabajar (líneas 74 a 77).

Si el número de la línea de impresión es mayor que 1 se deja una línea en blanco para separar la información de un encabezado de la del siguiente (líneas 78 a 81), si fuera la línea número 1 se estaría iniciando una nueva página por lo que no habría necesidad de dejar la línea en blanco.

Si al llegar a la línea 82 del programa hacen falta encabezados por incluir, el programa retorna a la línea 39.

Antes de retornar al menú principal se selecciona como archivo secundario el archivo2, se activa la pantalla, se desactiva la impresora y se deja el margen izquierdo en 0 (líneas 83 a 87).

rchivo: DBASE\TESAURO\TESV.PRG
agina: 1

```
1 * TESV.PRG
2 * PROGRAMA PARA ESCRIBIR LA SECCION CLASIFICADA
3 SELECT PRIMARY
4 LOCATE FOR CUMPLE='*'
5 DO WHILE .NOT. EOF
6     REPLACE CUMPLE WITH ' '
7     CONT
8 ENDDO
9 ERASE
10 STORE 'A ' TO REG1
11 @ 03,10 SAY 'PREPARE LA IMPRESORA CON LETRA NORMAL'
12 @ 05,10 SAY 'IMPRESION DE LA *SECCION CLASIFICADA*'
13 @ 10,10 SAY 'LISTAR DESDE EL CODIGO ' GET REG1
14 @ 14,10 SAY '(A )=LISTAR DESDE EL PRIMER ENCABEZADO'
15
16 15,10 SAY '( )=EN BLANCO PARA RETORNAR AL MENU SIN OBTENER EL LISTADO'
17 READ
18 IF REG1=' '
19     RETURN
20 ENDIF
21 STORE !(REG1) TO REG1
22 SELECT SECONDARY
23 USE &ARCHIVO3 INDEX &ARCHIVO3
24 DO WHILE CODIGO < REG1 .AND. .NOT. EOF
25     SKIP
26 ENDDO
27 STORE 'ZZZZ' TO REG1
28 CLEAR GETS
29 @ 18,10 SAY 'LISTAR HASTA EL CODIGO ' GET REG1
30 @ 20,10 SAY '(ZZZZ)=HASTA EL ULTIMO'
31 READ
32 STORE !(REG1) TO REG1
33 STORE 1 TO PAGINA, ROW
34 SET CONSOLE OFF
35 SET PRINT ON
36 SET MARGIN TO 10
37 ? 'PAGINA'+STR(PAGINA,3)+' DEL TESAURO '+TRABAJO+' -SECCION CLASIFICADA'
38 ?
39 DO WHILE CODIGO <= REG1 AND. .NOT. EOF
40     STORE # TO REG
41     ? CHR(27)+'G'+CODIGO+' '+ES:0:NO+' '+DETALLE+CHR(27)+'H'
42     STORE DETALLE TO BUSCAR
43     STORE ES:0:NO TO DESCRIP
44     DO LINEAS
45     STORE CODIGO TO CODE
46     USE &ARCHIVO2 INDEX &ARCHIVO2
47     STORE 0 TO IV
48     LOCATE FOR HASTA=CODE
49     DO WHILE .NOT. EOF
50         STORE DESDE TO DES
51         SELECT PRIMARY
```

archivo: DBASE\TESAURO\TESV.PRG
pagina: 2

```
51      GO &DES
52      STORE IV+1 TO IV
53      REPLACE CUMPLE WITH '*'
54      SELECT SECONDARY
55      CONT
56  ENDDO
57  SELECT PRIMARY
58  IF DESCR1 = '*'
59      DO TESV2
60  ENDIF
61  GO TOP
62  STORE 0 TO JV
63  IF IV>0
64      LOCATE FOR CUMPLE='*'
65  ENDIF
66  DO WHILE JV < IV
67      STORE JV+1 TO JV
68      DO CEROS
69      DO TESV1
70      IF JV < IV
71          CONT
72      ENDIF
73  ENDDO
74  SELECT SECONDARY
75  USE &ARCHIVO3 INDEX &ARCHIVO3
76  GO REG
77  SKIP
78  IF ROW > 1
79      ?
80      DO LINEAS
81  ENDIF
82  ENDDO
83  SELECT SECONDARY
84  USE &ARCHIVO2 INDEX &ARCHIVO2
85  SET CONSOLE ON
86  SET PRINT OFF
87  SET MARGIN TO 0
88  RETURN
89  *
```

