

68  
271



**UNIVERSIDAD NACIONAL  
AUTONOMA DE MEXICO**

**FACULTAD DE INGENIERIA**

**SISTEMA PARA LA RECUPERACION DE  
INFORMACION TEXTUAL  
ALMACENADA EN CD-ROM**

**T E S I S**  
**QUE PARA OBTENER EL TITULO DE:**  
**INGENIERO EN COMPUTACION**  
**P R E S E N T A N:**  
**DAVID DEL MORAL CORDOBA**  
**BERTHA CAMPOS MARROQUIN**

**ASESOR: FIS. RAYMUNDO HUGO RANGEL G.**

MEXICO, D. F.

1997

DICIEMBRE

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

# ÍNDICE

---

	Pág.
<b>I.- INTRODUCCIÓN</b> .....	2
<b>I.1- Justificación</b> .....	3
<b>I.2- Antecedentes</b> .....	4
<b>II.- FUNDAMENTOS</b> .....	8
<b>II.1- La tecnología del disco compacto</b> .....	9
<b>II.2- Consideraciones físicas y lógicas del CD-ROM</b> .....	34
<b>II.3- Archivos Invertidos</b> .....	40
<b>III.- ANÁLISIS DEL SISTEMA</b> .....	42
<b>IV.- DISEÑO DEL SISTEMA</b> .....	55
<b>V.- CONCLUSIONES</b> .....	80
<b>APÉNDICE A</b> .....	93
Manual de usuario .....	94
<b>APÉNDICE B</b> .....	107
PC Multimedia .....	108
<b>APÉNDICE C</b> .....	109
Glosario .....	110
<b>APÉNDICE D</b> .....	120
Programas .....	121
<b>BIBLIOGRAFÍA</b> .....	167

**OBJETIVO DEL TRABAJO DE TESIS**

**DESARROLLO DE UN SISTEMA PARA LA RECUPERACIÓN DE  
INFORMACIÓN  
TEXTUAL ALMACENADA EN CD-ROM**

## **CAPÍTULO I**

### **INTRODUCCIÓN**

## L1 JUSTIFICACIÓN

---

En el caso concreto de este trabajo de tesis, se plantea la necesidad de utilizar un sistema recuperador de información aplicado a un directorio nacional de empresas comerciales almacenado en CD-ROM. Este producto, requiere de utilerías para la rápida localización de cualquier tipo de dato de la manera más concreta posible. Asimismo, es necesario contar con módulos que permitan, a partir de una búsqueda, el generar productos como cartas, reportes personalizados y etiquetas para correspondencia.

Desafortunadamente, los sistemas recuperadores de información disponibles actualmente en el mercado, son en su mayoría completamente cerrados, de tal manera que no es posible desarrollar una programación adecuada a esta aplicación.

Por otra parte, la licencia de uso de estos sistemas resulta muy elevada al utilizarlo en edición electrónica. Por citar un ejemplo el software ROMWARE de la empresa NIMBUS Information Systems, tiene un costo de \$ 40,000.00 U.S. para utilizarlo con licencia ilimitada. TEXTWARE LITE, de la compañía del mismo nombre, tiene un precio de \$99.00 U.S. por cada pieza de CD-ROM publicada. El tiraje estimado en su primera edición, para un directorio nacional de empresas comerciales almacenado en CD-ROM (llamado DIRFAX), se calcula en 5,000 piezas, lo que significaría un desembolso muy alto si se utilizara cualquiera de los sistemas mencionados anteriormente, y no se lograría satisfacer al 100% las necesidades de recuperación solicitadas para este producto.

Finalmente, dada la experiencia y conocimientos que tenemos los autores, en el campo del manejo de información, hemos desarrollado un sistema con las características y potencialidad necesaria para hacer de la base de datos DIRFAX un producto eficiente y competitivo.

## **1.2 ANTECEDENTES**

---

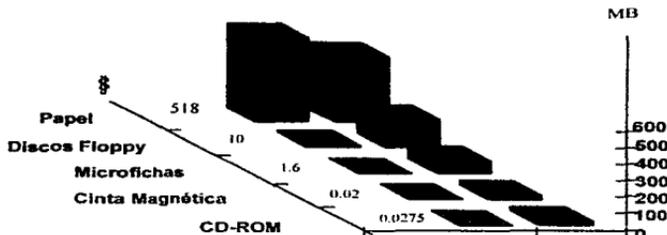
Debido a la gran cantidad de información generada en nuestros días, ha surgido la necesidad de contar con instrumentos que permitan un mejor manejo de este recurso. Las bases de datos en medios electrónicos y la cada vez mayor capacidad de almacenamiento de los dispositivos magnéticos han resuelto en parte esta necesidad. No obstante, el hablar de distribuir algunas decenas de megabytes, puede resultar una tarea muy tediosa debido a que los programas y/o datos son tradicionalmente transportados en medios de baja capacidad, como el disco flexible. Desde hace algunos años, la introducción de dispositivos periféricos de tecnología óptica y magneto-óptica, permiten almacenar cientos o miles de megabytes en un solo disco.

Como ha sucedido con otros productos, la definición, penetración e impacto de las tecnologías de almacenamiento masivo, ha sido determinada principalmente por el factor económico. Por citar un ejemplo, actualmente una unidad de disco magneto-óptica reescribible (MO), tiene un precio del orden de los \$3,000.00 U.S. mientras que una unidad de CD-ROM (Disco Compacto-Memoria de solo lectura) tiene un precio de \$ 100.00 U.S. por lo que este último se ha convertido en un dispositivo tan popular e importante como la unidad de disco flexible de cualquier equipo. Según estadísticas, se estima que hoy en día, el 80 % de los fabricantes de computadoras en Estados Unidos, Canadá y un porcentaje similar en México integran la unidad de CD-ROM como un componente estándar, en la figura 1 se muestra como se han incrementado las instalaciones de esta unidad. Asimismo, se ha proyectado que en 1995 serán vendidas más de 20 millones de unidades, entre partes para OEM y accesorios de multimedia para computadoras (CD-ROM con o sin tarjeta de sonido y bocinas). Por otro lado, varias compañías actualmente suministran sus sistemas operativos, utilerías, y toda la documentación de sus equipos en CD-ROM.

## I.2 ANTECEDENTES

SISTEMA	1991 instalados	1992 instalados	1993 instalados	1994 (ventas)	1994 instalados	1995 (ventas)
MS-DOS PC	1,250,000	2,000,000	2,150,000	(500,000) *	1,650,000	
MPC (o equiv)		800,000	3,800,000	9,500,000 *	13,300,000	15,600,000 *
Macintosh	200,000	400,000	1,525,000	1,700,000 *	3,225,000	2,800,000 *
UNIX	50,000	100,000	200,000	400,000	800,000	400,000
BASADOS EN HOST	1,500,000	3,300,000	7,675,000	11,600,000	19,275,000 *	18,000,000 *
TOTAL	INSTALADOS DE 1995 HASTA HOY =			37,275,000		

*Figura 1: ESTADÍSTICAS OPA - Instalaciones de CD-ROM por plataforma - E.U. y Canadá.  
Fuente: Executive Summary Report - Optical Publishing Association - Summer 1995*



*Figura 2: Costa en dolares por cada 100 MB de informacion almacenada en diferentes medios. (Información de 1993).*

## I.2 ANTECEDENTES

Por otra parte, el CD-ROM como medio de almacenamiento actualmente es el más económico, tanto que sin considerar el precio de los moldes y edición, la reproducción de una pieza cuesta alrededor de \$ 1.80 U.S., incluyendo empaque y etiquetado, lo que representa \$ 0.000275 U.S. por megabyte (fuente: Sonopress México 1995). El CD-ROM es tan accesible que hoy en día se ha convertido en un instrumento de uso cotidiano, según datos de la ISBN actualmente existen más de 10,200 títulos con registro oficial en el mundo, representando en 1994 un negocio de más \$ 3 billones de dólares por ventas desde los Estados Unidos.

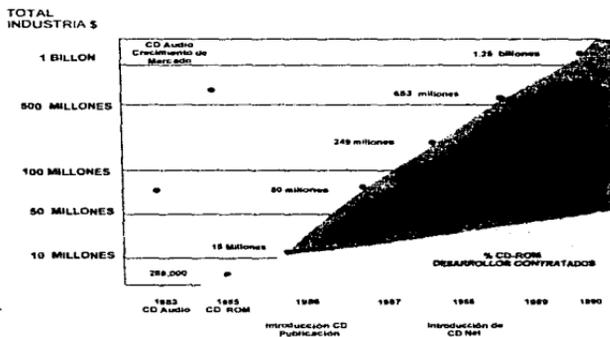


Figura 3: Crecimiento de la industria del CD-ROM

## I.2 ANTECEDENTES

Otro factor importante que motiva a impulsar y aprovechar las tecnologías de almacenamiento masivo es el ecológico, debido a que un sólo CD-ROM puede contener el equivalente a 250,000 páginas de texto. Algunas publicaciones como catálogos, directorios, diccionarios y en general, cualquier documento voluminoso que tradicionalmente se publicaba en papel ha emigrado a CD-ROM o solamente son publicadas en este medio, con lo que se consigue evitar la deforestación.

El manejo de grandes cantidades de datos requiere de programas especializados para su utilización, para explicar esto, supóngase por ejemplo, el caso de la búsqueda de una persona en un directorio telefónico en papel: ciertos apellidos comunes como "MARTÍNEZ" abarcan varias páginas del documento, a pesar de contar con ciertas facilidades de localización, es necesario recorrer un índice que únicamente se encuentra alfabetizado por apellido o nombre. Si esto resulta problemático debido al gran número de aciertos (Personas con apellido "MARTINEZ"), el operar sobre una cantidad mayor de datos podría hacer impráctica esta acción. Es por esta razón que en estos casos es necesario contar con la mayor cantidad de puntos de acceso posibles (índices) y a la vez con herramientas que permitan la selección exacta de los datos.

Para aplicaciones en donde se opera con información de tipo textual, los sistemas comercialmente disponibles generalmente utilizan la técnica de archivos invertidos, que consiste en crear índices a partir de la información contenida en un registro, cada entrada al índice lleva consigo todos los datos que determinan la dirección de donde fue extraída la llave y que permiten la localización del registro de origen. Aunado a esta técnica, generalmente se agregan herramientas de búsqueda selectiva, utilizando para este fin, operadores booleanos ("Y", "O" y "NO").

## **CAPÍTULO II**

### **FUNDAMENTOS**

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

El origen de la tecnología del Disco Compacto o CD (Compact Disc) se remonta a finales de la década de los setentas, como resultado de un proyecto conjunto entre las compañías Philips de Holanda y Sony Corp. de Japón. Inicialmente, el objetivo del proyecto consistía en crear un medio de grabación de audio digital. El impacto y penetración del Disco Compacto de Audio Digital o CD-DA (del inglés, Compact Disc - Digital Audio) permitió definir las características físicas y lógicas de este medio, mismas que fueron publicadas posteriormente (1.ed. 1980) como el "Libro Rojo".

Debido al gran potencial de almacenamiento de datos del CD-DA, en 1983 Philips y Sony propusieron extender sus especificaciones lógicas para implementar códigos binarios, obteniéndose así la publicación del "Libro Amarillo", en donde se exponen los estándares de fabricación del CD-ROM (del inglés, Compact Disc - Read Only Memory). Posteriormente, se publicaron nuevas versiones del "Libro Amarillo", las más importantes son la High Sierra (1986) y la ISO9660 (1987) de la Organización Internacional de Estándares, esta última, es a la que se apegan actualmente la mayoría de las aplicaciones basadas en CD-ROM. Este estándar permite que un CD-ROM pueda ser accedido por cualquier unidad lectora, instalada en computadoras que usan diferentes sistemas operativos.

Dada la versatilidad y capacidad de digitalizar diferentes tipos de información (audio, video, datos), la tecnología del CD ha dado lugar en los últimos años a una nueva familia tecnológica, en donde se integran, entre otros:

Disco Compacto de Audio Digital (**CD-DA** por sus siglas en inglés).

Disco Compacto - Memoria de Sólo Lectura (**CD-ROM** por sus siglas en inglés).

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

Disco Compacto de Arquitectura Extendida (**CD-ROM XA** por sus siglas en inglés).

Disco Compacto Interactivo (**CD-I** por sus siglas en inglés).

Disco Compacto de Video Digital Interactivo (**CD-DVI** por sus siglas en inglés).

Disco Compacto Escribible (**CD-R** por sus siglas en inglés) o Write Once.

Disco compacto de fotografías (**Photo-CD** por sus siglas en inglés).

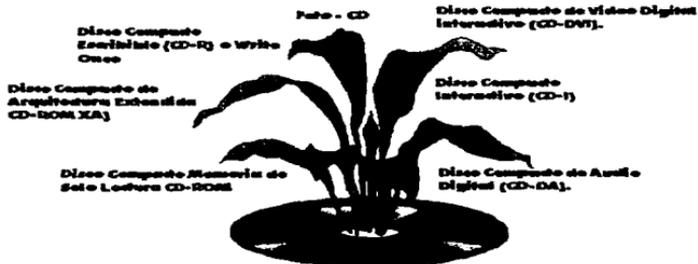


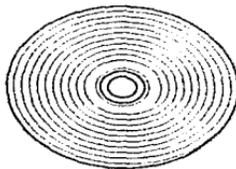
Figura 4: Familia tecnológica del CD.

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

### CARACTERÍSTICAS FÍSICAS Y LÓGICAS

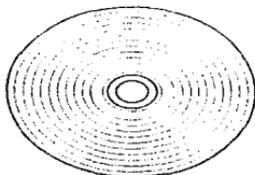
Un CD-ROM estándar, es un disco compuesto de varias capas, la más gruesa es de plástico de policarbonato, la siguiente, es un metalizado reflectante de aluminio, y para evitar la oxidación se aplica una cubierta de resina. Las dimensiones del CD-ROM son, 120 milímetros (mm) de diámetro, 1.2 mm de espesor y tiene en el centro una horadación de 15 mm. Los datos son almacenados en el disco en una sola pista en espiral, de forma semejante a los discos de audio de acetato, pero con la diferencia de que el inicio es en el centro y el fin en la periferia. La pista mide aproximadamente 4 km de longitud y se divide en sectores de igual tamaño cada uno. El ancho de la pista es de aproximadamente 600 nanómetros (nm) y las vueltas adyacentes de la pista están separadas aproximadamente 1,600 nm, dando una densidad de alrededor de 16,000 pistas (tracks) por pulgada. Los datos a lo largo de la espiral están grabados en forma de pequeñas depresiones de 120 nm de profundidad y longitud variable



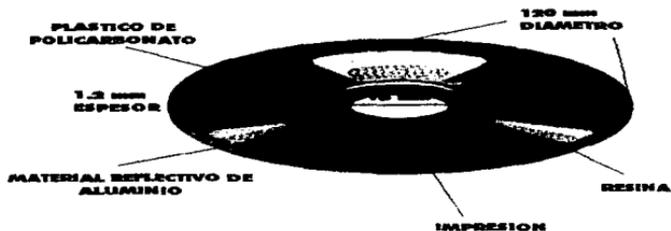
*Figura 5: Pista en espiral en el disco compacto, la pista inicia en el centro del disco.*

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---



*Figura 6: Pistas en un disco duro magnético, las pistas son círculos concéntricos.*



*Figura 7: Constitución del CD-ROM.*

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

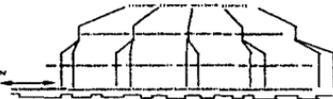
Bajo el estándar ISO9660 y debido a las técnicas de modulación empleadas, en un CD-ROM, cada byte se integra de 14 bits, mismos que son traducidos a código de 8 bits para efectos de comunicación con la computadora. En cada sector de un CD-ROM se almacenan 2352 bytes de información organizados de la siguiente manera:

	12	bytes de sincronización de datos.
	4	bytes de inicio (Header) para localización y modo.
+	2048	bytes de datos.
	4	bytes de detección de errores (CDE: por sus siglas en inglés).
	8	bytes no asignados.
	276	bytes códigos de corrección de errores (CCE: por sus siglas en inglés).
<hr/>		
	2352	

DATOS DE 8-BITS

DATOS DE 14-BITS

INTERCALACIÓN DE  
BITS PARA DEFINIR EL PATRÓN  
DE SINCRONÍA



NOTA:

UN 1 ES REPRESENTADO POR LA TRANSICIÓN DE UN REFLEJO A NO REFLEJO Y DE UN NO REFLEJO A REFLEJO. UN 0 ES REPRESENTADO POR LA AUSENCIA DE TRANSICIONES

*Figura 8: Modulación de Datos en un CD-ROM*

## 1.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

Cabe mencionar que mientras en un bloque de CD-ROM se opera con 2048 bytes para datos, en un CD-DA se utilizan 2336 bytes. La diferencia se debe a que el CD-ROM utiliza mayor cantidad de bytes en la corrección y detección de errores (ECCs/EDCs).

El bloque de inicio (heading) se compone de tres bytes de direccionamiento de sector y un byte de Modo. Se definen tres modos de sector para los CD-ROMs estándar; un sector de Modo "0" contiene solamente ceros (este sector podría representar un área en blanco en una imagen). En el modo "1", un sector contiene 2048 bytes de datos más 288 bytes de corrección/detección de errores (C/DE). En el modo "2" la cantidad de datos por sector se incrementa a 2336 bytes, utilizando el espacio de (C/DE) para datos, este modo es normalmente reservado para datos que son insensibles a errores ocasionales, tal como datos de audio y video digitalizados. Para datos sensibles, el Modo 1 de detección de error es muy efectivo, la probabilidad de un error no detectado es alrededor de  $10^{(-25)}$  o aproximadamente 1 bit por cada 2 mil billones de CD-ROM.

De forma semejante a los CD-DA, los CD-ROM pueden tener diferentes duraciones, los hay de 60 minutos que almacenan de 270.000 sectores y de hasta 74 minutos, con 333.000 sectores. Los 14 minutos adicionales se localizan en los 5 mm más externos del disco. Esta área es la más difícil de grabar bien y de mantener limpia, de modo que es común dejarla sin usar (de hecho los productores de CD-DA utilizan generalmente discos de 63 min.)

Los fabricantes citan capacidades de almacenamiento del CD-ROM ampliamente variables. Todo depende de si se están usando 270.000 o 333.000 sectores y de cómo se efectúen los cálculos. Un CD-ROM con 270.000 sectores y 2 K (2048 bytes) de datos error-correctados por sector tiene una capacidad de 552.960.000 bytes. Algunos fabricantes redondean la cifra a 552 MB, o dividen entre 1 K (1024 bytes) y redondean la cifra a 540 MB. El método correcto es dividir entre 1 MB (1'048.576 bytes), para dar un total

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

aproximadamente de 527 MB de datos error-correctados. Si los últimos 288 bytes de cada sector se usan para datos error-no-correctados en vez de ECCs/EDCs (dando 2336 bytes por sector), la capacidad de almacenamiento del disco se incrementa a 601 MB. Finalmente, usando 333,000 sectores por disco se tendrían 681,984,000 bytes de datos error-correctados que dan aproximadamente 650 MB de almacenamiento disponible, o cerca de 742 MB sin corrección de error.

### PRODUCCIÓN DE UN CD-ROM

El proceso de producción de un CD-ROM, se compone de varias etapas:

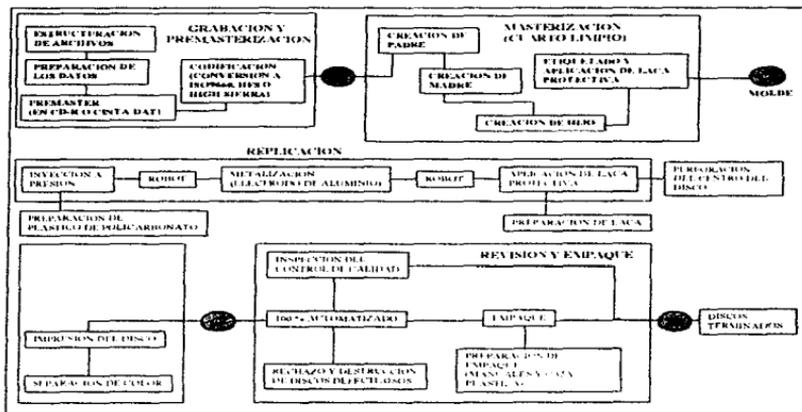


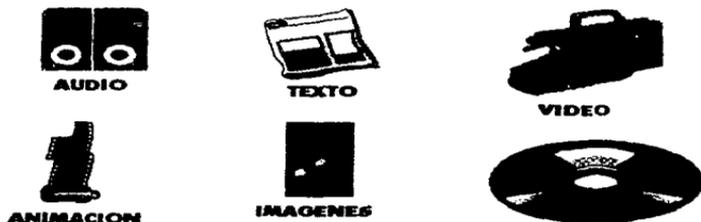
Figura 9: Proceso de producción del CD-ROM.

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

### Edición

Esta etapa, consiste en recopilar, traducir, seleccionar y codificar los datos que serán integrados en la aplicación. Como es natural, es necesario contar con un programa lo suficientemente poderoso y flexible como para manejar los diferentes tipos de datos a integrar, como lo pueden ser textos, gráficos, numéricos, sonidos, videos, etc. así como para tener un tiempo de respuesta lo suficientemente rápido como para contrarrestar la lentitud del tiempo de acceso del lector de CD-ROM (actualmente arriba de 180 ms). Por otra parte, es necesario contar con una interfaz adecuada al tipo de usuarios a los que va dirigida la aplicación.



*Figura 10: Diferentes tipos de datos almacenables en un CD-ROM.*

De todo el proceso de producción de un CD-ROM, este paso generalmente resulta el más costoso, el que toma más tiempo y el que resulta más problemático y difícil de cuantificar, sin embargo, es necesario tener siempre en cuenta que cualquier error cometido en este proceso significa un error repetido, a su vez, en cada uno de los discos y presentado a cada uno de los usuarios de la aplicación.

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

### Creación del disco Imagen

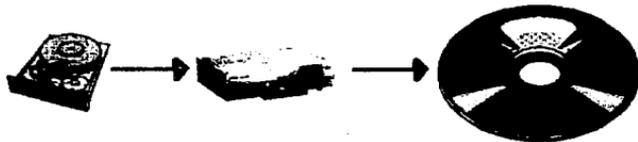
En esta etapa, todos los datos contenidos en la aplicación son trasladados a un factor de bloqueo de 2048 bytes, mismos que se usan en el CD-ROM. Generalmente el disco imagen es creado en un dispositivo de almacenamiento magnético.



*Figura 11: La simulación de un CD-ROM (disco imagen) se realiza en un medio magnético.*

### Simulación

Esta etapa consiste en ejecutar la aplicación como si se hiciera desde el propio CD-ROM. Para este proceso generalmente se utiliza hardware y software especiales para crear en un dispositivo magnético las mismas condiciones de operación de un CD-ROM.



*Figura 12: El CD-R o premaster requiere de un grabador especial.*

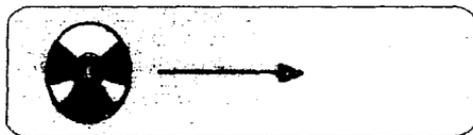
## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

### Premasterización

En esta etapa, los datos contenidos en el disco imagen son convertidos al estándar ISO9660 o HIGH-SIERRA proceso que consiste en agregar los bytes de sincronización, de inicio, y los códigos ECCs/EDCs. Asimismo, en este proceso los datos son grabados en un medio compatible con los equipos utilizados en la reproducción. Los medios comúnmente usados son: cinta magnética DAT y el CD-R, en el último caso, se obtiene un disco que ya es utilizable como un CD-ROM convencional.

### Masterización

Este proceso y el de replicación se llevan a cabo en fábricas con maquinaria altamente especializada y costosa (arriba de \$ 1,000,000.00 de U.S.D.). Los equipos son los mismos que se utilizan para reproducir los CD-DA. El proceso de masterización consiste en fabricar un molde, que es obtenido mediante la acción de un láser de alto poder sobre un plato de vidrio cubierto de un material fotoresistivo con el fin de escribir en el disco el patrón que corresponde a los datos codificados por medio del láser. Posteriormente el plato es cubierto por una capa metalizada (generalmente de níquel), para finalmente fabricar los moldes maestros de metal al separar, el níquel plateado del disco de vidrio, estos son llamados también "padres" de metal y sirven para obtener los moldes "madre" que serán utilizados por las máquinas de inyección de plástico con las cuales se fabricarán las réplicas del disco.



*Figura 13: Creación del master a partir del CD-R.*

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

### Replicación

En esta etapa final, intervienen los equipos que inyectan policarbonato puro dentro de los moldes madre. Una vez que se enfría, al plástico se le aplica una fina capa reflectante mediante la acción de un aparato especial denominado magnetron, el cual desprende átomos de una fuente de aluminio por medio de la acción de fuertes campos magnéticos. Posteriormente, es aplicada una capa protectora de resina, para finalmente proceder a la impresión de etiquetas y pruebas de calidad del disco.