CAL ACTIVIDADES DE CAMBIO

ADULTERACION
DETERIORACION

E CIENCIA

EB CIENCIAS MATEMATICAS

EBA MATEMATICA

ANALISIS DE DATOS
 UP: ANALISIS DE REGRESION
 ANALISIS ESTADISTICO
ANALISIS MATEMATICO
PARAMETROS
VARIABLES

EBD * ESTADISTICA

NA: *SE REFIERE AL CAMPO DE ESTUDIO*

ANALISIS DE DATOS
 UP: ANALISIS DE REGRESION
 ANALISIS ESTADISTICO

ENCUESTAS

 UP: CUESTIONARIOS
 RECOLECCION DE DATOS

ESTADISTICAS

NA: *CUANDO EL DOCUMENTO LLEVA DA-
TOS ESTADISTICOS, ETC.*

 UP: DATOS NUMERICOS

EXTRAPOLACION

MUESTREO

PROBABILIDAD

PRONOSTICO

 UP: PREDICCION
 PROYECCIONES (FUTURO)

EBK * INVESTIGACION OPERATIVA

 UP: INVESTIGACION DE OPERACIONES

ANALISIS DE REDES

C.P.M.

 UP: ANALISIS POR CAMINO CRITICO
 METODO DEL CAMINO CRITICO
 PERT
 RUTA CRITICA
 TECNICA PARA EVALUACION DE PROYECTOS

MODELOS

MODELOS MATEMATICOS

SIMULACION

ED FISICA

EDA CONCEPTOS FUNDAMENTALES DE FISICA

ALTA FRECUENCIA
DEFORMACION
ENERGIA

PAGINA: 2

EQUILIBRIO
ESFUERZO
ESTABILIDAD
FRECUENCIA
IMPACTO
INTERFERENCIA
MAGNETISMO
POLARIDAD
POTENCIA
PRESION
PUNTO DE EQUILIBRIO
ROZAMIENTO
UP: FRICCION
TENSION
TORSION

EDC FORMAS DE MOVIMIENTO

COMPRESION
DIFUSION
ENCOGIMIENTO
EXPANSION
UP: DILATACION
OSCILACION
RESONANCIA
ROTACION
SINCRONIZACION
VIBRACION

EDD * PROPIEDADES FISICAS

EDG ELECTRICIDAD

EDJ * TERMODINAMICA

EDM MECANICA DE FLUIDOS

EDR FISICA DE RADIACIONES

EJG COMPORTAMIENTO MECANICO DE MATERIALES

DEFECTOS
DEFORMACION PLASTICA
PANDEO

GKG * INSTRUMENTOS DE MEDICION

EXTENSOMETROS

JADA * PLANIFICACION

DEFECTOS

Programa TESV1

Este programa escribe la información de un descriptor en la sección clasificada.

Imprime el descriptor (línea 3), llama a la rutina que cuenta las líneas (línea 4) y borra la marca del campo CUMPLE para indicar que este descriptor ya fue escrito (línea 5).

Si el campo NOTA tiene un asterisco se abre el archivo3 (que tiene las notas de alcance) para imprimir la nota de alcance de este descriptor (líneas 7).

El programa localiza el registro que tiene la nota de alcance (línea 8), da valor inicial 1 a la variable IL que se usará para contar las líneas de la nota de alcance (línea 9).

En la variable TEMP se almacena el texto que identificará la información que se escribe como una Nota de Alcance (línea 10).

En la línea 14 está la instrucción para imprimir cada una de las líneas que tenga la Nota de Alcance. Si el impresor que se esté usando no es un EPSON MX-80 puede dar problema en el momento de imprimir, si esto sucede se deberá cambiar esta línea por la siguiente: ? ' '+TEMP+&LINEA ya que las demás instrucciones sirven para el control del tipo de letra.