*Figura 14: La replicación del CD-ROM se realiza a partir de un molde (master).*

En resumen, El CD-ROM es esencialmente un medio utilizado en la publicación de documentos interactivos y masivos, debido a su alta capacidad de almacenamiento, economía, y versatilidad para distribuir aplicaciones voluminosas en las que se requiera un gran volumen de copias.

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

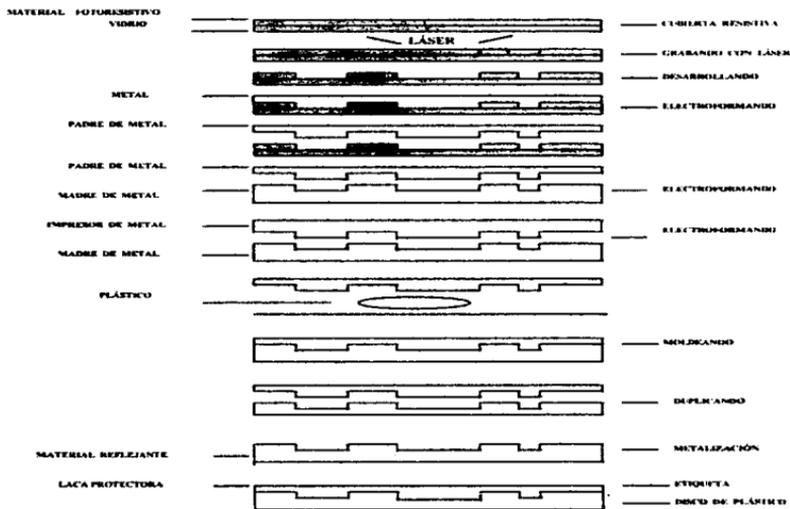


Figura 15: Producción de un CD-ROM.

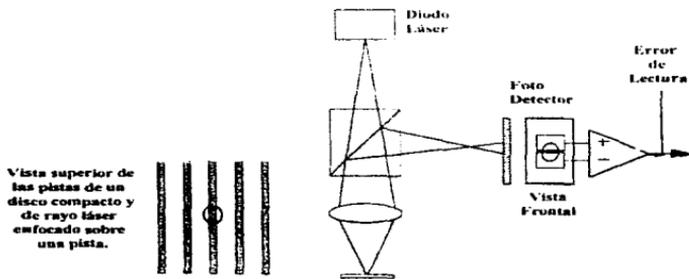
## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

### EQUIPO PARA LEER EL CD-ROM

A diferencia de los discos magnéticos convencionales, en los CD-ROM se almacena la información en una espiral de alta densidad, de forma continua, la cual inicia en el centro del disco. La espiral se encuentra dividida en sectores contiguos, lo cual debido a un proceso de optimización, durante la etapa de preparación, asegura que los archivos almacenados en este medio no sean grabados de manera fragmentada. La información que define archivos y directorios (FAT), reside en el inicio del espiral, ya que es el área que ofrece mayor rapidez de acceso. Los archivos de instalación que se usan algunas veces, conviene colocarlos al final de la espiral; archivos de uso constante conviene instalarlos lo más cerca posible del inicio. Archivos relacionados, conviene grabarlos de forma contigua. Otro aspecto importante de esta tecnología, es que la velocidad de lectura en un CD-ROM es constante y depende del tipo de lector. Por su velocidad de transferencia se clasifica en 1X, 2X, 3X, 4X, 8X y hasta 12X. En 1X, la velocidad de lectura es de aproximadamente 1.3 m de espiral por segundo, a una tasa de transferencia en la interfaz entre el controlador y la unidad de 150 Kbits/seg. La velocidad de rotación del disco es variable y depende de la posición de la cabeza de lectura, esto se debe a que existe mayor densidad de grabación en el interior que en el exterior del medio, por lo que en una unidad 1X convencional, las revoluciones pueden oscilar entre 200 RPM y 500 RPM. En el caso de un disco duro magnético, la velocidad de lectura es variable, ya que la velocidad de rotación es fija.

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO



*Figura 16: Sistema de lectura de un disco compacto*

Una unidad de CD-ROM convencional, se compone básicamente de una cabeza de lectura óptica, un soporte giratorio, un controlador y un módulo de procesamiento de señales. La cabeza de lectura, a su vez se encuentra compuesta de un diodo emisor de láser, un juego de lentes/espejos y una celda fotodetectora que recibe el haz de luz reflejado de la superficie del disco, todos estos elementos se encuentran montados sobre un mecanismo oscilante. La celda fotodetectora, incorpora un conjunto de fotodiodos encargados de determinar el enfoque del láser así como la posición en la pista del disco.

El controlador, es un circuito integrado que coordina las funciones de los sistemas mecánicos y lógicos.

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

El procesador de señales, demodula, descodifica y corrige los datos leídos del disco. En el proceso de lectura, la fluctuación en la intensidad de una señal luminosa es el medio por el cual es obtenida la información almacenada en los discos compactos.

### **Características determinantes en la elección de una unidad de CD-ROM**

- **Velocidad:** Este factor es determinante en el buen desempeño de las aplicaciones, por ejemplo, si se requiere utilizar el dispositivo en programas de multimedia, en donde se requiere acceder grandes cantidades de datos a una alta velocidad de transferencia, lo recomendable es seleccionar un lector 8X o 12X los cuales operan con un tiempo promedio de acceso típico de 135 ms a una tasa de transferencia de 1,2 Mbits/Seg y 1,8 Mbits/Seg respectivamente. Cuando la aplicación consiste en acceder datos de tipo textual o imágenes fijas, un lector de doble o triple velocidad con tasas de transferencia de 300 y 450 Kbits/Seg (2X y 3X respectivamente) es una opción razonable. Los lectores de velocidad sencilla, son dispositivos ya discontinuados del mercado y no son una opción recomendable. Es importante mencionar que si la aplicación originalmente fue grabada a velocidad 1X, la unidad 4X accederá los datos a 1X, en otras palabras, solamente se obtendrá un óptimo rendimiento cuando la aplicación coincide con la velocidad del lector.
- **Tamaño del buffer:** Típicamente se dispone de unidades con buffers de 64 K a 256 K, los más sofisticados cuentan con un buffer de 1 MB. Entre mayor sea el tamaño del

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

buffer, mejor es la respuesta de la unidad, ya que se requieren menos accesos al disco compacto en cada petición de lectura.

- **Interfaz:** Existen tres tipos de interfaz: la propietaria, IDE o AT y la SCSI en sus diferentes modalidades. La primera, es propia del fabricante, y sus características varían de acuerdo a la marca y modelo, en algunos casos, es posible conectar solo una unidad por tarjeta, en otros 8 o más. Las marcas propietarias más conocidas actualmente son SONY, MITSUMI y HITACHI. En cuanto a la interfaz IDE, el lector de CD-ROM se conecta directamente al controlador del disco duro inteligente, que es el estándar en la mayoría de las PCs actuales. En este caso, los principales fabricantes son SONY y TEAC. Por el momento, a la interfaz IDE, se conectan hasta dos dispositivos por controlador, y dos controladores por tarjeta. La configuración típica es un disco duro y una unidad de CD-ROM.

La interfaz SCSI es la más abierta, ya que la mayoría de los fabricantes ofrecen dispositivos compatibles con este estándar. Equipos PC, Macintosh, minis o macrocomputadoras, pueden incorporar interfaz interconstruida de tipo SCSI, por lo que en la mayoría de las plataformas, sólo basta conectar el dispositivo, instalar sus controladores y listo; en algunos casos no se requieren. En una interfaz SCSI, es posible instalar hasta 7 dispositivos por puerto, la cantidad de puertos depende del tipo de máquina y de la capacidad de direccionamiento de las tarjetas de interfaz, por mencionar un ejemplo, típicamente en una PC, es posible incorporar 4 puertos con hasta 28 dispositivos SCSI.

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

Debido a su capacidad de interconexión y a su velocidad de respuesta, las unidades de CD-ROM SCSI se utilizan en la mayoría de los servidores ópticos disponibles en el mercado.

- **Confiabilidad:** Esta característica es un reflejo de la calidad del dispositivo o bien de su tecnología. El valor de la confiabilidad de un dispositivo se mide en MTBFs, siglas que provienen del inglés *Mean Time Before Failure* (tiempo promedio de operación antes de falla). Un buen equipo ofrece MTBF mayor a 40,000 hrs., un equipo de regular calidad ofrece MTBF mayor a 25,000 hrs. Claro está que la vida útil depende en gran medida del trato que se le da y de las condiciones de operación de los equipos.
- **Precio:** Debido a la constante expansión del mercado, actualmente los precios de las unidades lectoras de disco compacto han visto disminuir su precio de forma importante, por mencionar algunas cifras, una unidad *Hitachi*<sup>®</sup> Mod.1503 cuando fue introducida al mercado en 1986, tenía un precio de lista de \$ 1,100 USD, actualmente una unidad de características superiores puede conseguirse por menos de \$150 USD.

### CD-ROM en ambiente de red y multiusuario

Según estadísticas, en la actualidad, cerca del 50% de los equipos tipo PC se encuentran conectados a redes de comunicación de datos, ya sea como terminales asíncronas o como clientes, estos equipos comparten o utilizan recursos de los servidores, tales como discos duros, impresoras, faxes, módem, equipos de comunicación, y por supuesto, no podían

## **II.1 LA TECNOLOGÍA DEL DISCO COMPACTO**

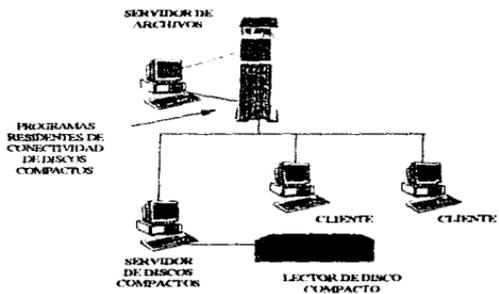
---

faltar los dispositivos ópticos tales como las unidades de CD-ROM, CD-R, Magneto-Ópticas y WORM.

La tecnología para compartir este tipo de dispositivos generalmente depende del sistema operativo o software de red que se esté utilizando, y podrían definirse dos tipos: los que llevan implícito el compartir estos recursos y los sistemas específicos, que pueden ser un complemento de los primeros, ofreciendo una mayor capacidad y confiabilidad en su funcionamiento.

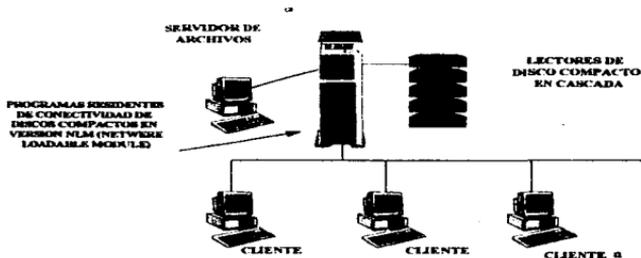
## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---



*Figura 17: Configuración típica de una red de área local con servidor de discos compactos dedicados (Versión para estación de usuario).*

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO



*Figura 18: Configuración típica de una red de área local con un servidor de archivos configurado para ofrecer funciones de servidor de disco compacto.*

### Sistemas con conectividad de CD-ROM implícita

- **Sistemas UNIX:** Comparten recursos de manera directa como una sesión en ambiente nativo multiusuario, o bien, via NFS a todos los clientes, incluyendo otros de diversas plataformas.

La principal ventaja de este ambiente, es el hecho de que es natural el acceso de múltiples usuarios, así como la capacidad de compartir recursos. La principal desventaja, es que al acceder aplicaciones en diferentes plataformas, se requiere contar con el software de la aplicación en ambiente nativo, lo cual no siempre puede ocurrir.

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

Esta contrariedad puede ser resuelta utilizando emuladores del ambiente deseado, como el SOFTPC o SOFTMAC para UNIX.

Si el software de la aplicación está disponible para el cliente, en ocasiones se requiere una emulación de los controladores del hardware de CD-ROM como las extensiones de MS para CD o el formato HighSierra en las PCs o el HFS para Mac.

- **Novell Netware 4.X y 3.12:** En este software se cuenta con un módulo de tipo NLM (del inglés: Network Loadable Module) en donde se pueden compartir los CD-ROM a través de la red a los clientes. La desventaja es que no es posible conectar varios dispositivos, ya que se ve afectado el rendimiento del servidor de forma drástica, llegando incluso a bloquear el sistema. Asimismo, los CD-ROM se montan como volúmenes, sin emular las extensiones de Microsoft para PCs.
- **Lantastic:** Para compartir los CD-ROM se utilizan comandos redirectores los cuales operan de manera semejante a los NFS de UNIX.
- **Windows for workgroups:** Este es uno de los sistemas operativos más económicos y populares del momento, con buenas características en cuanto a facilidad de uso y de emulación.

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

### Sistemas de conectividad de CD-ROM específicos

Algunos de los programas comerciales más conocidos son:

- OPTINET de On-Line Systems.
- LAN-CD de Logicaft.
- SCSI Express de MDI International.
- CD-NET de Meridian Data.

Todos, excepto SCSI Express emulan extensiones de Microsoft, lo cual garantiza la ejecución apropiada de la mayoría de las aplicaciones. El concepto de funcionamiento es semejante al de un servidor de archivos. En el servidor de CD-ROM se reserva y administra la memoria RAM, la cual opera a manera de caché. Sin gran carga de usuarios, la velocidad de respuesta puede ser más rápida que en modo local, sin embargo, conforme aumenta el número de usuarios, la velocidad de respuesta se degrada de forma sensible. Este problema se debe principalmente a los cuellos de botella que se generan por la baja velocidad de transferencia de las unidades de CD-ROM y al ensamble y desensamble de datos en la red, así como al ancho de banda de la misma.

Un caso excepcional es el operar redes de CD-ROM utilizando el software LanCD, en el cual es posible crear una copia de los discos compactos en discos duros de alta capacidad, mejorando significativamente el tiempo de respuesta de la aplicación. Otra solución que ayuda a mejorar la velocidad de transferencia de datos de los servidores de CD-ROM a la red, es la incorporación de conmutadores (*switch*) de alta velocidad, que permiten operar a tasas de 100 Mbits/Seg o más. Otra opción en este mismo sentido, es utilizar redes basadas en estándares de alta velocidad como Fast-Ethernet, FDDI, 100BaseVG o ATM.

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

### El Futuro de la Tecnología del CD-ROM

En su lucha constante por dominar el mercado con las nuevas tecnologías de almacenamiento de datos, la alianza de las compañías Philips/Sony y Toshiba/Warner han lanzado dos nuevas propuestas para discos compactos de alta densidad, se trata del **HDCD** (por sus siglas en inglés High Density Compact Disc) y el **DVD** (Digital Versatile Disc). Comparados con el CD-ROM actual, las características más sobresalientes son:

Características	CD-ROM	HDCD	DVD
Diámetro	12 cm	12 cm/8cm	12 cm
Espesor	1.2 mm	1.2 mm	1.2 mm (2 caras)
Capacidad	680 MB (modo 1)	3.7/1.3 GB (1 cara) 7.4/2.6 GB (2 caras)	5GB (1 cara) 10GB (2 caras)
Espesor de la pista	1.6 micrón	0.85 micrón	0.725 micrón
Longitud de onda del láser (nanómetros)	780	635	650
Apertura numérica	0.45	0.52	0.6
Método de corrección de errores	CIRC	CIRC+	RS-PC**
Modulación	EFM	EFM+	???
Vel. Transferencia (MB/Seg)	Variable (1X,2X...)	3.3 MB promedio 10.42 MB max.	4.94 MB promedio
Modos de sector	Modo 1; Forma 1,2	Modo 3	???
Tamaño de sector (bytes)	2048	2048	2048

Figura 19: Comparación de las características de los discos compactos de alta densidad.

\* del inglés: Cross Interleaved Reed Solomon Code

\*\* del inglés: Reed Solomon - Product Code

## II.1 LA TECNOLOGÍA DEL DISCO COMPACTO

---

Muy probablemente, la propuesta de Sony/Philips, sea la que defina el nuevo estándar, y esto se debe a que los futuros lectores de alta densidad también serán compatibles con las especificaciones actuales, lo que permitirá a la gran base de usuarios instalada, continuar utilizando los títulos actuales de disco compacto.



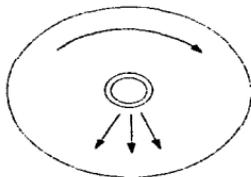
**Figura 20:** Microfotografía CD-ROM densidad estándar y CD-ROM de alta densidad

#### IV. DISEÑO DEL SISTEMA

---

##### **Limpieza y conservación**

En los discos compactos, los patrones de bits son leídos mediante la incidencia de un haz de láser en la superficie de espejo, esta acción no produce ningún tipo de desgaste físico, por lo que en teoría un CD puede durar indefinidamente, sin embargo, descuidos durante su manipulación pueden llegar a ocasionar daños irreparables, por lo que una de las principales recomendaciones para obtener un funcionamiento libre de errores, es mantener la superficie de espejo libre de impurezas, rayones, etc , evitando dejar huellas digitales, así como almacenar los discos fuera de su empaque original. El operar los discos en algunos tipos de lectores, sin trampa de polvo por varias semanas provoca la acumulación de partículas en la superficie óptica, por lo que es necesario limpiarlos periódicamente, con un paño o piel que no deje residuos, o bien mediante aparatos especiales de limpieza con agentes químicos (generalmente a base de alcohol isopropílico). El disco debe limpiarse siempre con movimientos perpendiculares al centro, y no en forma circular como mucha gente equivocadamente supone.



*Figura 21: Limpieza de un disco compacto - Debe ser siempre del centro hacia afuera.*

## **II.2 CONSIDERACIONES FÍSICAS Y LÓGICAS DEL CD-ROM**

---

### **OPTIMIZACIÓN EN EL ACCESO DE ARCHIVOS**

El CD-ROM al ser heredero de la tecnología del disco compacto de audio digital o CD-DA, es un medio de almacenamiento que presenta algunos problemas interesantes para el manejo de archivos. La desventaja principal del disco compacto es la organización de los sectores de datos en forma de espiral continua, lo que provoca un tiempo de acceso relativamente lento. Por citar un ejemplo, si se compara un proceso que opere con archivos en RAM, este podría tardar 20 seg. en un disco rígido tomaría 58 días, mientras que en un CD-ROM de velocidad sencilla tardaría dos años y medio. Ventajosamente, el disco compacto es un medio de alta capacidad, es económico y durable. Este tipo de características son las que obligan a considerar la construcción y organización de los archivos de índices y de datos como un factor crítico para optimizar el tiempo de respuesta de las aplicaciones desarrolladas en estos medios.

Con el propósito de optimizar el acceso a los dispositivos de almacenamiento secundario, los programas recuperadores de información en su mayoría utilizan estructuras de índices basadas en algoritmos de árboles B y B+, lo que permite localizar rápidamente las llaves deseadas con sólo recorrer algunos niveles en las ramas de los índices. Sin embargo, aún en el caso de las estructuras de árboles B simples, podría requerirse de una cantidad significativa de accesos a disco para localizar cualquier información.

El origen de las estructuras de árbol B, se remonta al año de 1972 cuando los señores R. Bayer y E. Mc. Creight de la compañía Boeing, publicaron el artículo "Organización y

## II.2 CONSIDERACIONES FÍSICAS Y LÓGICAS DEL CD-ROM

---

Mantenimiento de Índices Ordenados Extensos”, los autores de esta investigación nunca explicaron el porque del nombre “árbol B”, sin embargo, se piensa que podría deberse por una estructura “balanceada”, “amplia” (broad) o arborescente (bushy). algunas personas afirman que la “B” podría deberse también a “Boeing” ó “Bayer”

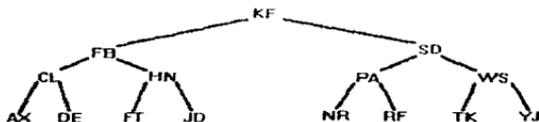
El problema fundamental que dió origen a las estructuras de árbol, es el hecho de que el acceso a dispositivos de almacenamiento secundario es demasiado lento y costoso. Este planteamiento, puede apreciarse mejor si se analizan los siguientes hechos:

- En un archivo secuencial ordenado, la búsqueda binaria requiere una gran cantidad de desplazamientos a diferentes pistas de un disco. Como los desplazamientos son costosos, esta acción podría consumir más tiempo del deseable. Por ejemplo, en un conjunto de 15 registros, cuatro desplazamientos serían suficientes para localizar cualquier llave. En el caso de un arreglo de 1000 elementos, se requiere en promedio de 9.5 accesos.
- En un archivo secuencial ordenado, el insertar una nueva llave, implica afectar a todas las demás. Esto resulta poco práctico, sobre todo cuando se trata de archivos con una gran cantidad de elementos. Por todo lo anterior, era necesario encontrar una estructura en donde cualquier nueva inserción solo afectara a los elementos del conjunto local.
- Para una lista ordenada de llaves, el proceso de una búsqueda binaria, puede representarse de la siguiente forma:

## II.2 CONSIDERACIONES FÍSICAS Y LÓGICAS DEL CD-ROM

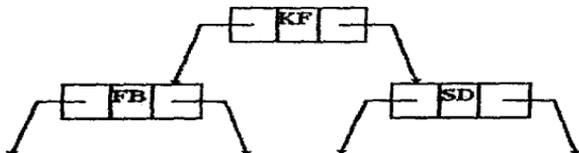
AX CL DE FB FT HN JD KF NR PA RF SD TK WS YJ

Lista de llaves ordenadas



*Figura 22: Representación de una búsqueda binaria en la misma lista a través de una estructura de árbol.*

Lógicamente, la representación de un nodo del árbol puede ser a través de registros que incorporen el campo de datos y los campos de ligas o direcciones a los nodos hijos de la izquierda y la derecha.



*Figura 23: Segmento de un árbol binario de búsqueda*

## II.2 CONSIDERACIONES FÍSICAS Y LÓGICAS DEL CD-ROM

Físicamente, el archivo que contiene la representación del árbol binario de búsqueda, podría ser como se muestra a continuación



	llave	lga	lga
	ca	der	
0	FB	10	8
1	JD		
2	RF		
3	SD	6	3
4	AX		
5	YJ		
6	PA	11	2
7	FT		

	llave	lga	lga
	ca	der	
8	HN	7	1
9	KF	0	3
10	CL	4	12
11	HR		
12	DE		
13	WS	14	5
14	TK		

Figura 24: Arreglo representando una estructura de árbol binario.

En la figura siguiente se puede observar, que al agregar una nueva llave, sólo se modifica la rama local, sin necesidad de reorganizar todo el arreglo.

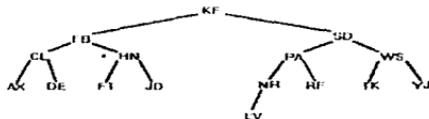
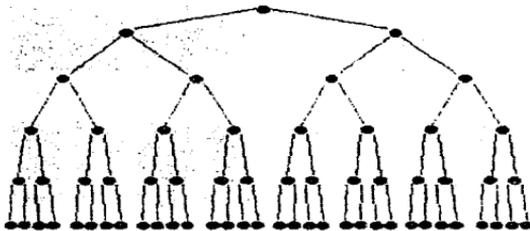


Figura 25: Representación en una estructura de árbol de una búsqueda binaria agregando un nuevo nodo (LV).

## II.2 CONSIDERACIONES FÍSICAS Y LÓGICAS DEL CD-ROM

---

Dado que un sector es la mínima unidad de información direccionable en un disco y que en el caso de un CD-ROM, el tamaño de un sector es de 2 K, y considerando además, que el tiempo de acceso en el mejor de los casos es de 165 mseg. se puede observar que utilizando un factor de bloque múltiplo de 2 K, se puede obtener un mejor rendimiento de cualquier aplicación. Entre mayor sea el tamaño de bloque, mayor cantidad de información se presentará al usuario, con una consecuente disminución del número de acceso de lectura. Esta característica, es posible aplicarla en el almacenamiento de archivos de índices, y es la base de los árboles binarios paginados.



*Figura 26: Árbol binario paginado.*

## II.2 CONSIDERACIONES FÍSICAS Y LÓGICAS DEL CD-ROM

---

### ESTRUCTURA DE DIRECTORIOS

La mayoría de los sistemas operativos como DOS y UNIX utilizan estructuras de directorios que utilizan ligas entre archivos. Por ejemplo, en el caso de UNIX la representación de la ruta de un archivo podría ser la siguiente:

```
/etc/usr/lpp/SoftWindows/bin/autoexec.bat
```

el acceso al archivo autoexec.bat implica abrir y leer el contenido de los directorios: etc, usr, lpp, SoftWindows y bin. Para archivos residentes en un disco duro o en memoria RAM, probablemente no es significativo el tiempo de acceso, pero en un CD-ROM esto podría resultar poco funcional. Es recomendable evitar estructuras de directorio con demasiadas ligas, y en la medida de lo posible colocar todos los archivos en un solo nivel.

## II.3 ARCHIVOS INVERTIDOS

---

### CONSTRUCCIÓN DE ÍNDICES MEDIANTE ARCHIVOS INVERTIDOS

Este proceso consiste en analizar el contenido de cada registro que compone una base de datos para construir una lista de los términos que permiten una identificación positiva del propio registro. Debido a que las bases de datos textuales generalmente consisten en abstracciones o bibliografías, es posible considerar a la mayor parte de los datos como candidatos a términos de búsqueda, debido a que de antemano existe un análisis de un documento. De esta forma solo es necesario eliminar de los índices los artículos y preposiciones.

Por ejemplo:

Supóngase que se tienen los siguientes registros:

30249

NR

Postpurchase consumer evaluations compilant action and

**TI1**            **TI2**            **TI3**            **TI4**            **TI5**

repurchase behavior

**TI7**            **TI8**

Franken, Dick A.

AU

Descriptors: CONSUMER ATTITUDES; CONSUMER BEHAVIOR

**DE1**

**DE2**

30450

NR

Micromedia Action I

**TI1**            **TI2**

Macromedia

AC

Descriptors: SOFTWARE-PRESENTATIONS; SOFTWARE-GRAPHICS

**DE1**

**DE2**

## II.3 ARCHIVOS INVERTIDOS

---

Un archivo invertido podria construirse de la siguiente forma

LLAVE	REGISTRO	POS
ACTION	30249	T15
	30450	T12
BEHAVIOR	30249	T18
COMPILANT	30249	T14
CONSUMER	30249	T12
CONSUMER ACTITUDES	30249	DE1
CONSUMER BEHAVIOR	30249	DE2
EVALUATIONS	30249	T13
FRANKEN, DICK A	30249	AU
MACROMEDIA	30450	T11
	30450	AC
POSTPURCHASE	30249	T11
REPURCHASE	30249	T17
SOFTWARE-PRESENTATIONS	30450	DE1
SOFTWARE-GRAPHICS	30450	DE2

*Figura 27: Ejemplo de un archivo invertido.*

La palabra "and" no forma parte del indice debido a que es una palabra no significativa.

Como se puede notar, por cada ocurrencia de un mismo termino, se derivará una nueva entrada a la columna de registro, de tal forma que no es necesario repetir el termino cada vez que ocurra.

## **CAPÍTULO III**

### **ANÁLISIS DEL SISTEMA**

### **III. ANÁLISIS DEL SISTEMA**

---

#### **Introducción**

El manejo de grandes volúmenes de datos requiere de programas que permitan una recuperación rápida y precisa, que eviten la molesta tarea de revisar registro por registro para localizar información específica. Actualmente, se pueden encontrar herramientas que se presten al procesamiento de datos textuales como procesadores de texto o algunos manejadores de bases de datos, sin embargo, no siempre se consigue tener la funcionalidad y resultados deseables. Por otro lado, la mayoría de los programas disponibles comercialmente requieren del pago de derechos de uso, lo cual implica una carga adicional por cada réplica. Es por estas razones y las que a continuación se exponen, que fue necesario desarrollar un programa a la medida, que considerara todos los productos y capacidades deseables para el tipo de información contenida en la base de datos DIRFAX, y tener una base de programación que puede actualizarse y utilizarse en versiones posteriores del producto.

Para recopilar y plantear los requerimientos del sistema de información, objeto de este trabajo, se han utilizado algunos aspectos de la metodología denominada CORE (del inglés *Controlled Requirements Expressions*), el cual propone una serie de etapas encaminadas a definir el problema a analizar, para después dividirlo en unidades o puntos de vista a considerar. Según este método, las etapas por cubrir son las siguientes:

#### **Definición del problema**

Actualmente, la base de datos DIRFAX se distribuye en papel, y en medios magnéticos, en este último caso, el producto no cuenta con un sistema de consulta y recuperación de

### III. ANÁLISIS DEL SISTEMA

---

información, el problema principal, es que el usuario debe tener experiencia en el uso de algún programa compatible con el formato de archivos de Xbase. Por otra parte, los archivos que componen el banco de datos ocupan tanto espacio, que resulta poco práctico distribuirla en discos flexibles, ya que esto consume una gran cantidad de tiempo en su reproducción e instalación, y no siempre el usuario cuenta con el espacio suficiente para copiar el producto a su disco duro.

Actualmente, la producción del documento en papel resulta muy costosa y difícil de cuantificar, sobre todo ante los constantes cambios en los precios de la materia prima, generalmente a la alza, por lo que es necesario encontrar un medio alternativo lo suficientemente económico y estable para su distribución, considerando al mismo tiempo que sea versátil y atractivo al usuario final.

Debido a que no se desea limitar el producto a plataformas de ambiente gráfico, es deseable que el sistema opere desde DOS, sin olvidar que el programa de consulta de la información debe ser de fácil manejo, puesto que pueden existir usuarios con poca o ninguna experiencia en informática. Al mismo tiempo, se debe contar con capacidades de recuperación tales, que permitan búsquedas sofisticadas que cubran las necesidades de usuarios experimentados.

### **III. ANÁLISIS DEL SISTEMA**

---

#### **Expectativas del cliente y usuarios**

En esta etapa, se entrevistaron directamente a los productores y editores del banco de información (Ediciones el Globo y Difusión Científica CD-ROM), mismos que externaron las siguientes necesidades

#### **Expectativas del productor**

- Contar con un programa que permita la recuperación de la información.
- El programa debe ser fácil de operar y no requerir una interfaz gráfica sofisticada.
- Su licencia no puede ser costosa, ya que será distribuido de forma masiva.
- Dar la sensación de ser rápido en su operación.
- Permitir la incorporación de imágenes digitalizadas.

#### **Expectativas del editor**

- Contar con capacidades típicas de los sistemas recuperadores de información disponibles comercialmente.
- Ser compatible con el formato proporcionado por el editor, ya que el programa será utilizado durante el proceso de recopilación de información y actualización.
- Ser de fácil manejo, incluso para usuarios sin experiencia.
- Contar con la instalación incluida en el programa.

#### **Estructuración de los datos**

La estructura de datos de DIRFAX, se muestra en la figura 28.

### III. ANÁLISIS DEL SISTEMA

---

NOMBRE DEL CAMPO	LONGITUD	TIPO
NOMBRE	50	Caracter
DIRECCIÓN	50	Caracter
COLONIA	30	Caracter
CIUDAD	20	Caracter
CODIGO	5	Caracter
ESTADO	20	Caracter
TELÉFONO	15	Caracter
FAX	15	Caracter
GIRO	35	Caracter
IMAGEN	1	Lógico

Figura 28: Estructura del archivo DIRFAX

#### **Análisis de las restricciones**

Las restricciones consideradas para el desarrollo del proyecto fueron las siguientes:

- Los programas deben operar en cualquier tipo de equipo, incluyendo en computadoras personales sin ambiente gráfico.
- El programa debe ser de fácil operación con ayuda disponible en cualquier momento, ya que los usuarios pueden no tener experiencia, sin embargo, debe ser lo suficientemente ágil, para ser atractivo para usuarios avanzados.
- Los programas deben mantener la compatibilidad Xbase.
- El tiempo de acceso del medio de almacenamiento (CD-ROM) es lento.

#### **Herramientas**

##### **Programas de desarrollo**

- La base de datos DIRFAX, es desarrollada en Dbase IV \*.
- Dada la compatibilidad, disponibilidad y experiencia, los programas recuperadores fueron compilados utilizando Clipper 5.2.

### **III. ANÁLISIS DEL SISTEMA**

---

- Para la visualización de gráficos, se utilizaron las librerías TBASE, mismas que utilizan los formatos TIF, PCX y BMP.
- El programa de edición de CD-R es propietario de la compañía Topix, y se suministra con el equipo de grabación descrito a continuación.

#### **Equipo**

- Para la codificación de los programas: computadora Action Note 4SLC/25, que cuenta con: Procesador 80486 SLC, 4MB de memoria en RAM, 80MB en disco duro, monitor VGA, sistema operativo 6.0 .
- Scanner de color HP IIC. Impresora HP Láser Jet IV .
- Debido a que el proceso de preparación de datos y su traslado a CD-R, requiere de equipos con altas prestaciones, se utilizó una computadora DTK 80486DX2 a 66 Mhz. con 20 MB de memoria RAM y disco duro de 1 GB con un tiempo de acceso promedio de 14 ms. Monitor SVGA de 14" .
- El equipo grabador de CD es un Kodak CDR-521 de doble velocidad, con tarjeta de interfaz Adaptec modelo 1542CF. Los controladores forman parte de los programas de instalación de la tarjeta.

#### **Requerimientos mínimos para los usuarios de la aplicación**

- Microcomputadora PC compatible con DOS 3.0 o superior.
- Unidad de CD-ROM.

### III. ANÁLISIS DEL SISTEMA

---

- Cualquier tipo de monitor
- 640 K de RAM mínimo
- Impresora de matriz o laser

#### **Lugar de trabajo**

La empresa editora: Difusión Científica CD-R OM

Según se expone en el libro "Ingeniería de Software Explicada" de Norris y Ricby, el análisis de sistemas puede enfocarse básicamente desde dos puntos de vista:

- a) En base a las funciones que debe desempeñar
- b) Por el tipo de datos a manipular

El diseño de la presente aplicación ha considerado ambos enfoques, sin embargo, no se encontró una notación lo suficientemente adecuada para el tipo de sistema desarrollado. Para propósitos prácticos, dada la complejidad y lo extenso del código generado por este programa, se exponen sus especificaciones funcionales por medio de la descripción de sus procedimientos, así como también, la incorporación de algunos diagramas de flujo de datos que describen de manera gráfica algunos módulos fundamentales

En cuanto a la metodología utilizada en el diseño del sistema, se consideró el SSAD (Structured Systems Analysis and Design- Diseño de Análisis Estructurado), el cual resultó el más adecuado después de evaluar los expuestos por algunos autores de la bibliografía a la que se tuvo acceso

Aplicando el método SSAD al presente caso, se separó el diseño del sistema en las siguientes etapas:

### III. ANÁLISIS DEL SISTEMA

---

#### **Análisis de los requerimientos operativos y problemas**

Dado que las partes involucradas tanto en la producción como en la edición del disco compacto propusieron requerimientos demasiado abiertos (generales) se decidió, estudiar algunos sistemas recuperadores de información con el propósito, de obtener un marco de referencia. Se realizó un estudio de las funciones y capacidades de los sistemas recuperadores comerciales disponibles. Los productos revisados fueron: Medline, que utiliza el software SPIRS de Silver Platter, Computer Select de ZIT Davis Electronic que utiliza CD-ANSWARE de la compañía Datavare, el Financiero en CD-ROM con el software ROMWARE de Nimbus, DIALEX 1 que utiliza el software Micro-CDS/ISIS de UNESCO y COMPENDEX de Dialog Information Services, de los cuales se obtuvieron los siguientes resultados:

**Definición.** Los sistemas recuperadores de información (IRS Information Retrieval System) tienen como objetivo principal proporcionar herramientas que permitan procesar, organizar, almacenar, diseminar y recuperar elementos de información que satisfagan las necesidades de determinados sectores de usuarios.

**Capacidades.** La mayoría de los sistemas recuperadores de información disponibles en el mercado funcionan con esquemas de archivos invertidos, es decir, el sistema se compone de un archivo de documentos y uno o más archivos auxiliares de índices, generalmente almacenados en estructuras de acceso rápido. En el índice, son incluidas las direcciones que dieron origen al término indexado, de esta forma, para localizar un conjunto de registros que traten sobre determinado tema, bastara con localizar el término y recuperar la información en base a las direcciones asociadas. Para obtener la información mas exacta y

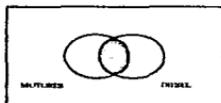
### III. ANÁLISIS DEL SISTEMA

concreta posible, los sistemas comerciales utilizan técnicas basadas en el álgebra de conjuntos, de tal forma que una búsqueda puede construirse por combinaciones de términos y operadores lógicos ("Y", "O" y "NO"). Los más sofisticados pueden incluir además, unión exclusiva, proximidad y uso de paréntesis. Por término medio, las operaciones implementadas son intersección, unión y negación.

#### DIAGRAMAS DE OPERACIÓN DE CONJUNTOS

#### DESCRIPCIÓN

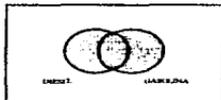
OPERACION "Y" (AND)



En este caso literalmente se recuperan los registros que contengan las palabras motores y diesel.

**MOTORES "Y" DIESEL**

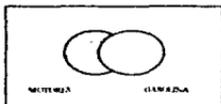
OPERACION "O" (OR)



Para recuperar registros que contengan la palabra diesel o gasolina.

**DIESEL "O" GASOLINA**

OPERACION "NO" (NOT)



Para recuperar registros que contengan la palabra motores pero no gasolina.

**MOTORES "NO" GASOLINA**

Figura 29: Operaciones de conjunto.

### III. ANÁLISIS DEL SISTEMA

---

Para efectos de presentación de resultados, generalmente se cuenta con la capacidad de enviar a impresión a través de diferentes puertos o bien, con la posibilidad de descargar a disco duro o disco flexible, indicando simplemente el nombre de archivo y ruta. Asimismo, generalmente se permite seleccionar diferentes tipos de reportes de salida ya predefinidos.

En algunos productos, se utilizan técnicas de compresión de archivos para optimizar la capacidad de los medios de almacenamiento, sin embargo, esto generalmente consume mayores recursos de procesamiento en el acceso a los datos.

Algunos de estos productos son versiones en CD-ROM de sistemas diseñados para disco duro.

A fin de evitar la carga innecesaria de los índices, se omiten palabras carentes de significado o bien palabras que no permitan la identificación del registro de donde provienen.

Los bancos de datos especializados en ciertas áreas del conocimiento como es el caso de Medline en Medicina, utilizan diccionarios para permitir la localización de información a través de términos generales o específicos. Un diccionario en otras palabras, es una lista de los términos autorizados para determinadas áreas del conocimiento y son establecidos por entidades normalizadoras en la materia.

Todos los sistemas cuentan con ayuda en línea.

La búsqueda puede efectuarse a través de la consulta directa a índices o bien a un carrusel de términos. En los casos más sofisticados, es posible ampliar una búsqueda a través de técnicas de hipertexto, es decir, marcando un término en el registro recuperado para que a partir de él, se realicen nuevas búsquedas.

### III. ANÁLISIS DEL SISTEMA

---

Algunos programas permiten salvar estrategias de búsqueda (conocido en inglés como bookmarks).

Todos los productos trabajan en el sistema operativo DOS, solamente dos productos tienen también la opción de trabajo en ambiente WINDOWS. Solamente el programa SPIRS de Medline tiene versión en CD-ROM multiplataforma (DOS, Windows, Mac y algunos UNIX). Todos los programas pueden ejecutarse utilizando emuladores de DOS y Windows en MAC y UNIX.

En una base de datos de tipo directorio, resulta necesario recuperar a través de cualquier término que produzca una identificación positiva del registro, no importando la posición que guarde dentro del campo de origen. En los lenguajes de programación basados en el estándar Xbase y en la mayoría de los sistemas que operan sobre bases de datos de campos fijos, la construcción de índices se limita al tratamiento de campos completos o subcadenas de estos, por lo que con esta técnica sólo es posible localizar información a través de una consulta en donde se conozca exactamente como se ingresó el dato, quedando fuera de identificación aquellos registros en donde una cadena se encuentre a la mitad del campo. Un método para localizar registros en donde una cadena intermedia podría representar una identificación positiva, es el recorrer secuencialmente toda la base de datos y comparar carácter por carácter agrupando los éxitos. Es evidente que esto resulta muy ineficiente en archivos de gran tamaño debido a la gran cantidad de tiempo que el proceso implica.

En los sistemas Xbase, la indexación y recuperación por palabra sólo es posible desarrollando rutinas para este propósito.



### **III. ANÁLISIS DEL SISTEMA**

---

#### **Capacidades consideradas en el sistema**

- Estructura del sistema formada por un grupo de herramientas, fácilmente adaptables para la indexación y recuperación de información de bases de datos de tipo directorio, compatibles con el formato Xbase
- Búsqueda por palabra, inclusive palabras intermedias de un campo
- Recuperación a través de operaciones lógicas ("Y", "O" y "NO")
- Impresión en formato personalizado, tanto de reportes en general como de etiquetas y cartas (mail merge)
- Instalación personalizada y operación en CD-ROM.
- Búsqueda por frase
- Recuperación por "carrusel de términos".
- Remarcado de los términos localizados.
- Índices basados en estructuras de "archivos invertidos".
- Ligado de imágenes en algún formato gráfico.
- Manejo de historia de búsqueda (bookmarks).
- Impresión y descarga a disco por conjuntos, en formatos predefinidos y personalizables.
- Menús seleccionables en varios idiomas.
- Menús de operación estandarizada con selección por flechas.
- Ayuda en línea.
- Configuración del entorno de operación (unidad de disco, memoria).

## **CAPÍTULO IV**

### **DISEÑO DEL SISTEMA**

## IV. DISEÑO DEL SISTEMA

---

### Diseño Lógico de datos

A pesar de que los sistemas Xbase utilizan campos de longitud fija e indexación de campo a llave de uno a uno, se ha implementado la programación necesaria para recuperar la información por medio de estructuras de archivos invertidos. La descripción y relación entre archivos se describe a continuación.

### Archivo: DIRFAX.DBF

**Descripción:** Base de datos principal. Almacena los datos de un directorio empresarial, el cual incorpora más de 80,000 registros. La estructura del archivo se muestra en la figura 31.

NOMBRE DEL CAMPO	LONGITUD	TIPO
NOMBRE	50	Caracter
DIRECCIÓN	50	Caracter
COLONIA	30	Caracter
CIUDAD	20	Caracter
CÓDIGO	5	Caracter
ESTADO	20	Caracter
TELÉFONO	15	Caracter
NUMFAX	15	Caracter
GIRO	35	Caracter
IMAGEN	1	Lógico

*Figura 31: Estructura del archivo DIRFAX.*

#### IV. DISEÑO DEL SISTEMA

---

##### Archivo: KEYLST.DBF

**Descripción:** Archivo de llaves. Se utiliza para almacenar las llaves generadas por campo y palabra de DIRFAX.DBF. La información que integra permite localizar el conjunto de anotaciones en el archivo de ligas para cada llave. Su estructura se muestra en la figura 32.

NOMBRE DEL CAMPO	LONG	TIPO	DESCRIPCIÓN
KNO	6	Caracter	Número de llave
KEY	30	Caracter	Llave o término
PIN	8	Número	No. de registro inicial de anotación de la primer liga
POS	6	Número	No de veces que se repite la llave ( <i>postings</i> )
REC	6	Número	No de registros en donde aparece la llave ( <i>hits</i> )
FLD	1	Caracter	No. de campo de donde se extrae la llave

Figura 32: Estructura del archivo KEYLST.DBF

##### Archivo: LNKLIST.DBF

**Descripción:** Archivo de ligas. Se utiliza para almacenar las anotaciones KEYREF (número de registro en DIRFAX.DBF), para cada número de llave KEYNUM. Este archivo contiene una gran cantidad de registros (cerca de 1,000,000). Su estructura se muestra en la figura 33

NOMBRE DEL CAMPO	LONG	TIPO	DESCRIPCIÓN
KEYNUM	6	Caracter	Número de llave
KEYREF	8	Caracter	Número de registro que contiene la llave.

Figura 33: Estructura del archivo LNKLIST.DBF

## IV. DISEÑO DEL SISTEMA

---

### Archivo: SHIST.DBF

**Descripción:** Archivo de operaciones temporales. En él se almacenan los conjuntos de registros generados en cada búsqueda. Para un óptimo rendimiento, se recomienda que este archivo resida en memoria RAM por medio de un disco virtual. La estructura se muestra en la figura 34.

NOMBRE DEL CAMPO	LONG	TIPO
Set	4	caracter
recNum	8	caracter

*Figura 34: Estructura del archivo SHIST.DBF*

Los archivos de llaves y de ligas en conjunto forman lo que se ha denominado "archivos invertidos", que como se ha mencionado, no es otra cosa que la descomposición de un archivo fuente en una lista de palabras. Su estructura se encuentra diseñada para permitir la manipulación de los conjuntos de registros ligados a cada una de las llaves generadas a través de operaciones booleanas ("Y", "O" y "NO").

### Diseño Lógico de procesos

El sistema se compone de tres módulos principales, que de acuerdo a su función se dividen en:

**IFGEN:** Módulo generador de índices. Archivos invertidos.

**OSIRIS:** Módulo recuperador de información.

**INSTALAR:** Módulo de instalación asistida desde CD-ROM.

## IV. DISEÑO DEL SISTEMA

El desarrollo de la presente aplicación puede resumirse en los procesos mostrados en la figura 35, en donde la captura y la definición de la estructura de la base de datos ya ha sido elaborada previamente por el productor.

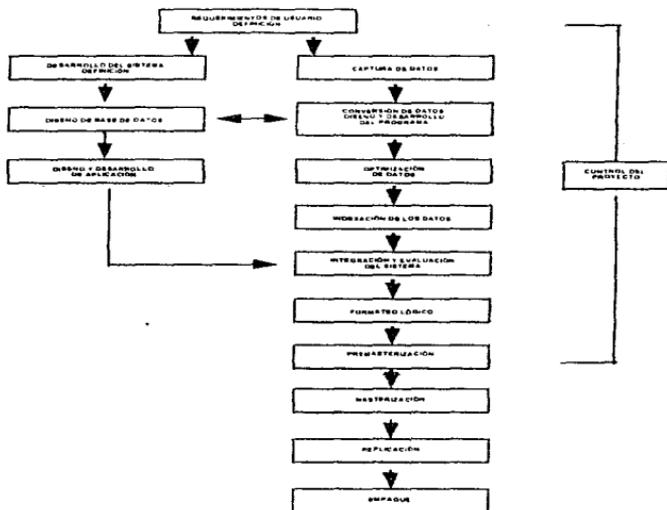


Figura 35: Procesos para el desarrollo de la aplicación.

#### **IV. DISEÑO DEL SISTEMA**

---

##### **Secuencia de procedimientos implementados en IFGEN.PRG**

Primeramente, se recorre la base de datos y se extraen de todos los campos definidos para la recuperación, cada una de las palabras que lo componen (Procedimiento EXTRAELlave), se considera una palabra cualquier cadena de caracteres contenida entre espacios, debido a que pueden existir caracteres de puntuación al término de cada cadena, se requiere de un proceso adicional para eliminar este inconveniente (Procedimiento DEPURA). Posteriormente, cada palabra se pasa por un proceso de filtrado (Procedimiento CHKSTOPWORD), en donde se descartan las "palabras no significativas", es decir, todas aquellas que no produzcan ninguna identificación de información, este conjunto esta compuesto principalmente por preposiciones, articulos, etc. (A, AL, ANTE, BAJO, ...etc.) En el programa IFGEN, el conjunto es representado a través de un arreglo, ordenado en forma secuencial ascendente, denominado STOPWORD, el cual es declarado como una variable de memoria. El proceso de filtrado, se basa en un algoritmo de búsqueda binaria, el cual consiste en aproximaciones sucesivas mediante la reducción del arreglo de comparación. La palabra extraida es comparada con el elemento central (pivote) del arreglo STOPWORD, si existe una igualdad, la palabra es "no significativa" y esta será descartada. Si la palabra extraida es mayor, el conjunto de comparación se reduce a la mitad superior del arreglo, si es menor a la mitad inferior. El proceso termina cuando la tabla de comparación queda vacia, lo cual significa que la palabra es "significativa", y pasa a formar un término en el indice. Para dar de alta un nuevo término se revisa si ya existe en el archivo correspondiente (KEYLST), si esto ocurre, simplemente se hará una nueva anotación (entendemos anotación como el número del registro en donde ocurre el término)

#### **IV. DISEÑO DEL SISTEMA**

---

al archivo de ligas (LNKLST). Después del proceso de extracción de términos, el archivo de ligas se organiza para reunir en orden ascendente todas las anotaciones encontradas para cada llave. Finalmente, a cada llave se le asigna la dirección de inicio y fin de sus anotaciones en el archivo de ligas (Procedimiento LNKGEN), lo que permitirá en el programa de búsqueda la operación entre conjuntos de ligas de manera eficiente.

##### **IFGEN.PRG**

- Extracción de llaves (EXTRAELLAVE)
- Filtro de caracteres especiales (DEPURA)
- Filtro de palabras no significativas (CHKSTOPWORD)
- Agrupamiento de anotaciones (SORTLNK)
- Definición de límites para cada llave (LNKGEN)

## IV. DISEÑO DEL SISTEMA

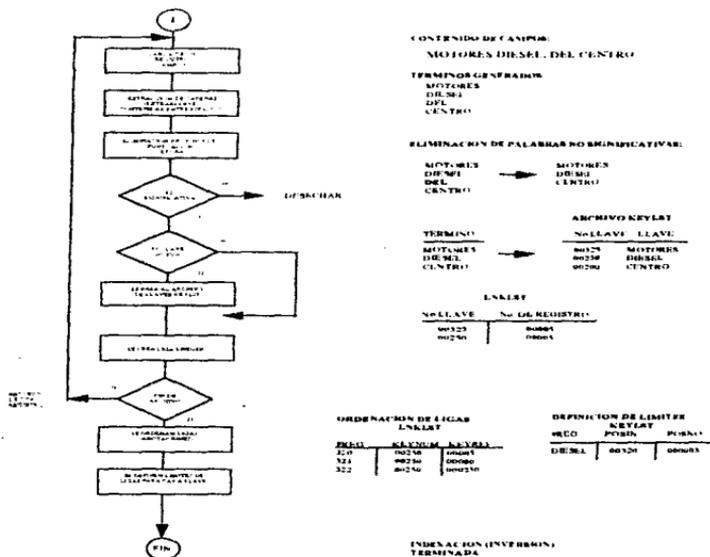


Figura 36: Diagrama de flujo del proceso de generación de índices (Archivos invertidos). Programa IFGEN.