Después de escribir la Nota de Alcance se selecciona el archivo2 (línea 22) para buscar si este descriptor tiene la relación 'UP' (código '2') con otros términos (líneas 23 a 25). Si se encuentra esta relación se llama a la rutina que ordena la información (líneas 26 a 30) y se escriben los descriptores (líneas 31 a 40).

Ya que se escribió la información de este descriptor se selecciona nuevamente el archivo1 (la variable REGO tiene el número de registro del descriptor que se acaba de imprimir) (líneas 41 y 42).

archivo: DBASE\TESAURO\TESV1.PRG

pagina: 1

```
1 * TESV1.PRG
2 * PROGRAMA PARA ESCRIBIR UN TERMINO EN LA SECCION CLASIFICADA
3 ? ' '+DESCRIPTOR
4 DO LINEAS
5 REPLACE CUMPLE WITH ' '
6 IF NOTA='*'
7     USE &ARCHIVO4 INDEX &ARCHIVO4
8     FIND &REGO
9     STORE 1 TO IL
10    STORE ' NA: ' TO TEMP
11    DO WHILE IL < 5
12        STORE 'LINEA'+STR(IL,1) TO LINEA
13        IF &LINEA > ' '
14            ? ' '+TEMP+CHR(27)+'4'+&LINEA+CHR(27)+'5'
15        DO LINEAS
16    ENDIF
17    STORE IL+1 TO IL
18    STORE ' ' TO TEMP
19 ENDDO
20 USE &ARCHIVO1 INDEX &ARCHIVO1
21 ENDIF
22 SELECT SECONDARY
23 STORE REGO+'2' TO BUSCAR
24 FIND &BUSCAR
25 STORE ' UP: ' TO TEMP
26 IF # > 0
27     STORE # TO INICIO
28     DO ORDEN
29     GO INICIO
30 ENDIF
31 DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
32     STORE HASTA TO HAS
33     SELECT PRIMARY
34     GO &HAS
35     ? TEMP+DESCRIPTOR
36     DO LINEAS
37     STORE ' ' TO TEMP
38     SELECT SECONDARY
39     SKIP
40 ENDDO
41 SELECT PRIMARY
42 GO &REGO
43 RETURN
44 *
```

Programa TESV2

Este programa se usa para escribir la información de un encabezado cuando también es descriptor.

El campo BUSCAR tiene el nombre del encabezado, por lo que se busca en el archivo de descriptores (línea 3). Si no se encuentra se retorna al programa TESV (Líneas 4 a 6).

Si el campo CUMPLE de este descriptor está en blanco significa que no se le ha dado la clasificación, por lo que se retorna al programa TESV. Es conveniente recordar en este momento que si un descriptor es también un encabezado se debe ingresar en ambas secciones, es decir que se debe ingresar como encabezado (opción P del menú principal) y como descriptor (opciones A o B del menú) asignándole el código que le corresponde en la sección clasificada.

La variable IV, que tiene el número de descriptores que se deben imprimir, se decrementa en 1 (línea 10) ya que en este caso el término se escribe sólo como el Encabezado, con la indicación de que es descriptor.

Se llama a la rutina que retorna el número de registro del descriptor que se trabaja (línea 11) y se le borra la marca del campo CUMPLE (línea 12) para indicar que ya se imprimió.

Si el campo NOTA tiene un asterisco ('*') (línea 13) se abre el archivo4 (línea 14) y se localiza el registro que tenga la Nota de Alcance de este descriptor (línea 15).

La variable IL se usa para contar el número de líneas que tiene la Nota de Alcance (línea 16) y la variable TEMP para identificar la información que se escribe como una Nota de Alcance (línea 17).

En la línea 21 está la instrucción para imprimir cada una de las líneas de la Nota de Alcance. Si se tiene un impresor distinto al EPSON MX-80 puede dar problemas en el momento de imprimir, lo que se soluciona cambiando esta línea por la siguiente: ? TEMP+&LINEA ya que las demás instrucciones se usan para controlar el tipo de letra.

Se selecciona el archivo2 para buscar si tiene la relación 'UP' con otros términos (líneas 29 a 33).

Si se encuentra esta relación se llama a la rutina que ordena la información (línea 34 a 38) y se imprimen los descriptores que se encontraron (líneas 40 a 47).

Después de escribir la información de este encabezado, que también es descriptor, se selecciona el archivo primario y se retorna al programa TESV.

Archivo: DBASE\TESAURO\TESV2.PRG
na: 1

* TESV2.PRG

```
PROGRAMA PARA ESCRIBIR UN ENCABEZADO CUANDO ES DESCRIPTOR (SECC. ALFABETICA)
FIND &BUSCAR
IF #=0
    RETURN
ENDIF
IF CUMPLE=' '
    RETURN
ENDIF
STORE IV-1 TO IV
DO CEROS
REPLACE CUMPLE WITH ' '
IF NOTA='*'
    USE &ARCHIVO4 INDEX &ARCHIVO4
    FIND &REGO
    STORE 1 TO IL
    STORE 'NA: ' TO TEMP
    DO WHILE IL<5
        STORE 'LINEA'+STR(IL,1) TO LINEA
        IF &LINEA > ' '
            ? TEMP+CHR(27)+'4'+&LINEA+CHR(27)+'5'
            DO LINEAS
        ENDIF
        STORE IL+1 TO IL
        STORE ' ' TO TEMP
    ENDDO
    USE &ARCHIVO1 INDEX &ARCHIVO1
ENDIF
SELECT SECONDARY
USE &ARCHIVO2 INDEX &ARCHIVO2
STORE REGO+'2' TO BUSCAR
FIND &BUSCAR
STORE 'UP: ' TO TEMP
IF # > 0
    STORE # TO INICIO
    DO ORDEN
    GO INICIO
ENDIF
DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
    STORE HASTA TO HAS
    SELECT PRIMARY
    GO &HAS
    ? TEMP+DESCRIPTOR
    DO LINEAS
    STORE ' ' TO TEMP
    SELECT SECONDARY
    SKIP
ENDDO
SELECT PRIMARY
RETURN
```


Rutina LINEAS

Cuenta las líneas que van al imprimir el tesoro, para cambiar de página cada 56 líneas y escribir el número correspondiente a cada página.

Lo primero que hace es incrementar en 1 el valor de la variable que indica cuántas líneas van (línea 3), ya que a esta rutina se llama después de escribir cada línea.

El número de líneas para el que está diseñado este formato de escritura es de 56 por página, que es el tamaño normal en la impresora EPSON MX-80.

Las líneas en la hoja que se imprime están repartidas de la siguiente forma: Línea #1 para colocar el número de página, línea #2 en blanco y 54 líneas para la escritura de los datos de los descriptores. Es por ello que si el número de línea (ROW) es mayor que 54 se le da el valor 1 a la variable ROW (línea 5), se incrementa el número de página y se avanza a la página siguiente en la impresora (línea 7). Ya en la nueva página se escribe el número que le corresponde (línea 8) y se deja una línea en blanco (línea 9).

Después de controlar la numeración de las líneas se retorna al programa que llamara a esta rutina (puede ser llamada del programa TESN, o de la rutina TESN1).

```
1 * LINEAS.PRG
2 * PROGRAMA PARA CONTROLAR EL PAGINEO
3 STORE ROW+1 TO ROW
4 IF ROW>54
5     STORE 1 TO ROW
6     STORE PAGINA+1 TO PAGINA
7     EJECT
8     ? 'PAGINA: '+STR(PAGINA,3)
9     ?
10 ENDIF
11 RETURN
12 *
```

Rutina LEERFILE

Para leer el nombre de un tesoro.

La variable LEIDO indicará si la lectura fue realizada y la variable VALIDO si el nombre es válido, por eso toman valores iniciales de 'falso' (línea 3).

Se pedirá el nombre del tesoro hasta que éste sea válido o sea dejado en blanco (líneas 4 a 36).

El nombre del tesoro debe principiar con una letra y puede continuar con letras o dígitos, hasta un máximo de siete, sin dejar espacios en blanco. Cualquier caracter distinto no será permitido.