#### IV. DISEÑO DEL SISTEMA

##### DEPURACIÓN Y AGRUPAMIENTO DE ÍNDICES POR CAMPO

El programa IFGEN genera como salida los archivos invertidos correspondientes a cada una de las formas de localizar información en la base de datos dirfax.

Archivos de Llaves	Archivos de Ligas	Descripción
KeyLst.dbf	LnkLst.dbf	Archivos invertidos que contienen todos los términos significativos contenidos en la base de datos.
NomLst.dbf	NomLnk.dbf	Archivos invertidos que contienen todas las frases del campo nombre de la base de datos.
DirLst.dbf	DirLnk.dbf	Archivos invertidos que contienen todas las frases del campo dirección de la base de datos.
ColLst.dbf	Collnk.dbf	Archivos invertidos que contienen todas las frases del campo colonia de la base de datos.
CiuLst.dbf	CiuLnk.dbf	Archivos invertidos que contienen todas las frases del campo ciudad de la base de datos.
EstLst.dbf	EstLnk.dbf	Archivos invertidos que contienen todas las frases del campo estado de la base de datos.
GirLst.dbf	GirLnk.dbf	Archivos invertidos que contienen todas las frases del campo giro de la base de datos.

Figura 37: Archivos invertidos: permiten obtener los archivos de tipo índice del sistema.

#### **IV. DISEÑO DEL SISTEMA**

---

Los procedimientos IFGEN generan los índices para cada campo en archivos separados, mismos que deben ser agrupados en el archivo KeyLst.dbf que contiene todas las llaves y en el archivo LnkLst que contiene todas las ligas, este procedimiento se realiza en forma manual revisando todas las llaves generadas, con el propósito de eliminar aquellas que terminen resultando inútiles o no significativas aun después del proceso de extracción automático realizado por IFGEN (el cual, incorpora un proceso de eliminación de palabras no significativas)

El procedimiento se lleva a cabo de la siguiente forma: Primeramente se carga el archivo KeyLst.dbf y se procede a eliminar las llaves carentes de utilidad, se toman los números correspondientes a los registros inicial (LI) y final (LF) del bloque de datos, para incorporarlos al programa OSIRIS.PRG con el fin de efectuar la búsqueda de este campo, posteriormente se cargan los siguientes archivos de llaves realizando, el mismo procedimiento

Una vez agrupadas todas las llaves se procede a crear el archivo KEYLST.NTX el cual se construye con la instrucción siguiente:

INDEX ON Fld + " " + Key

Por otra parte es necesario también agrupar todos los archivos de ligas en un archivo final LnkLst.dbf. Durante este procedimiento se toman los números correspondientes al registro final de cada bloque de datos (OS)

Para la base de datos DIRFAX utilizada en este sistema los valores obtenidos se muestran en la figura 38.

#### IV. DISEÑO DEL SISTEMA

KEYLST.DBF LI = 1 LF = 38302	LNKLST.DBF OS = 0
NOMLST.DBF LI = 38303 LF = 72934	NOMLNK.DBF OS = 538623
DIRLST.DBF LI = 72935 LF = 107277	DIRLNK.DBF OS = 585375
COLLST.DBF LI = 107278 LF = 109848	CLLNK.DBF OS = 632093
CIULST.DBF LI = 109849 LF = 110183	CIULNK.DBF OS = 678000
ESTLST.DBF LI = 110184 LF = 110215	ESTLNK.DBF OS = 724751
GIRLST.DBF LI = 110216 LF = 111586	GIRLNK.DBF OS = 771502
<b>Total de registros en :</b> <b>KEYLST.DBF = 111586</b>	<b>Total de registros en :</b> <b>LNKLST.DBF = 818242</b>

*Figura 38: Desplazamientos y cantidad de registros contenidos en los archivos invertidos, generados para DIRF.LV*

#### **IV. DISEÑO DEL SISTEMA**

---

Todos estos valores son requeridos en el programa "OSIRIS.PRG" y en el procedimiento "BÚSQUEDA".

En general todos los módulos que componen al sistema, han sido desarrollados siguiendo la técnica de programación estructurada, de tal forma que el control de flujo se lleva a través de un procedimiento principal, desde el cual y dependiendo de las opciones seleccionadas en los menús, son llamados cada uno de los procedimientos, mismos que al terminar, regresan el control al menú principal. El flujo de control puede observarse en el diagrama de procedimientos mostrado en la figura 39.

#### IV. DISEÑO DEL SISTEMA

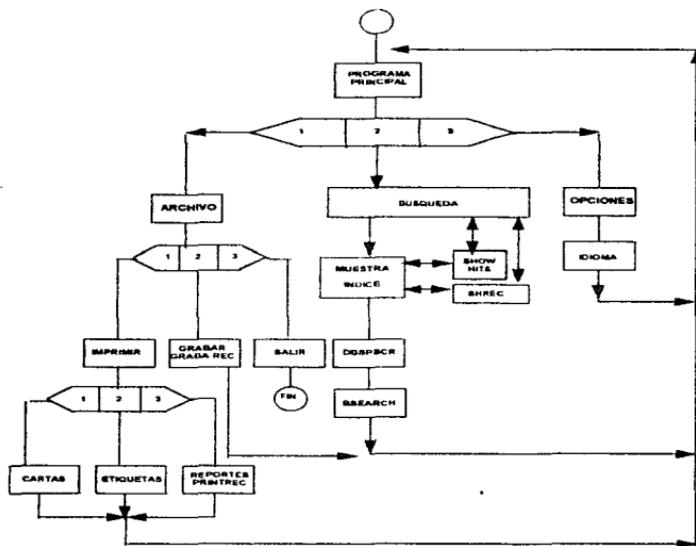


Figura 39: Diagrama de procedimientos. OSIRIS.PRG.

#### IV. DISEÑO DEL SISTEMA

---

En su primera versión, el sistema aplicado a la base de datos DIRFAX, dispone de las opciones mostradas en la figura 40.

PRIMER NIVEL	SEGUNDO NIVEL	TERCER NIVEL
ARCHIVO	IMPRIMIR  GRABAR SALIR	CARTAS ETIQUETAS REPORTES PERSONALIZADOS
BUSQUEDA	TODOS NOMBRE DIRECCIÓN COLONIA CIUDAD ESTADO GIRO INICIAR	
OPCIONES..	IDIOMA	ESPAÑOL INGLÉS

*Figura 40: Menus del sistema*

Se ha procurado ofrecer al usuario final características estandarizadas en el manejo de los menús, de tal forma que por medio de las teclas de movimiento del cursor, y <Enter>, cualquier opción puede ser seleccionada, mostrando un submenú a manera de cascada.

#### **IV. DISEÑO DEL SISTEMA**

---

Ejemplo:

**Menú principal de OSIRIS aplicado a DIRFAX**

<b>Archivo</b>	<b>Búsqueda</b>	<b>Opciones</b>	<b>OSIRIS 1.0</b>
----------------	-----------------	-----------------	-------------------

##### **OSIRIS.PRG - Programa Principal**

En el programa principal de OSIRIS.PRG, básicamente se ha implementado la definición de archivos de trabajo, variables globales, arreglos, constantes, y el menú principal que dentro de ciclos DO WHILE y un conmutador de tipo CASE, realiza las llamadas correspondientes a los procedimientos ARCHIVO, BÚSQUEDA Y OPCIONES.

##### **OSIRIS.PRG - Procedimiento ARCHIVO**

En este procedimiento, solamente se ha implementado las llamadas de las opciones definidas: IMPRIMIR , GRABAR y SALIR. En el primer caso, se puede elegir entre imprimir CARTAS, ETIQUETAS o REPORTES. La elección de alguna opción llamará a los procedimientos del mismo nombre. Al seleccionar GRABAR, se invoca al procedimiento GRABAREC, en el cual se ha implementado descarga a un archivo especificado por el usuario y SALIR, nos posiciona en el S.O.

##### **OSIRIS.PRG - Procedimiento BÚSQUEDA**

Concretamente, en este caso, al aplicar OSIRIS a DIRFAX, las opciones de búsqueda disponibles son las siguientes:



#### **IV. DISEÑO DEL SISTEMA**

---

Al seleccionar alguno de los campos de búsqueda, el contenido del índice se muestra en forma de carrusel de términos, esto es, el cursor se mantiene fijo en una posición y los datos son los que efectúan un movimiento de recorrido, el cual puede ser controlado por las teclas de movimiento del cursor (Inicio, Fin, Flecha arriba y abajo, Av. Pág. y Ret. Pág.). Esta característica, facilita la localización de datos, ya que solo se requiere ingresar la raíz del término deseado, lo que evita tener que conocer exactamente como se escribe. El procedimiento que realiza esta tarea es MuestraIndice.

#### **OSIRIS.PRG - Procedimiento MUESTRAINDICE**

Cada vez que se selecciona un campo, después de ingresar el término a localizar, es creado un conjunto que contiene la información de los registros que cumplen con condición de búsqueda. Las operaciones de conjunto implementadas en el programa ("Y", "O" y "NO") son realizadas en el procedimiento MAKESET, el cual recibe dos parámetros de referencia, POSIN, que indica el desplazamiento inicial en el archivo de ligas LNKLST, y POSNO el cual contiene el número de apuntadores para cada término. El traslado de apuntadores se realiza por medio de un arreglo de memoria en bloques de 4096 elementos, simulando una memoria caché, los datos recolectados se trasladan a SHIST.DBF, el cual se utiliza como archivo de trabajo. Opcionalmente y con el propósito de ofrecer una mayor velocidad de respuesta, el programa de instalación puede configurar el sistema para que el archivo SHIST resida en un disco virtual en memoria RAM.

A cada conjunto generado le es asignado un número que lo identifica. Si el conjunto generado es el primero, sus elementos son copiados directamente al archivo de trabajo, la

#### **IV. DISEÑO DEL SISTEMA**

---

razón, es que se requieren como mínimo dos conjuntos para realizar cualquier operación. A partir del conjunto 2, los elementos del último conjunto primero son operados con los del primero, trasladándose el resultado al final del archivo.

##### **OSIRIS.PRG - Procedimiento SHIST.DBF**

El proceso de búsqueda puede resumirse en las siguientes acciones:

- Muestra el submenú de búsqueda.
- Dependiendo del campo seleccionado, se inicializan los parámetros de inicio y de fin del conjunto de apuntadores (LI,LF) correspondiente en LNKLST (procedimiento BÚSQUEDA).
- Inicializa el desplazamiento en el archivo LNKLST.
- Localiza el término más cercano y los que lo rodean, de tal modo que se muestre el acercamiento progresivo.
- Al seleccionar el término con <Enter>, se crea un conjunto de aciertos, si antes se han realizado otras búsquedas el programa solicitará la operación lógica que aplicará entre el último conjunto y el anterior.
- Traslada el conjunto a SHIST para operarlo con el conjunto anterior si es que existe.
- Crea un nuevo conjunto resultado de la operación, mostrándose el resultado de la operación.
- En este momento, se pueden mostrar los resultados de la búsqueda oprimiendo la tecla <F10> o la historia de búsquedas realizadas hasta el momento.

#### **IV. DISEÑO DEL SISTEMA**

---

- Durante el proceso, se han implementado las siguientes posibilidades y acciones:

##### **<Retroceder>**

- Retrocede un carácter en la solicitud del término a buscar y localiza en el archivo índice el que se desea visualizar.
- Muestra el nuevo término más cercano y los que lo rodean en la ventana de carrusel.

##### **<Ret. Pág.>**

- Retrocede 20 términos en el índice verificando no sobrepasar el límite inicial.
- Se muestra el nuevo término seleccionado y los que lo rodean en la ventana de carrusel.

##### **<Av. Pág.>**

- Avanza 6 posiciones en el índice verificando no sobrepasar el límite final.
- Presenta nuevamente el carrusel de términos.

##### **<Inicio>**

- Retrocede hasta el inicio de la lista de términos, y se muestra el carrusel a partir del primer elemento.

##### **<Fin>**

- Avanza hasta el último término en el archivo de índices.

#### **IV. DISEÑO DEL SISTEMA**

---

- Se muestra el carrusel de términos a partir de la nueva posición.

##### **<Flecha hacia arriba>**

- Retrocede un término a partir de la línea de selección.
- Se muestra el carrusel de términos a partir de la nueva posición.

##### **<Flecha hacia abajo>**

- Avanza un término a partir de la línea de selección.
- Se muestra el carrusel de términos a partir de la nueva posición.

##### **<F8>**

- Muestra el historial de búsquedas efectuadas hasta el momento.

##### **<Esc>**

- Cierra la ventana de índices y retorna al submenú de búsqueda.

##### **<Esc> + <Esc>**

- Sale del programa a DOS.

##### **<Enter>**

- Efectúa una búsqueda en el archivo de ligas (LNKLST) con el procedimiento BSEARCH de acuerdo al término que se encuentra localizado en la línea de selección.

#### **IV. DISEÑO DEL SISTEMA**

---

- Corre el procedimiento **MAKESET** el cual se encarga de colocar esta información resultante en el archivo **SHIST** como un primer conjunto de datos y en el caso de ya existir uno efectúa la operación lógica entre ambos y genera un tercer conjunto que representa el resultado de dicha operación
- Guarda los datos en el archivo histórico

##### **<F10>**

- Efectúa el procedimiento **DISPLAY RECORD** el cual permite ver todos los datos que cumplen con la llave de búsqueda en una ventana. Esta información se obtuvo tomando los números de registro del archivo resultante **SHIST** y efectuando una búsqueda de ellos en la base de datos principal.

##### **Cualquier otra entrada**

- Localiza el término más acercado a la cadena ingresada.
- Muestra el carrusel de términos a partir del primer registro que contiene.

Para evitar el manejo simultaneo de varios archivos abiertos, en **KEYLST.DBF**, se han agrupado los índices generados para todas las opciones de búsqueda, por lo que en cada llamada para consulta, son necesarios los valores de desplazamiento que determinan los límites correspondientes a cada campo. Las variables utilizadas son: **L1** que almacena el límite inferior y **LF** para el límite final.

#### IV. DISEÑO DEL SISTEMA

---

##### **OSIRIS.PRG - Procedimiento MAKESET**

Este procedimiento se encarga de realizar las operaciones de conjunto seleccionadas en el menú de búsqueda. Inicialmente se solicita la operación que se aplicará entre el conjunto nuevo y el anterior, para esta tarea, es utilizado un menú en el cual se muestran las opciones disponibles ("Y", "O" y "NO"), mismas que para efectos de presentación, en este programa, han sido sustituidos por los caracteres "\*" , " + " , y " ^ " respectivamente.

Fisicamente, los archivos LNKLST y KEYLST representan una estructura de "archivos invertidos", ambos tienen un campo en común que los relaciona. Una vez que se ha seleccionado un término o llave de KEYLST, a través de MUESTRAINDICE, en MAKESET se copia el conjunto de ligas del archivo LNKLST.DBF a SHIST.DBF por medio de un arreglo de memoria, el cual tiene la función de actuar como un *buffer*, la selección de ligas se lleva a cabo gracias a que durante el proceso de indexación se han definido las dimensiones del conjunto de ligas. El proceso funciona de la siguiente forma, cada llave (KEY) de KEYLST.DBF tiene definido un número único que la identifica KNO, el cual es el mismo que se utiliza para identificar la procedencia de cada anotación en LNKLST.DBF. El conjunto de ligas se localiza directamente por la posición del primer elemento cuyo valor se almacena en el campo PIN, el número de elementos a procesar es determinado por el campo POS, debido a que los índices de todos los campos han sido almacenados físicamente en los mismos archivos, lo que los diferencia es un valor de desplazamiento que pasa a través de la variable OS (desplazamiento).

## **IV. DISEÑO DEL SISTEMA**

---

### **OSIRIS.PRG - Procedimiento BDSPSCR**

Este procedimiento muestra el índice a partir de la llave más cercana a la cadena que se ingresa, verificando que los términos mostrados en la ventana de índice no sobrepasen el límite inicial y final, dejando resaltada la línea de selección. La ventana consta de 15 renglones y el término localizado en el renglón 8 es resaltado. Literalmente, el procedimiento realiza la siguiente secuencia de acciones:

- Limpia la ventana de exhibición de índices entre los puntos 7,19 y 21,77
- Muestra los encabezados de la ventana en el idioma en curso.
- Inicializa el contador de renglones (variable R) con 7, posición representa el primer renglón de la ventana.

Si el término localizado -7 es menor que el límite inicial, lo mueve al inicio de la ventana. Mientras R sea menor que la última línea de la ventana, y si el término está entre los límites inicial y final.

- Muestra el término verificando que al llegar a la línea 14 se resalte la línea.
- Incrementa el contador de renglones.

Fin del ciclo.

Retorna al procedimiento que lo llamó.

### **OSIRIS.PRG - Procedimiento BSEARCH**

Este procedimiento, se encarga de localizar el término más cercano a la cadena de caracteres ingresada en MUESTRAINDICE. En el proceso de preparación de datos, al

#### **IV. DISEÑO DEL SISTEMA**

---

archivo de llaves KEYLST.DBF le es construido un índice de tipo \* NTX para el cual es utilizada la siguiente fórmula

**FLD+"-" +KEY (Número de campo, Delimitador, Llave)**

por lo que al realizar una búsqueda, a la cadena a localizar se le debe agregar el prefijo **FLD+"-"**.

Literalmente, el procedimiento BSEARCH realiza las siguientes acciones:

- Concatena el número de campo, el caracter especial y la cadena de caracteres a localizar.
- Efectúa la búsqueda con esta llave total
- Inicializa el apuntador del índice con el número de término encontrado.
- Regresa el control al procedimiento que lo llamó.

#### **OSIRIS.PRG - Procedimiento SHOWHIST**

Este procedimiento, tiene la tarea de mostrar la historia de todas las búsquedas realizadas en la sesión de consulta. El limite máximo antes de elegir la opción INICIAR en BÚSQUEDA es 99.

Literalmente, las acciones realizadas en este procedimiento son las siguientes:

- Salva la pantalla actual.
- Abre una ventana y muestra el contenido del vector hist línea por línea.
- Al teclear un caracter cierra la ventana y restaura la ventana anterior.
- Regresa a la subrutina que los llamó

#### **IV. DISEÑO DEL SISTEMA**

---

##### **OSIRIS.PRG - Opciones - Cambio de Idioma**

Debido a que se pretende distribuir aplicaciones a nivel internacional, el sistema ha sido diseñado para ofrecer menús y mensajes en inglés y español, por lo que la instalación define los valores predeterminados para este propósito, sin embargo, se puede cambiar el idioma en cualquier momento; en todos los procedimientos del programa, se utiliza una bandera global (LN=1 español, 0=inglés), para determinar el idioma activo.

ESTA TESIS HA DEBE  
SALIR DE LA BIBLIOTECA

## **CAPÍTULO V**

## **CONCLUSIONES**

## V. CONCLUSIONES

---

Al aplicar los parámetros de evaluación expuestos por Paul Nieuwenhuysen<sup>1</sup>, encontramos que el programa OSIRIS, objeto de este proyecto, cumple con la mayoría de las capacidades deseables en un sistema recuperador de información textual, según el citado autor, los criterios son los siguientes

### **Recuperación interactiva**

#### **I - Exhibición de índices.**

El software puede listar los términos indexados, por ejemplo, para dar a un usuario sin experiencia una idea del contenido de la base de datos.

**RESULTADO:** Si lo cumple, se ha implementado en el procedimiento MUESTRAINDICE

Si esto ocurre, el software puede mostrar el número de ocurrencias.

**RESULTADO:** Si cumple.

En este punto, el programa debe presentar términos equivalentes o relacionados, o bien, incluir la facilidad de acceso a través de diccionarios.

**RESULTADO:** Cumple en cierto grado, ya que el acceso a la información se realiza en todo momento vía diccionario, la implementación de sinónimos no depende tanto del programa, ya que es posible indexar cualquier palabra ingresada en cualquier campo. Esta

---

<sup>1</sup> CD-ROM Year Book 1989-1990, Evaluating Text Storage and Retrieval Software, pag. 313-319. © 1989, Microsoft Press, Washington. Impreso con permiso

## V. CONCLUSIONES

---

característica, más bien depende de la labor de los analistas de la información, mismos que tendrían que elegir la equivalencia entre términos

### **Búsqueda**

**1 - Es posible consultar la base de datos utilizando operadores lógicos o su equivalente:**

- AND
- OR
- NOT

por medio de menús o comandos como: "\*", "+" y "^".

**RESULTADO:** Si cumple. Se han implementado dichas operaciones por medio de menús.

**2 - Búsqueda por proximidad.**

¿Pueden localizarse dos términos adyacentes ?

**RESULTADO:** No implementado. Implica un mayor nivel de procesamiento en los índices y en las operaciones de conjunto

¿Pueden localizarse dos términos separados por un número determinado de palabras?

**RESULTADO:** No cumple.

¿Pueden localizarse registros que contengan dos términos en un mismo campo ?

**RESULTADO:** Si cumple, es posible efectuar búsquedas sucesivas en un mismo campo.

## V. CONCLUSIONES

---

¿Pueden localizarse registros que contengan dos términos en una misma frase ?

**RESULTADO:** No aplica, ya que en este caso, el producto no maneja textos.

**3 - ¿Existe límite en el número de operadores por cada búsqueda? (en la práctica esta limitación puede no ser importante en algunos casos).**

**RESULTADO:** No, los conjuntos generados en una búsqueda pueden ser el resultado de aplicar una serie de operadores

**4 - ¿Puede utilizarse el paréntesis junto con operadores lógicos para indicar el orden en que las búsquedas deben ser ejecutadas ?**

**RESULTADO:** No aplica, dado que las consultas no se ingresan de manera literal, sino por menús.

**5 - Esta disponible la operación de truncación a la derecha.**

¿Si esto ocurre, el usuario determina esta operación o se encuentra implícita ?

**RESULTADO:** Si, se encuentra implícita al ingresar el término a localizar, ya que conforme este se vaya digitando, el programa muestra los términos más cercanos, permitiendo identificación de aquellos que contengan la raíz.

¿Puede el usuario especificar el número de caracteres a operar con truncación ?

**RESULTADO:** Si, como se indicó anteriormente.

## **V. CONCLUSIONES**

---

¿Esta disponible la operación de truncación a la izquierda ?

**RESULTADO:** No implementado.

¿Si esto ocurre la respuesta es rápida o es secuencial y lenta?

**RESULTADO:** No implementado.

¿Es posible la utilización de más de una forma de truncación?

**RESULTADO:** No implementado.

¿Es posible el uso de la operación de truncación junto con operadores lógicos o de adyacencia?

**RESULTADO:** Si, implícito en la búsqueda.

6 - ¿Puede la búsqueda ser limitada a un campo determinado ?

**RESULTADO:** Si, la búsqueda se realiza seleccionando de entrada un campo específico o a todos los campos.

7 - ¿Puede el programa realizar búsquedas secuenciales, es decir por términos no indexados?

**RESULTADO:** No implementado, ya que esto no es práctico en un CD-ROM por su tiempo de acceso.

## **V. CONCLUSIONES**

---

**8 - ¿Se presenta el número de éxitos (hits) encontrados?**

**RESULTADO:** Si, en todo momento, se presenta la cantidad de registros recuperados en la última búsqueda.

**9 - ¿Pueden ser realizadas búsquedas con varios términos y ser desplegado el resultado (hits) de cada uno?**

**RESULTADO:** Si, se ha implementado al mostrar la historia de búsqueda

**10 - ¿Puede realizarse una búsqueda fonética?**

**RESULTADO:** No estrictamente, ya que esta característica es simplemente un acercamiento al término requerido.

**11 - ¿Puede darse peso a un término determinado ?**

**RESULTADO:** Esto se aplica en todo momento, ya que implica ejecutar en orden una búsqueda, siendo esta la filosofía de uso del programa.

**12 - ¿El sistema puede utilizar el código ASCII extendido como parte de los caracteres que componen un término de búsqueda?**

**RESULTADO:** Si.

**13 - ¿Pueden realizarse búsquedas con caracteres en mayúsculas y minúsculas ?**

**RESULTADO:** Si. Es indistinto.

## **V. CONCLUSIONES**

---

**14 - Se pueden realizar búsquedas numéricas como:**

Mayor que

Menor que

Igual a

De ... hasta ...

Otras ?..