Se prepara la variable TEMP para leer el nombre del tesoro, el que deberá ser de un máximo de 7 caracteres (línea 5). Se prepara la variable DISCO para leer el nombre de la unidad de disco a la que corresponde el tesoro, de default se da el valor 'B' (línea 6). Se pregunta al usuario el nombre del tesoro (líneas 7 a 9). El programa lee la respuesta (línea 10). Si la variable TEMP fue dejada en blanco se retorna al programa que llamó a esta rutina (puede ser: TESS, TEST o TESU) (líneas 11 a 13).

Se quitan los espacios en blanco que pueda tener la variable TEMP a la derecha, en el caso de que no se usaran los 7 caracteres disponibles para el nombre (línea 14).

Se da valor 'verdadero' a la variable VALIDO.

Si el primer caracter del nombre del tesoro no es una letra el nombre no es válido, por lo que se da valor 'falso' a la variable VALIDO (líneas 16 a 18).

Se revisarán los demás caracteres, del segundo en adelante. Se da valor 2 a la variable POS, la que indicará qué posición se está revisando (línea 19). La repetición se realizará mientras la variable POS no rebase el número de caracteres que tiene el nombre del tesoro (línea 20).

Si el caracter que se verifica no es letra o dígito se da valor 'falso' a la variable VALIDO (línea 22 a 24).

Si el nombre no es válido se informa al usuario que debe ingresar otro nombre (líneas 27 a 35). Si el nombre es válido se da valor 'verdadero' a la variable LEIDO (línea 37). Si el nombre del disco fue dejado en blanco se asume que es para el disco 'B' (líneas 38 a 40).

Luego se forman los nombres de los cuatro archivos colocando antes del nombre del tesoro las letras 'A', 'B', 'C' y 'D' los cuales se asignan a las variables Ficha 1, 2, 3 y 4.

```
1 * LEERFILE.PRG
2 * PIDE EL NOMBRE DE UN TESAURO Y VERIFICA SI EXISTE
3 STORE F TO LEIDO, VALIDO
4 DO WHILE .NOT. VALIDO
5     STORE ' ' TO TEMP
6     STORE 'B' TO DISCO
7     @ 08,10 SAY 'PARA VOLVER AL MENU PRESIONE <RETURN>'
8     @ 10,10 SAY 'NOMBRE DEL TESAURO ' GET TEMP
9     @ 12,10 SAY 'EN EL DISCO ' GET DISCO
10    READ
11    IF TEMP=' '
12        RETURN
13    ENDIF
14    STORE !(TRIM(TEMP)) TO TEMP
15    STORE T TO VALIDO
16    IF $(TEMP,1,1)<'A' .OR. $(TEMP,1,1)>'Z'
17        STORE F TO VALIDO
18    ENDIF
19    STORE 2 TO POS
20    DO WHILE POS < LEN(TEMP)+1
21        STORE $(TEMP,POS,1) TO REL
22        IF (REL<'A' .OR. REL>'Z') .AND. (REL<'0' .OR. REL >'9')
23            STORE F TO VALIDO
24        ENDIF
25        STORE POS+1 TO POS
26    ENDDO
27    IF .NOT. VALIDO
28        @ 16,16 SAY 'CARACTERES NO VALIDOS EN EL NOMBRE, INGRESE OTRO.'
29        @ 17,16 SAY 'PRESIONE <RETURN> DESPUES DE LEER ESTE MENSAJE'
30        SET CONSOLE OFF
31        WAIT
32        SET CONSOLE ON
33        @ 16,16 SAY '
34        @ 17,16 SAY '
35    ENDIF
36 ENDDO
37 STORE T TO LEIDO
38 IF DISCO=' '
39     STORE 'B' TO DISCO
40 ENDIF
41 STORE DISCO+' :A'+TEMP TO FICHA1
42 STORE DISCO+' :B'+TEMP TO FICHA2
43 STORE DISCO+' :C'+TEMP TO FICHA3
44 STORE DISCO+' :D'+TEMP TO FICHA4
45 STORE FILE(FICHA1) TO EXISTE
46 RETURN
47 *
```

Rutina CEROS

Retorna el número de registro del descriptor en el que se esté ubicado.

Esta rutina utiliza la función '#' la cual tiene el valor correspondiente al número de registro actual, para retornar la variable REGO de la siguiente forma:

Es decir que en el caso de que se trate del registro número 4 la variable REGO tendrá el valor '0004'.