**RESULTADO:** No aplica, ya que el procedimiento no incorpora datos numéricos

**15 - ¿Puede utilizarse una búsqueda anterior en una nueva?**

**RESULTADO:** No completamente, cada nuevo conjunto, es el resultado de las operaciones de búsqueda anteriores

**16 - ¿El programa puede ordenar un conjunto seleccionado por el usuario ?**

**RESULTADO:** No implementado

**17 - ¿Pueden realizarse análisis de los conjuntos recuperados?**

**RESULTADO:** No implementado

Si esto ocurre, los datos pueden ser exportados más o menos en forma automática.

**RESULTADO:** A nuestro juicio, esto no tiene que ver con la pregunta anterior, sin embargo en OSIRIS, es posible descargar resultados en archivos ASCII.

## **V. CONCLUSIONES**

---

**18 - ¿Puede ser desplegada la historia de búsqueda, con el resultado de cada conjunto creado ?**

**RESULTADO:** Si. Existe una opción específica

**19 - ¿Puede ser almacenada la estrategia de búsqueda ?**

**RESULTADO:** No, implementado

### **Facilidades de salida**

**1 - ¿El usuario puede elegir los campos a ser impresos o almacenados ?**

**RESULTADO:** Si, el usuario puede definir sus propios formatos de salida.

**2 - ¿Puede el usuario definir un orden en la salida ?**

Si esto ocurre,

Se puede elegir cualquier campo para ordenación ?

Se pueden definir varias llaves de ordenación ?

Pueden eliminarse palabras no significativas de una llave de ordenación ?

**RESULTADO:** No implementado.

## **V. CONCLUSIONES**

---

**3 - ¿La salida puede contener identificadores (nombres de campo)?**

**RESULTADO:** Si, ya que los reportes son completamente personalizables

**4 - ¿Se pueden eliminar registros duplicados ?**

**RESULTADO:** Si, el programa lo hace automáticamente.

**5 - ¿Puede ser marcado el término recuperado en los registros que lo contienen ?**

**RESULTADO:** No en la salida

**6 - ¿En la salida se indica el número relativo dentro del conjunto ?**

**RESULTADO:** Si, pero es a criterio del usuario.

**7 - ¿Pueden ser producidas salidas de tipo Llaves en contexto KWIC ? o Llaves y contexto KWAC ?**

**RESULTADO:** No implementado

**8 - ¿Pueden ser mostrados caracteres ASCII extendidos ? Estos pueden ser utilizados en ordenaciones de salida ?**

**RESULTADO:** Si son mostrados, sin embargo no se ha implementado la ordenación en la salida

## **V. CONCLUSIONES**

---

**9- ¿Si en la salida un registro excede el fin de página, este puede ser recorrido integro a la siguiente para evitar cortarlo ?**

**RESULTADO:** Si, esto funciona de manera automática

**10 - ¿Puede un campo ser recorrido a la página siguiente si este excede el fin de página ?**

**RESULTADO:** Si, esto funciona de manera automática

**11- ¿En la salida puede definirse el orden de los campos ?**

**RESULTADO:** Si, el formato es definido por el usuario.

**12 - ¿Pueden definirse formatos de salida y utilizar diferentes tipos de letras y tamaños de caracteres ?**

**RESULTADO:** Parcialmente, el usuario personaliza los formatos de salida, y puede agregar secuencias de control que definen tipos y tamaños de letras.

**13 - ¿Pueden definirse encabezado y pie de página ?**

**RESULTADO:** Si, pero deben ser indicados por el usuario en el formato de salida.

**14 - ¿Pueden almacenarse formatos de salida ? Varios ?**

**RESULTADO:** Si, el usuario almacena el formato de salida en archivos especiales.

## V. CONCLUSIONES

---

### Estrategias de búsqueda

1 - ¿Cual es el número máximo de términos y/o líneas de búsqueda en una estrategia almacenada ?

RESULTADO: 99 términos o 99 operaciones entre términos.

2 - ¿Puede modificarse una estrategia almacenada ? Los términos pueden borrarse, agregarse o modificarse ?

RESULTADO: No implementado.

3 - ¿Puede ser almacenada una estrategia de búsqueda en :

Intervalos especificados por el usuario ?

Intervalos definidos en el programa ?

A petición del usuario ?

Automáticamente cuando nuevos registros son agregados a la base de datos ?

RESULTADO: No implementado.

### Seguridad

1 - ¿Se tienen claves de acceso ?

RESULTADO: No aplica, ya que los datos no son alterables por el usuario.

## **V. CONCLUSIONES**

---

**2 - ¿El acceso es restringido a solo lectura ? Es posible acceder y modificar la información de la base de datos ?**

**RESULTADO** Dado que el CD-ROM es de sólo lectura los datos no pueden ser alterados.

**3 - ¿El usuario es notificado del número de accesos ? De su último acceso ?**

**RESULTADO:** No implementado.

**4 - ¿Pueden ser cambiados las claves ? Están encriptadas ?**

**RESULTADO:** No aplica.

### **Costo**

**1 - ¿El precio de la licencia de uso del programa permite múltiples copias ?**

**RESULTADO:** Como es un desarrollo a la medida, no existen restricciones en cuanto al número de licencias.

**2 - ¿Pueden adquirirse licencias limitadas a una comunidad de usuarios ?**

**RESULTADO:** No aplica.

**3 - ¿La licencia limita el uso de programa a una sola aplicación o puede ser utilizada en otras bases de datos de diferente estructura ?**

**RESULTADO:** Si, por el momento se requiere adaptar el programa a cada aplicación de manera específica, con relativa facilidad.

## V. CONCLUSIONES

---

### **Integración con otros programas**

**1 - ¿El programa puede utilizarse como parte de todo un sistema e interactuar con otros módulos del mismo ?**

**RESULTADO:** Si, debido a sus dimensiones puede ser ejecutado desde menús o bien desde otros programas.

**2 - ¿Pueden importarse datos de otros sistemas ?**

**RESULTADO:** No aplica, una vez generada la aplicación, los datos son de sólo lectura.

Como puede observarse, OSIRIS, resulta potencialmente adecuado tanto para esta aplicación, como para otras basadas en archivos invertidos. Comparado con otros sistemas comercialmente disponibles, se observan características equivalentes y en algunos casos hasta superiores.

Finalmente, se puede concluir que el programa OSIRIS, cubre y rebasa por mucho los requerimientos planteados.

**APÉNDICE A**

**MANUAL DE USUARIO**

El objetivo del presente sistema para la recuperación de información textual almacenada en CD-ROM, es efectuar búsquedas de información en una base de datos. Las cuales son realizadas por campo completo o palabras intermedias. Tiene la capacidad de establecer búsquedas relacionadas con operadores booleanos ("Y", "O" y "Y NO").

El producto final del sistema es manipular el conjunto de registros resultado de la búsqueda en diferentes formas, por lo que el sistema proporciona la facilidad de mostrar los datos localizados en una ventana, brinda la opción de grabarlos en un archivo y por último los utiliza en la impresión de cartas y etiquetas personalizadas.

La base de datos que utiliza este sistema es DIRFAX la cual, contiene un directorio nacional de empresas comerciales y se encuentra almacenada en CD-ROM.

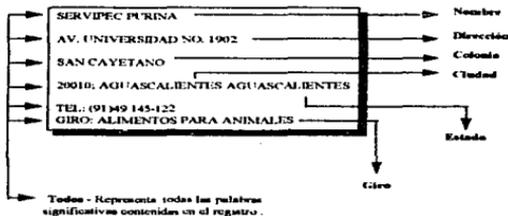


Figura 42: Descripción de un registro en la base de datos DIRFAX.

## APÉNDICE A

---

### I Cómo instalar el sistema?

La instalación se efectúa solo una vez para crear el ambiente propicio que permita operar con el sistema.

- Introduzca el CD-ROM en la unidad lectora de CD de la computadora.
- Desde el sistema operativo, teclee el nombre de la **unidad lectora de CD**, seguido de la palabra **Instalar**, presione la tecla (**Enter**).

D:\INSTALAR

- Si desea que el idioma interactivo, entre el sistema y usted, sea el inglés ejecute el programa **install** en lugar de **instalar**.

D:\INSTALL

- El programa de instalación crea un directorio llamado **DIRFAX** en el disco duro de la computadora, con los programas necesarios para iniciar el sistema, para lo cual se requiere que conteste las preguntas mostradas en pantalla, tecleando los datos de la configuración de la computadora, tales como nombre de la unidad de disco duro, unidad lectora de CD y directorio donde se copiarán los programas.

### II Cómo iniciar el sistema?

Al iniciar el sistema se presenta la pantalla principal con todas las opciones disponibles, ilustradas en la figura 43.

- En el directorio **DIRFAX** creado en el disco duro teclee el nombre del archivo de inicio **DF** seguido de la tecla **Enter**.

C:\DIRFAX>DF

## APÉNDICE A

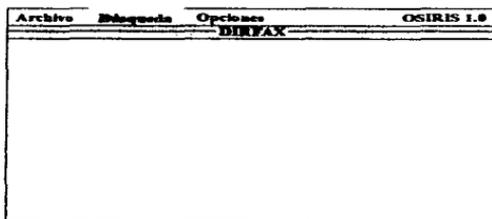


Figura 43: Menú principal del sistema.

### III Cómo desplazarse en los menús?

El sistema ofrece al usuario final características estandarizadas en el manejo de los menús, de tal forma que por medio de las teclas de movimiento del cursor, y (**Enter**), cualquier opción puede ser seleccionada, mostrando un submenú a manera de cascada.

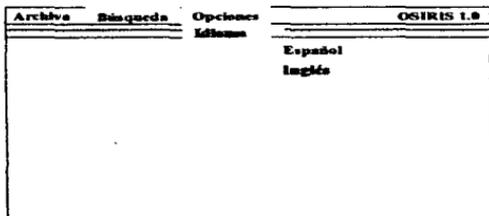
- En los menús horizontales se navega con las flechas <←→> por cada una de las opciones disponibles, que se encuentran a lo largo de la línea superior horizontal del marco, como se muestra en la figura 43.
- En los menús verticales se navega con las flechas <↑↓> por cada una de las opciones disponibles, que se encuentran a lo largo de la línea vertical en la ventana, como se muestra en la figura 47.

### IV Cómo seleccionar el Idioma?

El sistema presenta los menús en más de una lengua, permitiendo seleccionar entre el inglés y el español.

- Seleccione **Opciones** en el menú principal mostrado en la figura 44.
- En el submenú desplegado elija la opción **Idioma**.
- Seleccione la opción del idioma deseado, y oprima la tecla (**ESC**) sucesivamente hasta llegar al menú principal.

## APÉNDICE A



*Figura 44: Las pantallas se presentan en el idioma seleccionado.*

### ■ V Cómo salir del sistema?

- Seleccione en la línea superior del menú principal mostrado en la figura 43, la opción **Archivo**.
- Elija la opción **salir** en el submenú ilustrado en la figura 52.

## BÚSQUEDAS

### ■ VI Seleccionar la forma de localizar términos.

El sistema realiza búsquedas en diferentes formas, las cuales se describen ampliamente en la figura 45.

- Seleccione en el menú principal mostrado en la figura 43, la opción de **búsqueda**.
- Elija en el submenú de búsqueda mostrado en la figura 47, una opción que defina la forma de localizar términos en el sistema

Ejemplo:            **Seleccionar Todos**

## APÉNDICE A

OPCIONES	DESCRIPCIÓN
Todos	Establece una búsqueda por palabras intermedias, la cual consiste en seleccionar una llave y efectuar su búsqueda en todos los campos del registro (Nombre, Dirección, Colonia, Ciudad, Estado y Giro). La llave se selecciona en la ventana de términos que muestra un índice de todas las palabras contenidas en la base de datos.
Nombre	Establece una búsqueda por campo completo, la cual consiste en seleccionar una llave y efectuar su búsqueda únicamente en el campo Nombre del registro. La llave se selecciona en la ventana de términos que muestra un índice de los nombres (razón social) de las empresas contenidas en la base de datos.
Dirección	Establece una búsqueda por campo completo, la cual consiste en seleccionar una llave y efectuar su búsqueda únicamente en el campo Dirección del registro. La llave se selecciona en la ventana de términos que muestra un índice de la dirección en donde están ubicadas las empresas contenidas en la base de datos.
Colonia	Establece una búsqueda por campo completo, la cual consiste en seleccionar una llave y efectuar su búsqueda únicamente en el campo Colonia del registro. La llave se selecciona en la ventana de términos que muestra un índice de la Colonia en donde están ubicadas las empresas contenidas en la base de datos.
Ciudad	Establece una búsqueda por campo completo, la cual consiste en seleccionar una llave y efectuar su búsqueda únicamente en el campo Ciudad del registro. La llave se selecciona en la ventana de términos que muestra un índice de la Ciudad en donde están ubicadas las empresas contenidas en la base de datos.
Estado	Establece una búsqueda por campo completo, la cual consiste en seleccionar una llave y efectuar su búsqueda únicamente en el campo Estado del registro. La llave se selecciona en la ventana de términos que muestra un índice del Estado en donde están ubicadas las empresas contenidas en la base de datos.
Giro	Establece una búsqueda por campo completo, la cual consiste en seleccionar una llave y efectuar su búsqueda únicamente en el campo Giro del registro. La llave se selecciona en la ventana de términos que muestra un índice del Giro de las empresas contenidas en la base de datos.
Iniciar	Esta opción borra todos los resultados de búsqueda almacenados en la historia. Al seleccionarla desplaza el cursor al inicio del menú. - opción búsqueda - para que uno de los campos descritos anteriormente se seleccione y de comienzo una nueva búsqueda.

Figura 45: Descripción de las opciones de búsqueda utilizadas en la localización de términos.

## APÉNDICE A

---

### ■ VII Establecer una condición de búsqueda

La condición de búsqueda esta formada por el dato a localizar en la base de datos, en el caso de ser solo un término, como ejemplo: buscar todos los registros que contengan la palabra UNIVERSIDAD. En el caso de ser dos los términos a buscar, se relacionan con un operador booleano, como ejemplo: localizar todos los registros que contengan las palabras UNIVERSIDAD y AGUASCALIENTES.

- Seleccione en la ventana de términos mostrada en la figura 47, la llave (o término) que desea localizar en la base de datos.
- El sistema presenta dos formas de seleccionar uno de estos términos.
  - ♦ La primera es desplazar el cursor hasta que el término deseado se encuentre resaltado, y presionar la tecla <Enter> para seleccionarlo. Esto se efectúa al navegar en la ventana de términos utilizando las teclas descritas en la figura 46.
  - ♦ La segunda forma y mas rápida es teclear el término deseado en la posición del cursor, seguida del mensaje "LOCALIZAR POR" como se indica en la figura 47, y presionar la tecla <Enter> para seleccionarlo.

Ejemplo: **LOCALIZAR POR: UNIVERSIDAD**

TECLAS	MÓVIMIENTO DEL CURSOR
Con las flechas <←→>	Se desplaza una línea hacia arriba y un línea hacia abajo respectivamente.
Av. Pág.	Avanza una página. En la ventana se muestra la información de la página siguiente.
Ret. Pág.	Retrocede una página. En la ventana se exhibe la información de la página anterior.
ESC	Cierra la ventana actual y muestra la ventana anterior.

*Figura 46: Descripción de las teclas para navegar en la ventana de términos.*

Esto dará comienzo a la búsqueda de todos los registros contenidos en la base de datos que cumplan con la condición indicada. Al terminar, en la parte inferior de la pantalla se muestra el número de registros localizados (aciertos última búsqueda) como se ilustra en la figura 47.

## APÉNDICE A

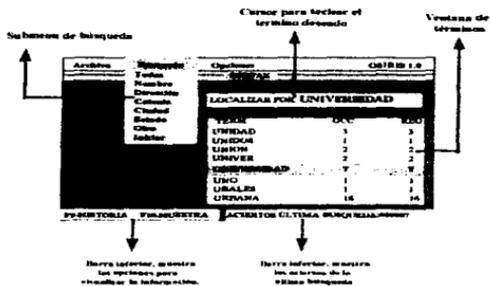


Figura 47: Establecimiento de condiciones para una búsqueda.

### ■ VIII Efectuar una búsqueda sencilla o compleja

#### • VIII.1 Búsqueda sencilla

Consiste en localizar registros que cumplan con una condición de búsqueda.

- ◆ Efectúe las indicaciones de los incisos VI y VII.

#### • VIII.2 Búsqueda compleja

Consiste en localizar registros que cumplan con dos condiciones relacionadas con un operador booleano como ("Y", "O" y "Y NO").

- ◆ Efectúe las instrucciones indicadas en los incisos VI y VII al finalizar, el grupo de registros localizados (aciertos), quedan grabados en la historia formando la primer condición.

Ejemplo: Seleccionar **UNIVERSIDAD** en la ventana de términos.

- ◆ Establezca una segunda condición efectuando una vez mas, las instrucciones indicadas en los incisos VI y VII.

Ejemplo: Seleccionar **AGUASCALIENTES** en la ventana de términos.

## APÉNDICE A

- ◆ Elija en el menú de operadores booleanos el que relacionará las condiciones que se han establecido

Ejemplo: seleccionar el operador booleano AND/Y

La búsqueda comenzará y al finalizar se obtendrá como resultado los registros que contienen las palabras UNIVERSIDAD y AGUASCALIENTES.

(UNIVERSIDAD) AND (AGUASCALIENTES)

TECLAS	MOVIMIENTO DEL CURSOR
Av. Pág.	Muestra el siguiente registro del conjunto resultado de la búsqueda.
Ret. Pág.	Exhibe el registro anterior del conjunto resultado de la búsqueda.
ESC	Cierra la ventana y muestra la lista de terminos.
F8	Muestra la imagen ligada al registro, si este indica que la contiene.

Figura 48: Descripción de las teclas para navegar en la ventana de aciertos.

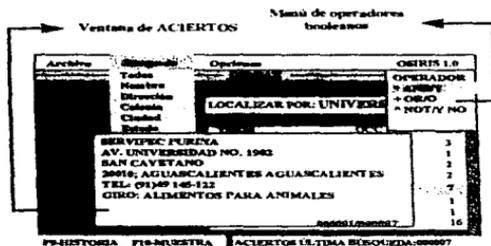


Figura 49: Ventana de aciertos, muestra los registros localizados

## APÉNDICE A

---

### IX Efectuar búsquedas sucesivas

#### • Búsqueda sencilla sucesiva

- ◆ Seleccione la opción **Iniciar** en el submenú de búsqueda mostrado en la figura 47. Esto borra la **historia**, indicando que la siguiente condición que se establezca, no se aplicará al grupo de datos almacenados en la historia
- ◆ Efectúe las indicaciones de los incisos **VI,VII y VIII.1**

#### • Búsqueda compleja sucesiva

El último conjunto de aciertos resultado de cada búsqueda, es utilizado como grupo de datos al que se aplicará la siguiente condición, siempre y cuando no se borre la historia con la opción **Iniciar**.

- ◆ Efectúe las indicaciones de los incisos **VI,VII y VIII.2** sucesivamente hasta conformar la búsqueda deseada

### X Cómo mostrar resultados?

Los mensajes que se presentan, en la barra de estado ubicada en la parte inferior de la pantalla que se ilustra en la figura 49, se utilizan para ayudar al usuario a visualizar los datos localizados. La figura 50 presenta una descripción detallada de estos mensajes.

#### • Visualizar los datos de la historia.

- ◆ Presione la tecla <F9>, si la barra de estado presenta esta opción, como se muestra en la figura 51 .

#### • Mostrar los registros localizados (aciertos).

La forma de navegar en la ventana de aciertos se describe en la figura 48.

- ◆ Presione la tecla <F10>, si la barra de estado presenta esta opción, como se muestra en la figura 49.

## APÉNDICE A

TECLA	DESCRIPCIÓN
F9	Presenta la historia de las búsquedas efectuadas hasta el momento. La historia muestra la siguiente información: SET = número consecutivo HITS = número de aciertos en la búsqueda OP = operador booleano utilizado KEY = condición de búsqueda
F10	Se visualizan en la ventana de aciertos, los registros localizados en una búsqueda.

Figura 50: Descripción de las teclas para mostrar los registros localizados en una búsqueda.

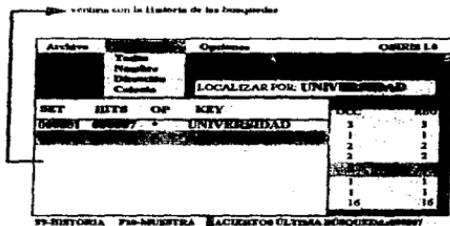


Figura 51: Historia de las búsquedas que se han efectuado.

### ■ XI Cómo imprimir resultados?

El sistema tiene la capacidad de imprimir el conjunto de registros resultado de las búsquedas, en tres formatos, cartas, etiquetas y reportes.

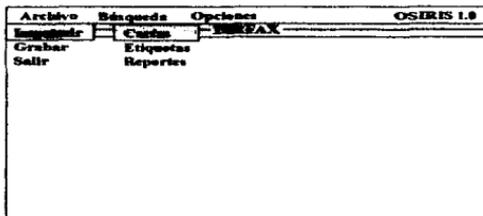
Para ejemplificar el procedimiento de impresión, se seleccionó el de cartas, debido a que la forma de imprimir es similar para todos, y los tres se basan en el concepto de documento personalizado. Esto es, con la elaboración de un documento maestro el sistema imprime un reporte por cada registro localizado en la búsqueda, lo cual es útil cuando se desea enviar documentos con la misma información a diferentes destinatarios, donde los datos propios de cada destinatario son la única variante en el documento.

## APÉNDICE A

---

- **Como seleccionar el documento cartas?**

- ◆ Seleccione la opción **Archivo** en el menú principal mostrado en la figura 43
- ◆ Elija la opción **Imprimir** en el submenú ilustrado en la figura 52
- ◆ En submenú mostrado en la figura 52, elija la opción **cartas**.



*Figura 52: Menú de impresión de resultados.*

- **Cómo escribir una carta?**

En la figura 53 se muestra el formato para editar cartas, en la parte inferior de la pantalla se visualiza un menú con diferentes opciones que indican como desplazarse en el documento.

- ◆ Comience a escribir la carta maestra en el formato mostrado en la figura 53.
- ◆ Si se desea incluir en la carta, campos de la base de datos, mueva el cursor hasta que esté ubicado en la posición donde se requiere insertar el campo, oprima la tecla **F2** para exhibir la ventana con los campos disponibles, como se muestra en la figura 53 -Nombre, dirección, colonia, etc.-. Seleccione el campo que desea incluir.

## APÉNDICE A

1	EDICIÓN DEL CUERPO DE LA CARTA		OSIRIS 1.0
2	México, D.F., a 15 de Mayo de 1996		
3	[NOMBRE]		
4	[DIRECCIÓN] [COLONIA]		
5	[CIUDAD] [ESTADO]		
6			
7			
8			
9	Estimado cliente nos permitimos hacer de su conocimiento		
10	que su empresa dedicada a [GIRO] ha sido dada de alta en		
11	la base de datos Dirfax.		
12			
13			
14			
15			
16			
17			
18			
19			
INGRESE EL TEXTO DE SU CARTA		NOMBRE	
<PgDown> = Salir <ESC> = Aborta		DIRECCION	
		COLONIA	
		CODIGO	
		CIUDAD	
		ESTADO	
		GIRO	

Figura 53: Edición de una carta maestra.

### • Cómo salvar la carta?

- ◆ Al finalizar la redacción de la carta, oprima la tecla <ESC>, esto presentará en la parte inferior de la pantalla las opciones para revisar y salvar el documento, como se muestra en la figura 54.
- ◆ Seleccione la opción N o B seguida de la tecla <Enter> si desea revisar el documento.
- ◆ Seleccione la opción C seguida de la tecla <Enter> si desea regresar a editar la carta maestra.
- ◆ Seleccione la opción X seguida de la tecla <Enter> si desea salvar el documento.

1	EDICIÓN DEL CUERPO DE LA CARTA		OSIRIS 1.0
2	México, D.F., a 15 de Mayo de 1996		
3	[NOMBRE]		
4	[DIRECCIÓN] [COLONIA]		
5	[CIUDAD] [ESTADO]		
6			
7			
8			
9	Estimado cliente nos permitimos hacer de su conocimiento		
10	que su empresa dedicada a [GIRO] ha sido dada de alta en		
11	la base de datos Dirfax.		
12			
13			
14			
15			
16			
17			
18			
19			
Seleccione una de las opciones : X			
N = Pg.Sic. B = Pg.Ant. X = Salir v. Salvar. C = Cancelar			

Figura 54: Salvar, cancelar y revisar una carta maestra.

## APÉNDICE A

EDICION DEL CUERPO DE LA CARTA OSIRIS 1.0

Nombre de Archivo: (8caracteres máx.)

ARCHIVO EXISTENTE...SOBREScribir? ( S/N) N

*Figura 55: Ventana para proporcionar un nombre de archivo para almacenar la carta*

- ◆ Proporcione el nombre del archivo donde se salvará la carta maestra, como se muestra en la figura 55. El sistema verifica si el archivo existe en el directorio y proporciona la opción de sobrescribir o especificar un nombre diferente.

EDICION DEL CUERPO DE LA CARTA OSIRIS 1.0

Nombre de Archivo: (8caracteres máx.)

SELECCIONE EL CONJUNTO A IMPRIMIR

RET	IDTS	OP	KFV
000001	000007	=	UNIVERSIDAD
000002	000006	=	AGUASCALIENTES

*Figura 56: Selección del conjunto de datos que será impreso en las cartas.*

Elija en la pantalla de historia de búsquedas mostrada en la figura 56, el conjunto de resultados del cual se tomarán los datos, para sustituirlos en cada carta que se genere

## **APÉNDICE B**

### **PC MULTIMEDIA**

## APÉNDICE B

REQUERIMIENTOS	MPC nivel 1	MPC nivel 2	MPC nivel 3
<b>Procesador</b>	60 mhz 386 SX o equivalente	25 mhz SX o equivalente	75 mhz Pentium binario compatible del mismo nivel o superior
<b>RAM</b>	2 MB	4 MB (recomendable 6 MB)	8 MB mínimo
<b>DISCO DURO</b>	30 MB	160 MB	540 MB
<b>Unidad de CD-ROM</b>	Velocidad de transferencia de 150 Kbps y un tiempo promedio máximo de búsqueda de 1 seg	Velocidad de transferencia de 300 Kbps y un tiempo promedio mínimo de búsqueda de 400 ms. CD-ROM XA compatible Multi-session	Velocidad de transferencia de 600 Kbps y un tiempo promedio de acceso de 250 ms. CD-ROM XA compatible multisesión
<b>Sonido</b>	8 bit sonido digital 8 tonos de sintetizador reproductor MIDI	16 bit de sonido digital 8 tonos de sintetizador reproductor MIDI	16 bit de sonido digital reproductor MIDI Bocinas mínimo 3 wats/canal
<b>Video de despliegue (monitor)</b>	640 X 480, 16 colores se recomienda: 640 X 480, 256 colores	640 X 480, 65, 536 colores. Se recomienda que proporcione 1.2 megapixels/seg. Con un 40% de ancho de banda en el CPU.	Conversión a espacios de color y capacidad de escalamiento. Acceso directo a la memoria del recuadro para habilitar el subsistema gráfico con una resolución de 352x240 a 30 fps (352x288 a 25 fps) cerca de 15 bits/pixel
<b>Video reproductor</b>	No adicionado	No adicionado	Tarjeta desdificionada de formato PMG. Acceso directo a cuadros con una resolución de 352x240 a 30 fps (352x288 a 25 fps) a 15 bits/pixel. Todos los codecs (hardware y software) deben reproducir video de audio sincronizado a una resolución de 352x240 a 30 fps (352x288 a 25 fps) cerca de 15 bits/pixel
<b>Puertos I/O</b>	MIDI, serial, paralelo, joystick	MIDI, serial, paralelo, joystick	MIDI, serial, paralelo, joystick
<b>Teclado</b>	101 teclas IBM mouse con 2 botones	101 teclas IBM mouse con 2 botones	101 teclas IBM mouse con 2 botones
<b>Software del sistema</b>	Windows 3.0 plus Multimedia	Windows 3.0 plus Multimedia	Windows 3.11 y DOS 6.0

Figura 57: Especificaciones de la PC MULTIMEDIA.

## **APÉNDICE C**

## **GLOSARIO**

## APÉNDICE C

---

100BaseVG	Es un protocolo de red utilizado por Novell*, Microsoft, Hewlett Packard, AT&T y otros, para la transmisión de datos a 100 Mbits/seg. Usando cuatro hilos de cable categoría 3. El comité IEEE 802.12 es responsable de este desarrollo.
ATM	Del inglés Asynchronous Transfer Mode, es una tecnología para la transmisión de datos aplicable a redes LAN y WAN. Transporta diferentes tipos de tráfico, incluyendo voz, datos, fax, video, audio CD. Las implementaciones actuales operan con velocidades del orden de 155 Mbits/seg. a 622 Mbits/seg. sobre fibra óptica.
BMP	Del inglés Bit Map. Se refiere a un formato para el almacenamiento de imágenes digitalizadas.
BOEING	Es una Compañía de aviones.
CACHE	Es una forma de acceder información, almacenándola en memoria principal en vez de direccionarla a un disco. Con esto se disminuye considerablemente el tiempo de recuperación de los datos.
CCE (del inglés ECC)	Código de Corrección de Errores, del inglés Error Correction Code. Es un software que reside dentro de un firmware del CD-ROM que detecta errores en la lectura de un disco.
CD	Del inglés Compact Disc, disco compacto. Es una oblea de plástico de policarbonato transparente que mide 120 mm de diámetro, 1.2 mm de espesor y tiene en el centro una horadación de 15 mm. Esta cubierto con una capa de aluminio reflectante y una de resina, almacena información digitalizada en una pista en espiral.
CD-DA	Del inglés Compact Disc Digital Audio. Es un disco

## APÉNDICE C

---

	compacto de audio digital basado, en el estándar del libro rojo.
CD-DVI o CD-I	Del inglés Compact Disc - Digital Audio Interactive. Es un disco compacto de video digital interactivo, basado en el estándar del libro verde. Está diseñado para reproducir video en movimiento interactivo en un dispositivo Philips CD-I conectado a un televisor.
CD-MO CD-R	Del inglés Compact Disc Recordable, es utilizado como medio de grabación en el proceso de premasterización.
CD-ROM	Del inglés Compact Disc Read Only Memory, memoria de sólo lectura en disco compacto. Esta basado en el estándar del libro amarillo.
CD-ROM XA	Del inglés CD-ROM Extended Architecture, disco compacto de arquitectura extendida. Esta basado en el estándar extendido del libro amarillo el cual, define un formato nuevo para grabar información como audio comprimido y video/imagen, para ser leída y reproducida al mismo tiempo. Se requiere un equipo reproductor CD-ROM/XA para reproducir un disco.
CD-WORM o WO	Del inglés Write One Read Many, es un disco óptico de una sola escritura. Es multisesión debido a que se pueden emplear sesiones múltiples para llenar el disco. Se basa en el estándar del libro naranja y un reproductor de CD normal puede leer este disco.
CDE (del inglés EDC)	Código de Detección de Errores, del inglés Error Detection Code. Es un software que reside dentro de un firmware del CD-ROM que corrige errores detectados en la lectura de un disco.
CIRC	Del inglés Cross Interleave Reed-Solomon Code, es

## APÉNDICE C

---

CODEC	un código de corrección de errores desarrollado específicamente para CD-ROM Del inglés COMpression/DECompression, los productores de CD-ROM usan codecs para comprimir video. Los codecs manejan descompresión y despliegue desde una aplicación de multimedia.
COMPENDEX	Base de datos relativa de Ingeniería producida por Engineering Information Inc.
COMPUTER SELECT	Base de datos en el campo de la computación producida por la compañía Ziff Daviss.
CORE	Del inglés Controlled Requirements Expressions. Es una metodología para el análisis de sistemas.
DAT	Del inglés Digital Audio Tape, cinta magnética. Formato magnético para el almacenamiento de datos digitalizados.
DIALEX I	Banco de datos que contiene los índices del "Diario Oficial de la Federación" producido por el Archivo General de la Nación.
DIALOG INFORMATION SERVICES	Productor y distribuidor de Bancos de Información.
DIFUSIÓN CIENTÍFICA CD-ROM	Compañía Mexicana dedicada a la producción y distribución de Bancos de Datos.
DIRFAX	Nombre de un directorio nacional de empresas comerciales almacenado en CD-ROM.
DOS	Del inglés Disc Operating System, sistema operativo de disco.
DVD	Del inglés Digital Video Disc, disco compacto para el almacenamiento de video digital.
EDICIONES EL GLOBO	Nombre de la empresa propietaria de la base de datos

## APÉNDICE C

---

- Dirfax.
- EFM** Del inglés Eight-to-Fourteen Modulation, es un proceso realizado en el drive del CD-ROM por un decodificador que convierte los bytes de datos de 8 bits a bytes de datos de 14 bits.
- ESTÁNDAR LIBRO AMARILLO** Define el formato estándar básico para el CD-ROM, fue desarrollado por Philips y Sony.
- ESTÁNDAR LIBRO NARANJA** Define el formato estándar para los CD-ROM de una sola lectura (WO o WORM, CD-R) y para los reescribibles (CD-MO).
- ESTÁNDAR LIBRO ROJO** Define el formato estándar de audio digital, desarrollado por Philips y Sony para el CD-DA. Fue nombrado así por el color de la sobrecubierta del documento publicado en 1986.
- ESTÁNDAR LIBRO VERDE** Define el formato estándar establecido por Philips para el CD-I. Especifica un formato de arquitectura extendida y el sistema operativo (CD-RTOS) para ambiente propietario CD-I de Philips. Fue publicado en 1988.
- FAST-ETHERNET o 100 BASE X** Fue propuesto por 3Com, Intel y Synoptics. Es un método de transmisión de datos a 100 Mbits/seg. sobre cable sin blindaje (UTP) de categoría 5. El IEEE 802.3 es responsable de este proyecto.
- FAT** Del inglés File Allocation Table, tabla que define archivos y directorios en diversos medios de almacenamiento de datos.
- FDDI** Del inglés Fiber Distributed Data Interface, método de transmisión de datos a 100 Mbits/seg. sobre fibra óptica basada en Token Ring.

## **APÉNDICE C**

---

<b>FIRMWARE</b>	Es un código que se almacena dentro de un circuito integrado Normalmente se utiliza un EPROM.
<b>HDCD</b>	Del inglés High Density Compact Disc, disco compacto de alta densidad.
<b>HFS</b>	Del inglés Hierarchical File System , sistema jerárquico de archivos. Es un método para administrar archivos y carpetas en equipo Apple Macintosh.
<b>HIGH SIERRA</b>	Es un estándar que define una estructura común de archivos para CDs, dio origen al estándar ISO9660. El nombre lo tomó del hotel y casino del Webb's High Sierra en Nevada, donde se efectuó una junta de representantes industriales que establecieron este formato.
<b>HITACHI</b>	Compañía Multinacional de la Industria Electrónica, fabricante de partes para equipo de Cómputo y Multimedia.
<b>IDE o AT</b>	Del inglés Integrated Drive Electronics, es una interfaz para disco duro y CD-ROM.
<b>IRS</b>	Del inglés Information Retrieval System, sistema recuperador de información.
<b>ISBN</b>	Del inglés International Standard Book Number, número nternacional de libro estandarizado.
<b>ISO9660</b>	Es el estándar utilizado para la producción de CD-ROM, fue aprobado por la Organización Internacional de Estándares (ISO).
<b>KWAC</b>	Del inglés Key Word And Context, llave y contexto.

## APÉNDICE C

---

KWIC	Del inglés Key Word In Context, llave en contexto.
MAC	Contracción de la marca Macintosh propiedad de Apple Computer Inc.
MEDLINE	Banco de datos en el área de medicina producido por la Biblioteca Nacional de Medicina de los Estados Unidos.
MICRO CDS/ISIS	Sistema recuperador de información producido por la UNESCO.
MIDI	Del inglés Musical Instrument Digital Interface, interfaz digital de instrumento musical. Es una norma para tocar música en una PC.
MPC	Siglas de Multimedia Personal Computer. Es un estándar creado en el mercado de Multimedia, que establece los requerimientos mínimos aceptables para correr aplicaciones de multimedia en Pcs basadas en windows.
MPEG1	Del inglés Moving Picture Express Group, es un esquema de compresión popular para video.
MS	Siglas de la compañía Microsoft.
MTBFs	Del inglés Mean Time Before Failure, tiempo promedio de operación antes de falla.
MULTIMEDIA	Es cualquier combinación de información (datos, texto, arte gráfico, sonido, animación y video) digitalizados.
NFS	Del inglés Network File System, es un sistema de archivos distribuidos desarrollado por Sun Microsystems, altamente usado en sistemas TCP/IP.

## **APÉNDICE C**

---

<b>NIMBUS Information Systems</b>	Nombre de una compañía dedicada al desarrollo de software para el manejo de datos en medios electrónicos.
<b>NLM</b>	Del inglés Network Loadable Module. Tecnología en donde un programa se carga como parte del Sistema Operativo de red (Novell Netware).
<b>NOVELL NETWARE</b>	Componentes vendidos por Novell para una red (una colección de drivers ligadores de datos, un protocolo de transporte, el software para la estación de trabajo y el sistema operativo NetWare).
<b>OEM</b>	Por sus siglas en inglés Original Equipment Manufacturers, es una empresa ensambladora de unidades de CD-ROM.
<b>OPA</b>	Por sus siglas en inglés Optical Publishing Association, es una Asociación de publicaciones ópticas.
<b>OSIRIS</b>	Nombre del sistema recuperador de información almacenado en CD-ROM, desarrollado en este trabajo de tesis.
<b>PCX</b>	Formato para el manejo de imágenes digitalizadas.
<b>PHILIPS</b>	Compañía Multinacional que en unión con Sony desarrollaron las bases de la tecnología del Disco Compacto.
<b>PHOTO-CD</b>	Es un disco óptico de una sola escritura. Contiene imágenes digitalizadas de alta resolución, de fotografías originales (negativos). Es multisesión.

## APÉNDICE C

---

debido a que puede emplear sesiones múltiples para llenar el disco. Las imágenes del CD pueden verse utilizando un reproductor Kodak PHOTO-CD conectado a una televisión o empleando un reproductor CD-ROM/XA conectado a una computadora.

### RAM

Del inglés Random Access Memory, memoria dinámica algunas veces conocida como memoria principal.

### ROMWARE

Nombre de un software recuperador de información, desarrollado por la Compañía Nimbus Information Systems.

### SCSI

Por sus siglas en inglés Small Computer System Interface, interfaz de sistemas para computadoras pequeñas. Es un bus estándar de alta velocidad utilizado generalmente para conectar drives CD-ROM a computadoras.

### SILVER PLATTER

Productor del software recuperador de información SPIRS y distribuidor de bancos de datos en Disco Compacto e Internet.

### SONOPRESS México 1995

Compañía Mexicana dedicada a la replicación masiva de discos compactos.

### SONY

Compañía Japonesa dedicada a la electrónica que en asociación con Philips desarrollaron la tecnología del disco compacto.

### SPIRS

Software recuperador de información desarrollado por la compañía Silver Platter.

### SSAD

Del inglés Structured Systems Analysis and Design, metodología para el diseño y análisis estructurado de

## APÉNDICE C

---

	sistemas
TBASE-	Librerías de Clipper usadas para el desplegado de imágenes.
TCP/IP	Del inglés Transmission Control Protocol/Internet Protocol, protocolo de arquitectura de red no propietario, ampliamente usado en UNIX.
TEAC	Compañía fabricante de equipo electrónico.
TECNOLOGÍA ÓPTICA	Se utiliza en los medios de almacenamiento óptico (CD-ROMs). Mientras el disco gira en la unidad de CD, el metal refleja luz desde un pequeño láser infrarrojo en un receptor de diodo sensible a la luz. Estos reflejos se transforman en una señal eléctrica y después se convierten en bits y bytes significativos para utilizarse en equipo digital. La reproducción o lectura de un disco óptico, se basa en los cambios de la reflectividad.
TECNOLOGÍA ÓPTICA-MAGNÉTICA	Se utiliza en los medios de almacenamiento óptico reescribible (CD-MO). Para efectos de grabación se calienta la superficie del disco, para cambiar sus condiciones permitiendo que después una cabeza magnética grabe información con mayor densidad. La reproducción o lectura de un disco magneto óptico se basa en la dirección de polarización del rayo láser reflejado.
TEXTWARE LITE	Nombre de un software recuperador de información desarrollado, por la compañía del mismo nombre.
TIFF	Del inglés Tagged Interchange File Format, formato de archivo de imágenes exploradas. Es uno de los más utilizados en Windows se diseñó, para ser el formato de imágenes de mapas de bits universal.
TOKEN RING	Tecnología para la transmisión de datos en redes de computadoras en donde, los nodos reciben y

## **APÉNDICE C**

---

	transmiten los paquetes de datos en una estructura de anillo.
<b>TOPIX</b>	Software para la grabación de discos compactos escribibles.
<b>TOSHIBA</b>	Compañía fabricante de equipo electrónico.
<b>UNESCO</b>	Organización de las Naciones Unidas.
<b>UNIX</b>	Sistema operativo desarrollado por AT&T y licenciado para la mayoría de las plataformas de cómputo existentes actualmente.
<b>WARNER</b>	Compañía dedicada a los negocios del entretenimiento.
<b>XBASE</b>	Formato de archivos para DBASE, CLIPPER, etc.
<b>ZIFF DAVIS ELECTRONIC</b>	Compañía editora especializada en la tecnología electrónica.

## **APÉNDICE D**

### **PROGRAMAS**

```

/*
* PROGRAMA: INSTALAR.PRG
* AUTORES: BERTHA CAMPOS M.
          DAVID DEL MORAL C.
* DESCRIPCION: Instala los programas y archivos asociados de DIRFAX
               en CD-ROM a un directorio especifico en disco duro
               revisa la existencia de memoria expandida y crea un disco
               virtual en memoria. Graba parametros de instalacion en
               un archivo de variables en disco (DIRFAX.MEM)
*/

```

```

CLEAR
PUBLIC SCROT, I,KB,SN,OP,VCD,VHD,VTM,SCROB
PUBLIC FF
VHD="C:"                && Unidad de Disco Duro
VCD="D:"                && Unidad de CD-ROM
VTM="C:\TEMP\"         && Path Temporales
VLN=1                   && Idioma seleccionado 1= ESPAÑOL 2 = INGLES
SN="S"                  &&

```

```

/* ASIGNA AYUDA A LA TECLA F1 */
SET KEY ZB TO HELPP
DO WHILE .T.

```

```

  CLEAR
  DO CLEAR
  @ 0,3 SAY "INSTALACION DE DIRFAX"
  @ 1,0 SAY REPLICATE(" ",79)
  DO NORMAL
  @ 2,0 CLEAR
  TEXT
  DIRFAX EN CD-ROM

```

Este programa le permitirá instalar el programa de consulta al directorio comercial y de Servicios DIRFAX(MR) en su disco duro. Asimismo, Ud. puede configurar la instalación de acuerdo a la capacidad de su equipo con la finalidad de obtener el máximo rendimiento posible del sistema.

Los requerimientos de equipo son:

MIMIMOS:	RECOMENDABLES:
- Microcomputadora PC Compatible	80486 o superior
- 640 KB de memoria RAM	4 MB (200 KB Mem. Expandida)
- Disco duro	Vel. Acceso 14 ms o menos.
- CD-ROM	Doble velocidad o superior

Oprima cualquier tecla para iniciar la instalación...

```

ENDTEXT
KB=IKEY(0)
@ 4,0 CLEAR
@ 5,3 SAY "PARAMETROS DE INSTALACION:"
@ 7,5 SAY "Unidad de CD-ROM : " GET VCD
@ 9,5 SAY "Unidad de arranque: " GET VHD
@ 10,5 SAY "Ingrese por favor los valores solicitados."
READ
TEXT

```

Para un optimo funcionamiento del programa, es recomendable crear un disco virtual en memoria para crear archivos temporales. Desea Ud. configurar su equipo de esta forma ?.

```

ENDTEXT
@ 16,5 SAY "(S/M) : " GET SM
READ
IF UPPER(SM)="SM"
    VTM="D:\TEMP\"
    && Path Temporales
    @18,5 SAY "Ruta de archivos temporales en DISCO VIRTUAL: " GET VTM
    READ
ENDIF
@ 4,0 CLEAR
@ 5,3 SAY "Los parámetros de instalación proporcionados son:"
@ 7,5 SAY "Unidad de CD-ROM : " + VCD
@ 9,5 SAY "Unidad de arranque: " + VHD
@11,5 SAY "Ruta de archivos temporales en DISCO VIRTUAL: " + VTM
SM="SM"
@ 23,3 SAY "Los datos estan correctos y desea salvarlos (S/M)" get SM
READ
IF UPPER(SM)="S"
    EXIT
ENDIF
ENDDO
SAVE ALL LIKE V* TO DIRFAX.MEM

RETURN

PROCEDURE NORMAL
    SET COLOR TO M+/W+,W+/N,N,1+/7,W+/M
RETURN
PROCEDURE CIAN
    SET COLOR TO 3/W+,W+/3,N,1+/7,W+/2+
RETURN

PROCEDURE HELPP
LOCAL SCROB
SCROB=SAVESCREEN()
DO CIAN
@ 3,3,21,70 BOX " [4-] "
DO NORMAL
@ 4,4 CLEAR TO 10,69
@ 6,5 SAY "OPRIMA: "
@ 7,5 SAY " F1 - Muestra esta ayuda "
@ 8,5 SAY " F2 - Muestra nombres de campos "
@ 9,5 SAY " F10 - Graba el texto a un archivo en disco "
@11,5 SAY "DESPLAZAMIENTO: "
@12,5 SAY " Flechas < - > = Avanza/Retrocede una posición"
@13,5 SAY " <Ctrl> y <- -> = Avanza/Retrocede una palabra "
@14,5 SAY " <Pgdn> = Avanza una página "
@15,5 SAY " <PgUp> = Retrocede una página "
@17,5 SAY "PERSONALIZACIÓN: "
@18,5 SAY "Los campos deben ser precedidos por el caracter @ para "
@19,5 SAY "que sean considerados en la impresión "
@20,5
WAIT " "
RESTORE SCREEN FROM SCROB
RETURN

```

```

#INCLUDE "FILEIO.CH"
/* #INCLUDE "DBSTRUCT.H"*/
*
* DIRFAX: PROGRAMA DE CONSULTA A LA BASE DE DATOS DE DIRFAX PARA DOS
* AUTOR: BERTHA CAMPUS M.
* DAVID DEL MORAL C.
* ULTIMA REVISION: 19950224
* COMPILADOR: CLIPPERS.2
* COMENTARIOS: VEPSTON CONFIGURABLE A CD-ROM O DISCO DURO

&& DECLARACION DE VARIABLES PUBLICAS
LOCAL SCRO9 && Almacenamiento de pantallas
LOCAL PMSG && Apuntador a la tabla de mensajes
LOCAL BMSG && Buffer para carga de mensajes
LOCAL IMG5 && Indice de la tabla de mensajes
PUBLIC MSGS(256) && Tabla de mensajes
PUBLIC KB,CV,1,HIDX,MNO,AP,OFSET,TMO,TMT,DG,FF,FN
PUBLIC BTEXT0 && Variable de almacenamiento de formato de cartas
PUBLIC T11,T1F && Uso general
PUBLIC L1,LF && Limites de los archivos de indices
PUBLIC RW,CL && Renglon y columna
PUBLIC ITX,ITF && Indices de la Tabla de campos
PUBLIC CKE,LK6,MKE,LET,EXI,LIA,UDA,CIA,CDA && Variables utilizadas en SHTAG y ETIQUETAS
PUBLIC OP,SW,VLN,VD,VCD,VTM
PUBLIC VOO && Var.dispositivo de salida 1 screen 2 printer 3 file
PUBLIC OFN && Var.archivo de salida
PUBLIC FP && Apuntador al archivo de salida
PUBLIC NORMAL,PARPADED,CIAN
PUBLIC DFILE,_HIFILE,_KEYFILE,_INFFILE
PUBLIC FC(256) && Contenido de la base de datos
PUBLIC FV(256) && Tabla que contiene la estructura de la base de datos
PUBLIC MESES(12) && Meses del año en letra
PUBLIC HIST(99) && Almacena historia de búsquedas, Max: 99
PUBLIC IMGFILE && Toma el nombre de archivo de gráficos
PUBLIC TB_RETURN && Retorno de resultados llamadas a gráficos
PUBLIC TB_PARAMS && Parámetros de las funciones gráficas
PUBLIC KOFCARD && Tipo de adaptador de video
PUBLIC CMIPM && Tecnología de video
PUBLIC TEXT0(64) && Texto de cartas personalizadas
PUBLIC PTABLE(64) && Texto de cartas personalizadas macrosustituido
PUBLIC SM && Número de conjunto

RESTORE FROM DIRFAX.MEM ADDITIVE && Carga parámetros de instalación
DECLARE TBL1(4096) && Buffer de peso de apuntadores
DECLARE TBL2(4096)
DECLARE TBL3(4096)
DECLARE TBL4(4096)
DECLARE TBL5(4096)
DECLARE TBL6(4096)
DECLARE TBL7(4096)
DECLARE TBL8(4096)
DECLARE TBL9(4096)
DECLARE TBL0(4096)
DECLARE BUFF(4096)
DECLARE LNGT(3) && Longitud de submenús
DECLARE MENU(3) && Menú principal
DECLARE FILAS(3) && Definición de coordenadas de menús
DECLARE COLS(3)
DECLARE SUBMENU1(3) && Menú de Archivo
DECLARE SUBMENU2(3) && Menú de búsqueda
DECLARE SUBMENU3(1) && Menú de opciones
DECLARE POPUPS(3)

DECLARE PRINTM(3) && Menú de impresion