Siempre se retorna una variable cuyo contenido son cuatro dígitos, si es un número menor que 10 se le colocan tres ceros a la izquierda (líneas 4 y 5). Si es un número mayor que 9 y menor que 100 se le colocan dos ceros (líneas 6 y 7). Si es un número mayor que 99 y menor que 1000 se le coloca un cero a la izquierda (líneas 8 y 9). Si no se cumple cualquiera de los casos anteriores se toma como que es un número de cuatro dígitos (líneas 10 y 11).

Se retorna al programa que llamó a esta rutina (línea 12). A esta rutina la llaman la mayoría de programas del sistema de tesauros.

```
1 * CEROS.PRG
2 * PROGRAMA PARA RELLENAR DE CEROS UN STRING
3 DO CASE
4 CASE # < 10
5     STORE '000'+STR( #,1) TO REGO
6 CASE # < 100
7     STORE '00'+STR( #,2) TO REGO
8 CASE # < 1000
9     STORE '0'+STR( #,3) TO REGO
10 OTHERWISE
11     STORE STR( #,4) TO REGO
12 ENDCASE
13 RETURN
14 *
```

Rutina CLASIF

Agrega códigos de clasificación de un descriptor al archivo de claves.

Se llama a la rutina CEROS (línea 3). Se selecciona el archivo2 (línea 4).

Se da valor 'verdadero' a la variable que indicará si se continúa con esta rutina o si se retorna al programa que la llamó (línea 5).

En la variable LINEA se almacenarán los códigos que tenga asignados este término, por eso se prepara con un asterisco '*' (línea 6). Se prepara la variable BUSCAR para localizar los códigos que han sido asignados a este término (línea 7).

El programa busca los registros con códigos asignados al descriptor que se trabaja (línea 8), los que encuentre los agregará a la variable LINEA (líneas 9 a 12).

En la línea 13 empieza el grupo de instrucciones que se repiten. La instrucción CLEAR GETS se emplea para que el usuario no pueda escribir fuera de los espacios especificados para ello en la pantalla (línea 14). Luego se prepara la variable TEMP para la lectura de un código de clasificación (línea 15).

En la parte baja de la pantalla se escriben los códigos que ya tiene asignados este descriptor (línea 16) y se pide el nuevo código (línea 17). El programa lee el nuevo código (línea 18). Si el usuario dejó la variable TEMP en blanco significa que no desean agregar códigos, por lo que se retorna al programa que llamó a esta rutina (líneas 19 a 22).

Si se ingresó un código se pasa a mayúsculas (línea 23). Luego se buscan los códigos que tiene el descriptor ya asignados, para compararlos con el nuevo y que no se ingresen duplicados. Se da valor inicial 'falso' a la variable EXISTE (línea 26). Se compara el nuevo código con los que ya tiene asignados el descriptor que se trabaja (línea 28). Si se encuentra se da valor 'verdadero' a la variable EXISTE.

Si este código ya fue ingresado se informa al usuario y se piden nuevos datos (líneas 33 a 40). Si el código no ha sido ingresado se busca un registro disponible para ubicarlo, si no hay se agrega uno en blanco (líneas 42 a 45). Se llena este registro con la información proporcionada por el usuario (línea 46) y se agrega el código a la variable LINEA que se usa para mostrar en la pantalla los códigos que tiene el descriptor ya asignados (línea 47). Como esta rutina es llamada mientras otro programa tiene datos en pantalla se borra la línea 20, usada por esta rutina. Se selecciona el archivo1 (línea 50) y se retorna al programa que llamó a esta rutina (línea 51).