```

```

/* Inicialización de variables y declaración de constantes */
CR=13          && Car.Return
FF=CHR(12)    && Form Feed
NNO=0         && Numero de hits
VLN=2        && Idioma por default
SW=1
HIDe=0       && Apuntador a tabla de Hits (Aciertos)
Op=***
OB=***       && Operador booleano (para efecto de despliegue)
PARPADEO="3/4, W+/3, N, 1+/7, W+/2="      && Video parpadeando
NORMAL    ="N+/W, W+/N, N, 1+/7, W+/N"     && Video normal
CIAN     ="3+/W, W+/3, N, 1+/7, W+/2="     && Color
TI1=1    && Indices de la tabla de texto para cartas
TIF=1B

FILAS(1)=0          && Definición de coordenadas de menús
FILAS(2)=0
FILAS(3)=0

COLS(1)=1
COLS(2)=21
COLS(3)=41

LNGT(1)=3
LNGT(2)=8
LNGT(3)=1

&&SUBMENU3(2)=MSG3(25)      && Color

&& Tabla de meses
MESES(1) = "Enero"
MESES(2) = "Febrero"
MESES(3) = "Marzo"
MESES(4) = "Abril"
MESES(5) = "Mayo"
MESES(6) = "Junio"
MESES(7) = "Julio"
MESES(8) = "Agosto"
MESES(9) = "Septiembre"
MESES(10) = "Octubre"
MESES(11) = "Noviembre"
MESES(12) = "Diciembre"

DBFILE= VCD+"DIRFAX"      // Define nombres y rutas de archivos //
HITFILE=VTM+"SHIST"
KEYFILE=VCD+"KEYLST"
LNKFILE=VCD+"LNKLIST"
TMPPFILE=VTM+"TEMP.DBF"

DO LOADLANG              && Carga la tabla de mensajes

SET SOFTSEEK ON
SET ECHO OFF
SET WRAP ON

/* Definición del ambito de trabajo */
SELECT 1
USE &DBFILE ALIAS DIRFAX

/* Declaración de tabla de CAMPOS */
/*

```

```

dbstra:=( )
dbstra:=dbstruct( )
LF=LEN(dbstra)
FOR LI=1 TO LF
  FV(LI)="["+dbstra[LI,1]+"]"
  ? FV(LI)
NEXT
*/
clear
AFIELDS(FV)
ITF=FCOUNT( )
FOR ITX=1 TO ITF
  FV[ITX]="["+FV[ITX]+"]"
NEXT
/* El siguiente archivo e indice, se localizan en un disco definido
para crear archivos temporales, con un disco virtual en RAM se
obtiene una notable aceleracion en la busqueda */

SELECT 2
USE &HITFILE ALIAS SHIST          && SHIST01.NTX : Almacena postings
                                && encontrados para operaciones booleanas
IF IFILE("&HITFILE"+".NTX")      && Checa indice asoc. si no lo construye
INDEX ON SET+RECNUM TO &HITFILE
ENDIF
SET INDEX TO &HITFILE
ZAP                               && Inicialización

SELECT 3
IF IFILE("&KEYFILE"+".NTX")      && Este indice ya esta construido en el CD-ROM
USE &KEYFILE
INDEX ON FLD+"["+KEY TO &KEYFILE
CLOSE
ENDIF
USE &KEYFILE INDEX &KEYFILE ALIAS KEYS

SELECT 4                          && Archivo de ocurrencias
USE &LNKFILE ALIAS LINKS

/*-----*/
/*                               Ciclo Principal del programa
/*-----*/

CLEAR
DO WHILE .T.                      && Control del menú principal
SET COLOR TO N+/W+,W+/N,N,1+/7,W+/N
CLEAR
SET COLOR TO N+/W+,W+/N,N,1+/7,W+/N
  @ 1,1 SAY MSGS[001]
SET COLOR TO W+/1,W+/1,N,1/7,W/3
  @ 2,0 SAY MSGS[002]
SET COLOR TO W+/B+,B+/N,N,N+,N+/N
  @ 2,35 SAY "DIRFAX"
SET COLOR TO TO N+/1+,W+/3+,N,1+/7,W/3
FOR I=3 TO 24
  @ I,0 SAY MSGS[003]
NEXT
SET COLOR TO N+/W+,W+/N,N,1+/7,W+/N

POPUPS(1) = "ACHOICE(2,1,4,15,SUBMENU1)"
POPUPS(2) = "ACHOICE(2,21,0,35,SUBMENU2)"
POPUPS(3) = "ACHOICE(2,41,3,55,SUBMENU3)"
&& @ 1,0 TO 1,79

```

```

OP=POPUP(POPUPS,MENUP,FILAS,CLS,ENGT)
DO CASE
  CASE OP='11'
    IF HIDX>0
      @ 6,10 TO 10,27
      SMP=ACHOICE(7,11,12,26,PRINTM)
      DO CASE
        CASE SMP=1
          DO CARTAS
        CASE SMP=2
          DO ETIQUETAS
        CASE SMP=3
          DO PRINTREC
      ENDCASE
    ELSE
      @ 23,0 CLEAR
      @ 23,1 SAY MSGS(70)  && No hay registros para imprimir ...
      WAIT ""
    ENDIF
  CASE OP='12'
    IF HIDX=0
      DO GRABAREC
    ELSE
      @ 23,0 CLEAR
      @ 23,1 SAY MSGS(87)  && No hay registros para grabar...
      WAIT ""
    ENDIF
  CASE OP='13'
    SET COLOR 10
    CLEAR
    && SELECT 2
    && ZAP
    CLOSE DATABASE
    CLEAR
    BREAK
  CASE OP='21'
/* LI = Es el numero de registro inicial en el campo de indices,
  LF es el numero final, en ambos casos el valor depende del numero
  de campo. 0=Todos 1=Nombre...6=Giro.
  OS = Es el avance inicial dentro del archivo, de ligas KEYLST cada
  campo tiene X numero y van acumulandose como se especifica. */
  LI=1
  LF=1642
  OS=0
  DO MUESTRAXINDICE
  CASE OP='22'
    LI=1643
    LF=2079
    OS=5521
    DO MUESTRAXINDICE
  CASE OP='23'
    LI=2080
    LF=2544
    OS=6020
    DO MUESTRAXINDICE
  CASE OP='24'
    LI=2545
    LF=2627
    OS=6519
    DO MUESTRAXINDICE
  CASE OP='25'
    LI=2628
    LF=2633
    OS=7018
    DO MUESTRAXINDICE
  CASE OP='26'

```

```

      LI=2634
      LF=2635
      OS=7517
      DO MUESTRAXINDICE
CASE OP='27'
      LI=2636
      LF=2819
      OS=8016
      DO MUESTRAXINDICE
CASE OP='28'
      SAVE SCREEN TO SCRO9
      @ 24,1 SAY MSGS[83]
      HIST=ARRAY(90)
      MIDX=0
      HND=0
      SELECT 2
      ZAP
      RESTORE SCREEN FROM SCRO9
CASE OP='31'
      DO IDIOMA
ENDCASE
ENDDO
RETURN

FUNCTION POPUP
PARAMETERS TABLA1,TABLA2,TABLA3,TABLA4,TABLAS  && Menu general tipo scroll
PRIVATE PANTALLA,CONT,OPCION,OP
SAVE SCREEN TO PANTALLA
KEYBOARD CHR(13)
DO WHILE .T.

      FOR CONT=1 TO LEN(TABLA2)
        @ TABLA3[CONT],TABLA4[CONT] PROMPT TABLA2[CONT]
      NEXT
      MENU TO OP

UNPOPUP=TABLA1[1]+(OP=0,OP,1)
@ TABLA3[OP]+1,TABLA4[OP]-1 TO TABLA5[OP]=TABLA3[OP]+2,TABLA4[OP]+15
OPCION=STR(OP,1)+STR(UNPOPUP,1)
DO CASE
CASE LASTKEY()=13
  EXIT
CASE LASTKEY()=27
  OPCION=""
  EXIT
OTHERWISE
  KEYBOARD CHR(LASTKEY())+CHR(13)
ENDCASE
RESTORE SCREEN FROM PANTALLA
ENDDO
RETURN OPCION

FUNCTION A
RETURN ACHOICE()

PROCEDURE IDIOMA
LOCAL VLNTMP
VLNTMP=VLN
@ 2,56 CLEAR TO 5,65
@ 2,56 TO 5,65
@ 3,57 PROMPT MSGS[26]
@ 4,57 PROMPT MSGS[27]
MENU TO VLN
IF VLN<>VLNTMP
DO LOADLANG

```

```
ENDIF
RETURN
```

```
PROCEDURE LOADLANG
```

```
IF VLN=2
MSGFILE=VCD+"SMSG.DBF"    && Se elije la tabla de mensajes en uso
                          && Tabla en español
? "Idioma seleccionado ESPANOL"
ELSE
MSGFILE=VCD+"EMSG.DBF"
? "Default Language : English "
```

```
ENDIF
```

```
FOR ITX=1 TO 10000
```

```
NEXT
```

```
SELECT 5 // Cargando mensajes del sistema //
```

```
IF IFILE("SMSGFILE")
```

```
@ 23,0 SAY "ERROR FATAL - Al intentar abrir el archivo de mensajes"
```

```
BREAK
```

```
ELSE
```

```
USE EMSGFILE
```

```
ENDIF
```

```
IMSG=1
```

```
DO WHILE (EOF())
```

```
MSGS(1MSG)=TRIM(MSG)
```

```
SKIP
```

```
IMSG=IMSG+1
```

```
ENDDO
```

```
CLOSE
```

```
MENUP {1}=MSGS{004}    && Archivo
MENUP {2}=MSGS{005}    && Búsqueda
MENUP {3}=MSGS{006}    && Opciones
```

```
SUBMENUZ{1}=MSGS{15}   && Por Todos los campos
SUBMENUZ{2}=MSGS{16}   && Nombre
SUBMENUZ{3}=MSGS{17}   && Dirección
SUBMENUZ{4}=MSGS{18}   && Colonia
SUBMENUZ{5}=MSGS{19}   && Ciudad
SUBMENUZ{6}=MSGS{20}   && Estado
SUBMENUZ{7}=MSGS{21}   && Giro
SUBMENUZ{8}=MSGS{24}   && Inicio de una nueva búsqueda
SUBMENU {1}=MSGS{08}   && Imprimir
SUBMENU {2}=MSGS{09}   && Salvar
SUBMENU {3}=MSGS{10}   && Salir
PRINTM{1}=MSGS{11}    && Cartas
PRINTM{2}=MSGS{12}    && Etiquetas
PRINTM{3}=MSGS{13}    && Reportes
SUBMENUZ{1}=MSGS{24}   && Idioma
```

```
RETURN
```

```
PROCEDURE MAKESET
```

```
/* Procedimiento para crear y operar conjuntos de
```

```
Búsquedas
```

```
PARAMETERS POSIN,POSNO
```

```
LOCAL SS,K,L,M,N,PSN,RTM,SCR15,DF,RT,SOP
```

```
IF !IDX=1
```

```
SAVE SCREEN TO SCR15
```

```
SOP=1
```

```
SETCOLOR(CIAN)
```

```
@ 2,67,7,78 BOX " [ ] | - | "
```

```
@ 3,68 SAY MSGS{29}
```

```
SETCOLOR(NORMAL)
```

```
@ 4,68 PROMPT " * AND / Y "
```

```
@ 5,68 PROMPT " * OR / O "
```

```

2 0,08 PROMPT " " NOT/Y NO"
MENU TO SOP
DO CASE                                && Elección de operador para efectos de presentación
  CASE SOP=1                            && Intersección
    OB="*"
  CASE SOP=2                            && Unión
    OB="*"
  CASE SOP=3                            && Negación
    OB="*"
ENDCASE
RESTORE SCREEN FROM SCR15
ENDIF

SELECT 4
POSIN=POSIN+OS
GO POSIN
N=0
K=0
M=0
PSN=POSNO
L=0
TND="0"
SETCOLOR(PARPADEO)

/* Carga de ocurrencias a memoria */
@ 24,34 SAY MSGS(29)+STRZERO(L,6)
FOR K=1 TO PSN
  L=L+1
  N=N+1
  TBL&TND(N)=KEYREF
  SKIP
  IF N=4096
    @ 24,34 SAY MSGS(29)+STRZERO(L,6)
    N=0
    N=N+1
    TND=STR(N,1)
  ENDIF
NEXT
SETCOLOR(NORMAL)
@ 24,34 SAY MSGS(29)+STRZERO(L,6)

/* Combinación de conjuntos */
SELECT 2
N=0
M=0
HNO=0
TNO="0"
@ 24,34 SAY MSGS(30)+STRZERO(HNO,6)
IF MIDX>1 /* Si MIDX=1 se carga conjunto
           sin importar " o + */

/* Si es union, se agregan los elementos del ultimo conjunto
chocandose que no se repita el no. de registro */

/* Si es interseccion, se chequea que cada elemento del segundo
conjunto exista en el primero */
DO CASE
  CASE SOP=1
    /* Interseccion de conjuntos:
    Este procesamiento transfiere los numeros de registro
a través del arreglo IRL# a la base de datos SHITS.DBF
para en ella efectuar la operación de intersección */
    SET COLOR TO
    FOR K=1 TO L
      N=N+1
      RIM=TBL&TND(N)

```

```

SS=STRZERO(HIDX-1,4)+RTM
SEEK(SS)
IF FOUND()
  APPEND BLANK
  REPLACE SET WITH STRZERO(HIDX,4)
  REPLACE RECNUM WITH RTM
  HNO=HNO+1
ENDIF
@ 24,34 SAY MSGS[30]+STRZERO(HNO,6)
IF N=4096
  N=0
  H=H+1
  TNO=STR(N,1)
ENDIF
NEXT
CASE SOP=2
/* Unión de conjuntos:
   Pasa los datos del conjunto anterior al siguiente
   y agrega los escleros de la nueva búsqueda al
   nuevo conjunto chequeando que no se repitan los
   elementos.
*/
SS=STRZERO(HIDX-1,4)
SEEK(SS)
IF FOUND()
  DO WHILE SET=SS
    BF=0
    DO WHILE BF<=4095 .AND. SET=SS
      BF=BF+1
      BUFF[BF]=RECNUM
      SKIP
    ENDDO
    RT=RECNO()
    IF SET<=SS .OR. BF>4095
      FOR K=1 TO BF
        APPEND BLANK
        REPLACE SET WITH STRZERO(HIDX,4)
        REPLACE RECNUM WITH BUFF[K]
        HNO=HNO+1
      NEXT
    ENDF
  /* Negación de conjuntos:
   Esta rutina copia los elementos del conjunto anterior
   a el nuevo, posteriormente elimina los elementos negados
   */

```

```

SS=STRZERO(HIDX-1,4)
SEEK(SS)
IF FOUND()
DO WHILE SET=SS
  BF=0
  DO WHILE BF<=4095 .AND. SET=SS
    BF=BF+1
    BUFF[BF]=RECNUM
    SKIP
  ENDDO
  RT=RECNO()
  IF SET<=SS .OR. BF>4095
    FOR K=1 TO BF
      APPEND BLANK
      REPLACE SET WITH STRZERO(HIDX,4)
      REPLACE RECNUM WITH BUFF[K]
      HNO=HNO+1
    NEXT
  ENDIF
  GO RT
  ENDDO
ENDIF
FOR K=1 TO L
  M=M+1
  RTM=DTLETNO(M)
  SS=STRZERO(HIDX,4)+RTM
  SEEK(SS)
  IF FOUND()
    REPLACE SET WITH Spccc(4)
    HNO=HNO+1
  ENDF
  @ 24,34 SAY MSGS(30)+STRZERO(HNO,6)
  IF N=4096
    N=0
    M=M+1
    TNO=STR(M,1)
  ENDF
NEXT
ENDCASE
ELSE
  FOR K=1 TO L
    M=M+1
    RTM=DTLETNO(M)
    APPEND BLANK
    REPLACE SET WITH STRZERO(HIDX,4)
    REPLACE RECNUM WITH RTM
    HNO=HNO+1
  @ 24,34 SAY MSGS(30)+STRZERO(HNO,6)
  IF N=4096
    N=0
    M=M+1
    TNO=STR(M,1)
  ENDF
NEXT
ENDIF
SETCOLOR(CIAN)
@ 24,34 SAY MSGS(30)+STRZERO(HNO,6)
SETCOLOR(NORMAL)
SELECT 3
RETURN

PROCEDURE MUESTRAXINDICE
/* Muestrax Indices */
LOCAL LLAVE, MARCA, SCR04, POSIN, POSNO
SELECT 3
MARCA=LI

```

```

SAVE SCREEN TO SCRO4
@ 6,18,22,78 BDK " [ ] | ^ - | "
KB=0
LLAVE=""
GO LI
DO BDSPSCR WITH MARCA
DO WHILE .T.
  AP=RECNO()
  SELECT 3
  @ 5,20 SAY MSGS(31)+LLAVE+SPACE(30-LEN(LLAVE))
  SETCOLOR(CIAN)
  @ 24,0 SAY MSGS(32)
  SETCOLOR(NORMAL)
  KB=INKEY(0)
DO CASE
  CASE KB=8
    LLAVE=SUBSTR(LLAVE,1,LEN(LLAVE)-1)
    IF LEN(LLAVE)>0
      DO BSEARCH WITH LLAVE
      MARCA=AP
      GO MARCA
    ELSE
      GO LI
    ENDIF
    DO BDSPSCR WITH MARCA
  CASE KB=18
    IF AP-20>LI
      SKIP -20
      MARCA=RECNO()
    ELSE
      GO LI
      MARCA=LI
    ENDIF
    LLAVE=ALLTRIM(KEY)
    DO BDSPSCR WITH MARCA
  CASE KB=3
    IF AP+6<LF
      SKIP +6
      MARCA=RECNO()
    ELSE
      GO LF
      MARCA=LF
    ENDIF
    LLAVE=ALLTRIM(KEY)
    DO BDSPSCR WITH MARCA
  CASE KB=1
    GO LI
    MARCA=RECNO()
    LLAVE=ALLTRIM(KEY)
    DO BDSPSCR WITH MARCA
  CASE KB=6
    GO LF
    MARCA=RECNO()
    SKIP -7
    LLAVE=ALLTRIM(KEY)
    @ 7,17 CLEAR TO 21,77
    DO BDSPSCR WITH MARCA
  CASE KB=5
    GO MARCA
    SKIP -1
    LLAVE=ALLTRIM(KEY)
    MARCA=RECNO()
    DO BDSPSCR WITH MARCA
  CASE KB=24
    GO MARCA

```

```

SKIP +1
LLAVE=ALLTRIM(KEY)
MARCA=RECNO()
DO BDSPPSCR WITH MARCA
CASE KB=2B
CASE KB=19
CASE KB=4
CASE KB=-8 .OR. KB=-18
DO SHOWHIST
CASE KR=27
EXIT
CASE KB=CR
DO BSEARCH WITH LLAVE
MARCA=AP
LLAVE=KEY
@ 5,20 SAY MSGS(31)=LLAVE+SPACE(30-LEN(LLAVE))
POSIN=PIN
POSNO=POS
HIDX=HIDX+1
DO MAKESET WITH POSIN,POSNO
HIST(HIDX)*KN+STRZERO(PIN,B)+STRZERO(POS,6)+STRZERO(REC,6)+STRZERO(HIDX,6)+STRZERO(MNO,6)+OB+STR(VAL(OP)-20,1)+LLAVE
@ 5,35 SAY SPACE(30)
GO MARCA
@ 5,20 SAY SPACE(45)
CASE KB=-9 .OR. KB=-19
DO DISPLAYREC
OTHERWISE
LLAVE=LLAVE+UPPER(CHR(KB))
DO BSEARCH WITH LLAVE
MARCA=RECNO()
DO BDSPPSCR WITH MARCA
ENDCASE
ENDDO
RESTORE SCREEN FROM SCRD4
SELECT 1
RETURN

```

/\* Muestra Historia de Búsquedas \*/

PROCEDURE SHOWHIST

LOCAL SCRD5,1,RN

IDX=0

SAVE SCREEN TO SCRD5

@ 6,1,22,60 BOX " [ ] |J-1 | "

@ 6,1,8,60 BOX " [ ] |J-1 | "

@ 7,2 SAY MSGS(33)

RN=0

IF HIDX=0

IF HIDX=15

FOR I=HIDX-15 TO HIDX

@ RN,2 SAY SUBSTR(HIST(I),27,6)

@ RN,9 SAY SUBSTR(HIST(I),33,6)

@ RN,16 SAY SUBSTR(HIST(I),39,1)

@ RN,19 SAY SUBSTR(HIST(I),41,25)

@ RN,45 SAY SUBMENUZ(VAL(SUBSTR(HIST(I),40,1)))

RN=RN+1

NEXT

ELSE

FOR I=1 TO HIDX

@ RN,2 SAY SUBSTR(HIST(I),27,6)

@ RN,9 SAY SUBSTR(HIST(I),33,6)

@ RN,16 SAY SUBSTR(HIST(I),39,1)

@ RN,19 SAY SUBSTR(HIST(I),41,25)

@ RN,45 SAY SUBMENUZ(VAL(SUBSTR(HIST(I),40,1)))

RN=RN+1

NEXT

ENDIF

&& Conjunto  
&& No. de Hits  
&& Operación  
&& Termino  
&& Campo

```

@ 25,0
ENDIF
WAIT ""
RESTORE SCREEN FROM SCROS
KB=0
RETURN

```

```

PROCEDURE BDSPPSCR
PARAMETERS MARCA
@ 7,19 CLEAR TO 21,77
@ 6,21 SAY MSGS[34]
@ 6,52 SAY MSGS[35]
@ 6,71 SAY MSGS[66]
R=7
IF RECNO()-7>L1
SKIP -7
ENDIF
DO WHILE R<21
IF RECNO()<=LF .AND. RECNO()>>L1
IF MARCA=RECNO()
SETCOLOR(CIAM)
ELSE
IF R=14
SETCOLOR(NORMAL)
ENDIF
ENDIF
@ R,21 SAY SUBSTR(KEY,1,30)
@ R,53 SAY POS
@ R,68 SAY REC
SETCOLOR(NORMAL)
ENDIF
IF RECNO()<LF
R=R+1
SKIP
ELSE
RETURN
ENDIF
ENDDO
RETURN

```

```

/* Búsqueda por índices */
PROCEDURE BSEARCH
PARAMETERS WORD
LOCAL WORDTMP
/* El índice es construido utilizando el número de campo, un delimitador y
un carácter especial <"> seguido del término indexado. */
SELECT 3
SET SOFTSEEK ON
WORDTMP=STR(VAL(OP)-21,1)+"H"+WORD
SEEK(WORDTMP)
AP=RECNO()
SET SOFTSEEK OFF
RETURN

```

```

/* Cñcolor : Cambio de color en la presentación de datos para la cadena
que cumple con las condiciones de búsqueda */
PROCEDURE CHDCOLOR
PARAMETERS FLD,RW,CL,FLT
LOCAL FLDARR[64],FLDCHI[64],I,IX,POINTER,ST,RESULT,J
I=0
ST=ALLTRIM(FLD)
WHILE .T.
I=I+1
POINTER=AT(" ",ST)

```

```

IF POINTER=0
  FLDARR(1)=ST
  EXIT
ENDIF
FLDARR(1)=SUBSTR(ST,1,POINTER-1)
ST=SUBSTR(ST,POINTER+1,LEN(ST)-POINTER+1)
ENDDO

J=0

&& @ RM,19 SAY SUBSTR(HIST(1),41,25)
&& @ RM,45 SAY SUBMENUZ[VAL(SUBSTR(HIST(1),40,1))]

FOR IX=1 TO MIDX
  RESULT=ASCAM(FLDARR,ALLTRIM(SUBSTR(HIST(IX),41,25)))
  IF RESULT>0
    J=J+1
    FLDCHI(J)=RESULT
  ENDIF
NEXT

@ RW,CL
FOR IX=1 TO I
  IF ASCAM(FLDCHI,IX)>0
    SETCOLOR(CIAN)
    @ RW,COL()+1 SAY FLDARR(IX)

    SETCOLOR(NORMAL)
  ELSE
    @ RW,COL()+1 SAY FLDARR(IX)
  ENDIF
NEXT
RETURN

```

/\* Muestra registros resultado de búsquedas \*/

```

PROCEDURE SHREC
  LOCAL SCR1,AFLD(7)
  @ 9,15 CLEAR TO 17,68
  AFLD(1)=NOMBRE
  AFLD(2)=DIRECCION
  AFLD(3)=COLOMIA

  IF LEN(ALLTRIM(CODIGO))>0
    AFLD(4)=TRIM(CIUDAD)+" "+TRIM(ESTADO)
  ELSE
    AFLD(4)=CODIGO+" "+TRIM(CIUDAD)+" "+TRIM(ESTADO)
  ENDIF

  AFLD(5)="TEL : "+TELEFONO
  AFLD(6)="FAX : "+NUMFAX
  AFLD(7)="GIRO : "+GIRO

  DO CHCOLOR WITH AFLD(1),10,15
  DO CHCOLOR WITH AFLD(2),11,15
  DO CHCOLOR WITH AFLD(3),12,15
  DO CHCOLOR WITH AFLD(4),13,15
  DO CHCOLOR WITH AFLD(5),14,15
  DO CHCOLOR WITH AFLD(6),15,15
  DO CHCOLOR WITH AFLD(7),16,15

/* Si el campo IMAGEN es verdadero, se presenta la opción de despliegue,
el nombre del archivo de imagen es el número de registro seguido por la
extensión "PCX".
IF IMAGEN