chivo: DBASE\TESAURO\CLASIF.PRG
gina: 1

```
* CLASIF.PRG
* ESTA RUTINA AGREGA REFERENCIAS A LA SECCION CLASIFICADA
DO CEROS
SELECT SECONDARY
STORE T TO REPETIR
STORE ' * ' TO LINEA
STORE REGO+' ' TO BUSCAR
FIND &BUSCAR
DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
  STORE LINEA + TRIM(HASTA) + ' + ' TO LINEA
  SKIP
ENDDO
DO WHILE REPETIR
  CLEAR GETS
  STORE ' ' TO TEMP
  @ 20,40 SAY LINEA
  @ 20,00 SAY 'CLASIFICACION ' GET TEMP
  READ
  IF TEMP=' '
    STORE F TO REPETIR
    LOOP
  ENDIF
  STORE !(TEMP) TO TEMP
  FIND &BUSCAR
  IF # > 0
    STORE F TO EXISTE
    DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF .AND. .NOT. EXISTE
      IF HASTA=TEMP
        STORE T TO EXISTE
      ENDIF
      SKIP
    ENDDO
    IF EXISTE
      @ 21,20 SAY 'YA TIENE ESTA CLASIFICACION. PRESIONE <RETURN>'
      SET CONSOLE OFF
      WAIT
      SET CONSOLE ON
      @ 21,20 SAY '
    LOOP
  ENDIF
ENDIF
GO TOP
IF DESDE <> ' ' .OR. EOF
  APPEND BLANK
ENDIF
REPLACE DESDE WITH REGO, HASTA WITH TEMP
STORE LINEA + TRIM(TEMP) + ' + ' TO LINEA
ENDDO
@ 20,00 SAY BLANCO+BLANCO
SELECT PRIMARY
```

51 RETURN
52 *

Esta rutina ordena las claves correspondientes a un descriptor, en el archivo2, que indican los descriptors que tienen una relación dada con éste.

La variable CUENTA indicará cuantos registros se van a ordenar, por eso se le da un valor inicial de cero (línea 3).

Cada vez que se encuentre un registro de este grupo se incrementa la variable CUENTA (líneas 4 a 7).

Se da valor cero a la variable I (línea 8) y se repetirá mientras sea menor que CUENTA - 1. Esto quiere decir que se tendrán CUENTA - 1 ciclos y en cada uno se encontrará la posición que le corresponde a un registro dentro del grupo, al final el último registro también estará ordenado.

Cada registro 'I' se comparará con los siguientes, por eso se da valor inicial I+1 a la variable J, que será el control de cada ciclo de comparaciones.

Se localiza el primer registro del grupo (línea 12), si la variable I tiene valor mayor que 1, es decir si no es el primer ciclo, se avanzan tantos registros como lo indique la variable I (líneas 13 a 15). Se guarda el código que está en ese registro en la variable IR1 (línea 16). Ahora se localizará el registro con el cual se va a comparar (líneas 17 y 18).

Ya que se sabe cuáles son los registros que se van a comparar se selecciona el archivo1 (línea 20). Se localiza el primer descriptor (línea 21) y se guarda en la variable HA1 (línea 22). Luego se localiza el segundo descriptor (línea 23) y se almacena en la variable HA2 (línea 24). Se selecciona el archivo2 (línea 25). Si la variable HA1 es mayor que la variable HA2 significa que las claves en el archivo2 deben ser cambiadas, por lo que se graba en el registro 'I' el valor que estaba en el registro 'J' y viceversa (líneas 27 a 32).

Luego se incrementa el valor de la variable J (línea 34), al finalizar la comparación de cada registro con los demás se incrementa el valor de la variable I (línea 36) hasta que se llegue al valor CUENTA - 1.

Al terminar el ordenamiento se retorna al programa de reporte que llamó a esta rutina (línea 38).

Archivo: DBASE\TESAURO\ORDEN.PRG
na: 1

```
* ORDEN.PRG
* ESTE PROGRAMA ORDENA UN GRUPO DE REGISTROS EN EL ARCHIVO DE RELACIONES
STORE 0 TO CUENTA
DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
  STORE CUENTA+1 TO CUENTA
  SKIP
ENDDO
STORE 0 TO I
DO WHILE I < CUENTA -1
  STORE I+1 TO J
  DO WHILE J < CUENTA
    GO INICIO
    IF I > 0
      SKIP I
    ENDIF
    STORE HASTA TO IR1
    GO INICIO
    SKIP J
    STORE HASTA TO IR2
    SELECT PRIMARY
    GO &IR1
    STORE DESCRIPTOR TO HA1
    GO &IR2
    STORE DESCRIPTOR TO HA2
    SELECT SECONDARY
    IF HA1 > HA2
      REPLACE HASTA WITH IR1
      GO INICIO
      IF I > 0
        SKIP I
      ENDIF
      REPLACE HASTA WITH IR2
    ENDIF
    STORE J+1 TO J
  ENDDO
  STORE I+1 TO I
ENDDO
RETURN
*
```

Rutina ORDEN1

Se utiliza para ordenar los códigos de clasificación de un descriptor, los cuales están en el archivo2.

Se selecciona el archivo2 (línea 3). Se buscan los registros que tengan los códigos de clasificación del descriptor que se trabaja (línea 5). Si no hay ninguno se retorna al programa que llamó a esta rutina (líneas 6 a 8).

Se almacena el número del primer registro en la variable INICIO (línea 9). Se da valor inicial a la variable CUENTA que indicará cuántos registros se van a ordenar (línea 10). Para cada registro que se encuentre se incrementará el valor de la variable CUENTA (líneas 11 a 14).

Se repetirán CUENTA - 1 ciclos de comparaciones entre los registros, la variable que controlará esta repetición es la I, que recibe el valor inicial de cero (línea 15).

Cada registro se comparará con los siguientes, de modo que en cada uno de los ciclos de comparación dejará ubicado a un registro, al final el último registro también quedará ordenado.

La variable que controlará cada ciclo de comparaciones es la J, a la que se le da valor inicial I+1 (línea 17).

Se localiza el primer registro del grupo que se ordena (línea 19). Si la variable I tiene valor mayor que cero, es decir, si no es el primer ciclo, se avanzan tantos registros como lo indique la variable I (líneas 20 a 22). El código está ubicado en el campo HASTA de cada registro, por lo que se almacena en la variable HA1 (línea 23). Luego se localiza el registro con el cual se va a comparar (líneas 24 y 25). Si la variable HA1 es mayor, alfabéticamente, que el campo HASTA del registro localizado significa que los registros deben ser intercambiados (líneas 27 a 33). Se incrementa el valor a la variable J (línea 35) y al terminar cada ciclo de comparaciones se incrementa el valor de la variable I (línea 37).

Al terminar el ordenamiento se selecciona el archivo1 y se retorna al programa que llamó a esta rutina (líneas 39 y 40).

archivo: DBASE\TESAURO\ORDEN1.PRG

pagina: 1

```
1 * ORDEN1.PRG
2 * ESTE PROGRAMA ORDENA LA CLASIFICACION DE UN DESCRIPTOR
3 *
4 SELECT SECONDARY
5 FIND &BUSCAR
6 IF #=0
7     RETURN
8 ENDIF
9 STORE # TO INICIO
10 STORE 0 TO CUENTA
11 DO WHILE DESDE+RELAC=BUSCAR .AND. .NOT. EOF
12     STORE CUENTA+1 TO CUENTA
13     SKIP
14 ENDDO
15 STORE 0 TO I
16 DO WHILE I < CUENTA - 1
17     STORE I+1 TO J
18     DO WHILE J < CUENTA
19         GO INICIO
20         IF I > 0
21             SKIP I
22         ENDIF
23         STORE HASTA TO HA1
24         GO INICIO
25         SKIP J
26         IF HA1 > HASTA
27             STORE HASTA TO HA2
28             REPLACE HASTA WITH HA1
29             GO INICIO
30             IF I > 0
31                 SKIP I
32             ENDIF
33             REPLACE HASTA WITH HA2
34         ENDIF
35         STORE J+1 TO J
36     ENDDO
37     STORE I+1 TO I
38 ENDDO
39 SELECT PRIMARY
40 RETURN
41 *
```


BIBLIOGRAFIA

Cómo Organizar su Información Técnica.
1984 México, INFOTEC

Diccionario de Términos Científicos y Técnicos.
1981 Edición en Castellano, Barcelona, MARCOMBO, 5V.

Diccionario/Glosario de Proceso de Datos. 2a. ed.
1975 Madrid, IBM

Guidelines for the Establishment and Development of
Monolingual Thesauri.
1973 París, UNESCO, 37 pp.

Tesaurus de Información Industrial: Centroamérica y
República Dominicana.
1971 Guatemala, ICAITI