```

```

@ 17,16 SAY MSGS(36)
KB=INKEY(0)
IF KB=7 .OR. KB=17
SAVE SCREEN TO SCR17
IMGFILE=STRZERO(RECNO(),6)+"-PEX"
DO DSPIMG WITH IMGFILE
RESTORE SCREEN FROM SCR17
ENDIF
ENDIF
RETURN

```

```

PROCEDURE DISPLAYREC
LOCAL SCROT1,MIN,MAX,SETAUX,APREC,SU,SN,HN,RN
SAVE SCREEN TO SCR01
@ 8,10,23,69 BOX " | | | | | "
SN=STRZERO(HIDX,4)
MIN=0
APREC=0
SELECT 2
SEEK(SN)
SU=FOUND()
IF SU
MIN=RECNO()
ENDIF
HN=SUBSTR(HIST(VAL(SN)),33,6)
MAX=MIN+VAL(HN)-1
RN=1
DO WHILE SU
SELECT 2
APREC=VAL(RECNUM)
@ 23,55 SAY STRZERO(RN,6)+"/*"+HN
SELECT 1
GO APREC
DO SHREC
SELECT 2
KB=INKEY(0)
DO CASE
CASE KB=8
CASE KB=15
IF RN=1
SKIP -1
RN=RN-1
ENDIF
CASE KB=3
IF RN<VAL(HN)
SKIP
RN=RN+1
ENDIF
CASE KB=1
SEEK(SN)
RN=1
CASE KB=6
SEEK(SN)
SKIP MAX-1
RN=VAL(HN)
CASE KB=5
IF RN=1
SKIP -1
RN=RN-1
ENDIF
CASE KB=24
IF RN<VAL(HN)
SKIP
RN=RN+1
ENDIF
CASE KB=28

```

```

CASE KB=19
CASE KB=4
CASE KB=27
  KB=4
  EXIT
CASE KB=CR
  IF RN<VAL(HN)
    SKIP
    RN=RN+1
  ENDIF
ENDCASE
ENDDO
RESTORE SCREEN FROM SCRD1
RETURN

```

```

/* Impresion de resultados a partir del número de
conjunto que elije el usuario */

```

```

PROCEDURE PRJMTREC
LOCAL SCRD1,APREC,RW
SAVE SCREEN TO SCRD1
DO SELESET
APREC=0
SELECT 2
SEEK(SM)
RW=10
DO WHILE SM=SET
  APREC=VAL(RECNUM)
  SELECT 1
  GO APREC
  SET DEVICE TO SCREEN
  @ 23,1 SAY MSGS(37)+STRZERO(RECNO(),8)
  SET DEVICE TO PRINT
  @ RW,10 SAY NOMBRE
  RW=RW+1
  @ RW,10 SAY DIRECCION
  RW=RW+1
  @ RW,10 SAY COLONIA
  RW=RW+1
  IF LEN(ALLTRIM(CODIGO))=0
    @ RW,15 SAY TRIM(CIUDAD)+" "+TRIM(ESTADO)
    RW=RW+1
  ELSE
    @ RW,16 SAY SPACE(50)
    RW=RW+1
    @ RW,16 SAY CODIGO+" "+TRIM(CIUDAD)+" "+TRIM(ESTADO)
  ENDIF
  @ RW,16 SAY "TEL : "+TELEFONO
  RW=RW+1
  @ RW,16 SAY "FAX : "+NUMFAX
  RW=RW+1
  @ RW,16 SAY "GIRO: "+GIRO
  RW=RW+2
  IF RW=55
    EJECT
    RW=5
  ENDIF
SELECT 2
SKIP
ENDDO

```

```
SET DEVICE TO SCREEN
RESTORE SCREEN FROM SCR01
```

```
RETURN
```

```
/* Graba registros al dispositivo especificado a partir del
conjunto que elije el usuario */
```

```
PROCEDURE GRABAREC
```

```
LOCAL SCR01,APREC,RW,FM,IORES
```

```
SAVE SCREEN TO SCR01
```

```
DO SELECT
```

```
FM=""
```

```
IORES=-1
```

```
DO WHILE IORES=-1
```

```
FM="PRINT.LST"
```

```
@ 23,1 SAY MSGS(38) GET FM
```

```
READ
```

```
IORES=FCREATE(FM,0)
```

```
ENDDO
```

```
APREC=0
```

```
SELECT 2
```

```
SEEK(SN)
```

```
RW=10
```

```
DO WHILE SN=SET
```

```
APREC=VAL(RECLNUM)
```

```
SELECT 1
```

```
GO APREC
```

```
@ 23,1 SAY MSGS(39)+STRZERO(RECHOC(),8)
```

```
FWRITE(IORES,NOMBRE+CHR(13)+CHR(10))
```

```
FWRITE(IORES,DIRECCION+CHR(13)+CHR(10))
```

```
FWRITE(IORES,COLOMIA+CHR(13)+CHR(10))
```

```
IF LEN(ALLTRIM(CODIGO))=0
```

```
FWRITE(IORES,TRIM(CIUDAD)+CHR(13)+CHR(10))
```

```
ELSE
```

```
FWRITE(IORES,TRIM(CIUDAD)+""+TRIM(ESTADO)+CHR(13)+CHR(10))
```

```
ENDIF
```

```
FWRITE(IORES,"TEL : "+TELEFONO+CHR(13)+CHR(10))
```

```
FWRITE(IORES,"FAX : "+NUMFAX+CHR(13)+CHR(10))
```

```
FWRITE(IORES,"GIRO : "+GIRO+CHR(13)+CHR(10))
```

```
FWRITE(IORES,CHR(13)+CHR(10))
```

```
FWRITE(IORES,CHR(13)+CHR(10))
```

```
RW=RW+10
```

```
IF RW>50
```

```
FWRITE(IORES,CHR(13)+CHR(10)+CHR(12))
```

```
RW=10
```

```
ENDIF
```

```
SET DEVICE TO SCREEN
```

```
SELECT 2
```

```
SKIP
```

```
ENDDO
```

```
FCLOSE(IORES)
```

```
RESTORE SCREEN FROM SCR01
```

```
RETURN
```

```
/* Procedimiento de despliegue de imagenes en formato PCX
se utiliza llamada a libreria TBASE(TM) */
```

```
PROCEDURE DSPIMG
```

```
PARAMETERS IMGFILE
```

```
TB_PARAMS=""N"+IMGFILE
```

```
TB_RETURN=TB_PCX(TB_PARAMS)
```

```
IF LEFT(TB_RETURN,2)=""00"
```

```
    && Se chequea que el resultado del
```

```
KB=INKEY(0)
```

```
    && despliegue haya funcionado O.K.
```

```
TB_PARAMS=SPACE(4)
```

```
TB_RETURN=TB_TEXT(TB_PARAMS)
```



```

FOR K=1 TO TIF
TEXTOKJ=SUBSTR(TEXTOK,(K-1)*65+1,65)
NEXT
RESTORE SCREEN FROM SCR
DO EDITACARTA

CASE COP="M"
RETURN

ENDCASE
RETURN && CORTA

PROCEDURE EDITACARTA
LOCAL COP
// Obtiene fecha en cadena de caracteres //
SETCOLOR(CIAN)
I=1

DO WHILE .T.
CLEAR
@ 2,0,22,79 BOX "||| |"
@ 1,0 SAY MSGS(46)
SETCOLOR(NORMAL)
DO LEETABLA
COP="M"
DO WHILE AT(COP,"MEXC")>0
COP="M"
@ 23,0 SAY MSGS(47) GET COP PICTURE "I" VALID COP "S*CHRX"
@ 24,0 SAY MSGS(48)
READ
COP=UPPER(COP)
ENDDO
DO CASE
CASE COP="C"
@ 24,0 CLEAR
SM="M"
@ 24,1 SAY MSGS(49) GET SM PICTURE "I" VALID SM "S*SMY"
READ
SM=UPPER(SM)
IF SM="M" .OR. SM="Y"
RETURN
ENDIF
CASE COP="M"
IF I=58
LOOP
ELSE
I=39
LOOP
ENDIF
CASE COP="B"
DO CASE
CASE I=20
I=1
CASE I=39
I=1
CASE I=58
I=20
OTHERWISE
I=20
ENDCASE
LOOP
CASE COP="X"
DO FS
RETURN
ENDCASE
ENDDO

```

```
CLEAR
RETURN
```

```
PROCEDURE LEETABLA
```

```
## Ingreso del texto de una carta ...
```

```
LOCAL J,K
CLEAR GETS
@ 23,0 CLEAR
@ 23,0 SAY MSGS(50)
@ 24,0 SAY MSGS(51)
TIF=1+18
J=3
FOR K=1 TO TIF
  @ J,2 SAY STR(K,2) GET TEXTOKJ
  J=J+1
NEXT
READ
I=K
RETURN
```

```
PROCEDURE HELPP
```

```
LOCAL SCROB
SCROB=SAVESCREEN()
SETCOLOR(CIAM)
@ 3,3,21,70 BOX " | | | | "
SETCOLOR(NORMAL)
@ 4,4 CLEAR TO 10,69

@ 6,5 SAY MSGS(52)
@ 7,5 SAY MSGS(53) ## F1 - Muestra esta ayuda
@ 8,5 SAY MSGS(54) ## F2 - Muestra nombres de campos
@ 9,5 SAY MSGS(55) ## F10 - Graba el texto a un archivo en disco
@11,5 SAY MSGS(56) ## DESPLAZAMIENTO:
@12,10 SAY MSGS(57) ## Flechas < - > = Avanza/Retrocede una posición
@13,10 SAY MSGS(58) ## <Ctrl> V <- -> = Avanza/Retrocede una palabra
@14,10 SAY MSGS(59) ## <PgDn> = Avanza una página
@15,10 SAY MSGS(60) ## <PgUp> = Retrocede una página
@17,5 SAY MSGS(61) ## PERSONALIZACION:
@18,10 SAY MSGS(62) ## Los campos deben ser precedidos por el caracter @ para
@19,10 SAY MSGS(63) ## que sean considerados en la impresión
@20,5
WAIT " "
RESTORE SCREEN FROM SCROB
RETURN
```

```
PROCEDURE FS ## Salva archivo definido
```

```
LOCAL I,J,K,SCR10,SH,PI,FN
SAVE SCREEN TO SCR10
SETCOLOR(CIAM)

@ 3,3,21,70 BOX " | | | | "
SETCOLOR(NORMAL)

@ 4,4 CLEAR TO 12,59
FN="CARTAGO:"
DO WHILE .T.
  @ 5,5 SAY MSGS(64) get FN ## Nombre de archivo (8 caracteres max.):
  READ
  IF LEN(ALLTRIM(FN))=0
    BREAK
  ENDF
  SH="N"
  FN=FN+".MEM"
  IF FILE(FN)
```

```

SETCOLOR(PARPADEO)
@ 12,5 SAY MSGS(65) && ARCHIVO EXISTENTE...SOBREScribir 7 (S/N) :
SETCOLOR(NORMAL)
@ 12,66 GET SW PICT "" VALID SW $"SY"
READ
ELSE
EXIT
ENDIF
SW=UPPER(SW)
IF SW="S" .OR. SW="Y"
EXIT
ELSE
LOOP
ENDIF
ENDDO
BTEXT0=""
FOR K=1 TO 17
    BTEXT0=BTEXT0+TEXT0(K)
NEXT
SAVE TO &FN
DO PLETTER
RESTORE SCREEN FROM SCR10
RETURN

```

```

PROCEDURE SHFLD    && Procedimiento que se activa con <F2>, para
                  && seleccionar como macros los campos para intercalacion
LOCAL SCR11,FL,FX,CX,RX
RW=ROW()
CL=COL()
SAVE SCREEN TO SCR11
IF CL<40
    CX=60
ELSE
    CX=5
ENDIF
IF RW>11
    RX=3
ELSE
    RX=14
ENDIF
SETCOLOR(CIAN)
@ RX-1,CX-1,RX+17,CX+11 BOX " | | | | "
SETCOLOR(NORMAL)

FOR ITX=1 TO 17F
    @ RX+ITX-1,CX PROMPT FV(ITX)
NEXT
@ 23,1 SAY MSGS(67) &&Select a field form the list - use arrows to move

MENU TO FL
IF FL>=1 .AND. FL<=10
    TX=SUBSTR(TEXT0[RW-2],1,CL-4)
    TX=TX+FV(FL)
    TEXT0[RW-2]=TX+SPACE(65-LEN(TX))
ENDIF

RESTORE SCREEN FROM SCR11
RETURN

```

/\* Procedimiento para la impresion de cartas personalizadas \*/

```

PROCEDURE PLETTER
LOCAL SCR01,APREC,1,J,EXISTFLD

```

```

LOCAL K,FLD,PI,LT,TM
SAVE SCREEN TO SCR01
DD SELSET
DD SELOUTPUTDEV
APREC=0
SELECT 2          && Base de datos de postings
SEEK(SN)         && Se localizan los postings del conjunto SN (SetNumber)
RW=10
AFILL(PTABLE,SPACE(65))
DO WHILE SN=SET   && Mientras existan elementos del conjunto SN
  APREC=VAL(RECLUM) && Se obtiene el # de reg de archivo maestro a imprimir
  SELECT 1       && Seleccionamos archivo maestro
  GO APREC
  SET DEVICE TO SCREEN
  @ 23,1 SAY "IMPRIMIENDO REGISTRO: ""STRZERO(RECNO(),8)
  IF VOO=2
    SET DEVICE TO PRINT
  ENDIF
  FOR J=1 TO 10
    FC(J)=SUBSTR(FV[J],2,AT(*)",FV[J])-2) && Se carga los datos del reg.
                                         && actual en el arreglo FC
  NEXT
  FOR K=1 TO TIF
    EXISTFLD=.F.
    LT=TEXT0(K)
    FOR J=1 TO 10 && 10 Es el # max. de campos a macrosustituir
      PI=AT(FV[J],LT)
      IF PI>0
        EXISTFLD=.T.
        TH=SUBSTR(LT,1,PI-1)
        FLD=FC(J)
      && En la línea siguiente se macrosustituye el contenido del campo
      && especificado por FLD
      TH=TH+ALLTRIM(&FLD)+SUBSTR(LT,PI+LEN(FV[J]),LEN(LT)-PI-LEN(FV[J]))
      CLEAR
      LT=TH
    ENDIF
  NEXT
  IF !EXISTFLD && Si no existe un campo para macrosustitución
    PTABLE(K)=TEXT0(K) && se asigna su valor a la tabla de salida
  ELSE
    PTABLE(K)=TH
  ENDIF
NEXT
NEXT
DO CASE
CASE VOO=3
  FOR K=1 TO TIF
    FWRITE(FD,PTABLE(K)+CHR(13)+CHR(10))
  NEXT
CASE VOO=2
  FOR K=1 TO TIF
    ? PTABLE(K)
  NEXT
CASE VOO=1
  FOR K=1 TO TIF
    ? PTABLE(K)
  FOR I=1 TO 5000
  NEXT
NEXT
ENDCASE
SELECT 2
SKIP
ENDDO
IF VOO=3

```

```

FCLOSE(FP)
ENDIF
SET DEVICE TO SCREEN
RETURN

```

```

PROCEDURE ETIQUETAS && Impresión de etiquetas

```

```

/* Define tamaño de etiquetas,
  Selecciona campos para impresión
  Selecciona rango o conjunto de impresión
  Selecciona dispositivo de impresión
  Impresión */
SET COLOR TO M+/1+,W+/3+,N,1-/7,W/3
@ 0,0 CLEAR
@ 0,32 SAY MSGS[01] && Osiris
@ 1,0 SAY MSGS[02] && -----
@ 1,33 SAY MSGS[71] && ETIQUETAS

SET COLOR TO M+/W+,W+/H,M,1-/7,W*/M
FOR J=3 TO 23
  @ 1,0 SAY MSGS[03] && NO
NEXT J
/* Dimensiones de las etiquetas */
DO WHILE .T.
  CXE = 40
  LXE = 8
  SEE = 1
  EXL = 1
  MXE = 5

  @ 5,1 SAY MSGS[72] && Especifique las dimensiones de la etiqueta:
  @ 7,1 SAY MSGS[73] GET CXE PICT "999" && "ANCHO DE LA ETIQUETA :
  @ 8,1 SAY MSGS[74] GET LXE PICT "99" && "LINEAS POR ETIQUETA :
  @ 9,1 SAY MSGS[75] GET MXE PICT "99" && "MARGEN IZQUIERDO :
  @10,1 SAY MSGS[76] GET SEE PICT "99" && "SEPARACION EXNTR ETO. :
  @11,1 SAY MSGS[77] GET EXL PICT "99" && "NUM.DE ETIQ. X LINEA : "
  READ
  IF CXE<132 .OR. CXE<=0 .OR. LXE>66 .OR. LXE<=0 .OR. MXE<=0 .OR. MXE>55 ;
  .OR. SEE>55 .OR. EXL<=0 .OR. EXL>4
  @ 23,1 SAY MSGS[78] && "ERROR : Dimensiones de etiqueta invalidas"
  WAIT = "
  CH=INKEY(0)
  IF CH=27
    RETURN
  ENDIF
  @ 23,0 CLEAR
  LOOP
ELSE
  EXIT
ENDIF
ENDDO
/* Selección de campos */
IF=LCX
AFILL(TEXT0,SPACE(CXE))
@ 5,1 CLEAR
@ 5,1 SAY MSGS[79] &&Las dimensiones de la etiqueta serán como se muestra,
@ 6,1 SAY MSGS[80] &&Selecione el orden de los campos en impresión
@ 23,1 SAY MSGS[81] &&Oprima <F2> para seleccionar campos a imprimir"
SET KEY -1 TO SHTAG && Asigna procedimiento de selección de campos a <F2>
LIA=7
LDA=LIA+LCX-1
CIA=(79-CXE)/2
CDA=CIA+LCX-1
@ LIA,CIA,LDA,CDA BOX " | | | | "
I=0
FOR RW=LIA+1 TO LDA+1

```

```

I=I+1
  @ RV,CIA+1 GET TEXTO(I)
NEXT
READ
DO PTAGS
RETURN

PROCEDURE SHTAG
LOCAL SCR11,FL, TX, CX, RX
RV=ROW()
CL=COL()
SAVE SCREEN TO SCR11
IF CL<40
  CX=60
ELSE
  CX=5
ENDIF
ENDIF
IF RW<11
  RX=12
ELSE
  RX=5
ENDIF

SETCOLOR(CIAM)
  @ RX-1, CX-1, RX=ITF, CX+11 BOX " [ ] [ ] "
SETCOLOR(NORMAL)

FOR ITX=1 TO ITF
  @ RX+ITX-1, CX PROMPT FV(ITX)
NEXT
MENU TO FL
IF FL=1 .AND. FL=10
  TX=SUBSTR(TEXTO(RW-LIA+1), 1, CL-CIA+1)
  TX=TX+FV(FL)
  TEXTO(RW-LIA)=TX+SPACE(CX-LEN(TX))
ENDIF
RESTORE SCREEN FROM SCR11
RETURN

```

/\* Selecciona de las búsquedas realizadas el conjunto de registros a procesar para descarga a disco o Impresión \*/

```

PROCEDURE SETSET
LOCAL STP, RN, LN
  @ 7,2 CLEAR TO 23,51
  @ 6,1,22,60 BOX " [ ] [ ] "
  @ 6,1,8,60 BOX " [ ] [ ] "
  @ 7,2 SAY MSGS(33)
  @ 6,2 SAY MSGS(68) 68SELECCIONE EL CONJUNTO A IMPRIMIR
RN=0
SELECT 2
IF HIDX=0      && Sin HIDX=0 No se ha realizado alguna búsqueda
IF HIDX>15    && Implica buscar conjuntos en varias pantallas
FOR LN=HIDX-15 TO HIDX
  @ RN,2 PROMPT SUBSTR(HIST(LN),27,6)+ " *";
  SUBSTR(HIST(LN),33,6)+ " *";
  SUBSTR(HIST(LN),39,13)+ " *";
  SUBSTR(HIST(LN),41,25)+ " *";
  SUBMENU2(VAL(SUBSTR(HIST(LN),40,1)))
  RN=RN+1
NEXT
MENU TO STP
IF STP=0 .AND. STP=15
  SN=SUBSTR(HIST(HIDX+STP-14),29,4) && Obtiene # de conjunto selec.

```

```

ELSE
RETURN
ENDIF
ELSE
FOR LN=1 TO NIDX
@ RN,2 PROMPT SUBSTR(HIST(LN),27,6)+ " "+;
SUBSTR(HIST(LN),33,6)+ " "+;
SUBSTR(HIST(LN),39,1)+ " "+;
SUBSTR(HIST(LN),41,25)+ " "+;
SUBMEMU2(VAL(SUBSTR(HIST(LN),40,1)))
RN=RN+1
NEXT
MENU TO STP
IF STP>0 .AND. STP<15
SM=SUBSTR(HIST(STP),29,4)
ELSE
RETURN
ENDIF
ENDIF
ELSE
@ 23,0 CLEAR
@ 23,1 SAY MSGS(70) &&"Error : No existen conjuntos para imprimir... "
WAIT ***
RETURN
ENDIF
RETURN
PROCEDURE PTAGS
LOCAL APREC,EXISTFLD,K,J,TH,FLD,PI,LT,PK
DO SELSET
DO SELOUTPUTDEV
APREC=0
SELECT 2 && Base de datos de postings
SEEK(SM) && Se localizan los postings del conjunto SM (SetNumber)
PK=0
AFILL(PTABLE, "")
DO WHILE SM=SET && Mientras existan elementos del conjunto SM
APREC=VAL(RECNUM) && Se obtiene el # de reg de archivo maestro a imprimir
SELECT 1 && Seleccionamos archivo maestro
GO APREC
SET DEVICE TO SCREEN
@ 23,1 SAY "PROCESANDO EL REGISTRO: "+STRZERO(RECNO(),8)
DO CASE && Selecciona dispositivo de salida
CASE VOD=2
SET DEVICE TO PRINT
CASE VOD=1
SET DEVICE TO SCREEN
ENDCASE
FOR J=1 TO 10 && Se carga los datos del reg.
FC[J]=SUBSTR(FV[J],2,AT(")",FV[J])-2) && actual en el arreglo FC
NEXT
FOR K=1 TO IIF
EXISTFLD=.F.
LT=TEXTOK
FOR J=1 TO 10 && 10 Es el # max. de campos a macrosustituir
PI=AT(FV[J],LT)
IF PI=0
EXISTFLD=.T.
TH=SUBSTR(LT,1,PI-1)
FLD=FC[J]
&& En la linea siguiente se macrosustituye el contenido del campo
&& especificado por FLD
TH=TH+ALLTRIM(FLD)+SUBSTR(LT,PI+LEN(FV[J]),LEN(LT)-PI-LEN(FV[J]))

```

```

CLEAR
LT=TM
ENDIF
NEXT
IF TEXTISFLD          ** SI no existe un campo para macrosustitución
  TEXTO(K)=ALLTRIM(TEXTO(K))
  IF LEN(TEXTO(K))=CXE
    TEXTO(K)=TEXTO(K)=REPLICATE(" ",CXE-LEN(TEXTO(K)))
  ELSE
    IF TEXTO(K)=CXE
      TEXTO(K)=SUBSTR(TEXTO(K),1,CXE)
    ENDIF
  ENDIF
  PTABLE(K)=PTABLE(K)+SPACE(NCXE)+TEXTO(K)+SPACE(SEE) ** se asigna su valor a la Tabla de salida
ELSE
  TM=ALLTRIM(TM)
  IF LEN(TM)=CXE
    TM=TM=REPLICATE(" ",CXE-LEN(TM))
  ELSE
    IF LEN(TM)>CXE
      TM=SUBSTR(TM,1,CXE)
    ENDIF
  ENDIF
  PTABLE(K)=PTABLE(K)+SPACE(NCXE)+TM+SPACE(SEE)
ENDIF
NEXT
PRX=PX+1
IF PRX=EKL
  PRX=0
  FOR K=1 TO IIF
    IF VCD=3
      FWRITE(FP,PTABLE(K)+CHR(13)+CHR(TO))
    ELSE
      ? PTABLE(K)
    ENDIF
  NEXT
  AFILL(PTABLE,"")
ENDIF
SELECT 2
SKIP
ENDDO
IF VCD=3
  FCLOSE(FP)
ENDIF
SET DEVICE TO SCREEN
RETURN

```

/\* Selección del dispositivo de salida \*/

```

PROCEDURE SELOUTPUTDEV
LOCAL DEVICES(3),SCR *
DEVICES(1)="SCREEN"
DEVICES(2)="PRINTER"
DEVICES(3)="FILE"
FN="PRINT.LST"
SAVE SCREEN TO SCR
SET COLOR TO M+/M,M+/M,M,1-/7,M+/M
@ 0,14 CLEAR TO 14,40
@ 0,14 TO 14,40
@ 0,14 TO 0,40
@ 7,15 SAY "Select output device: "
@ 0,14 TO 13,23
@ 0,15 SAY ""

```

```
FOR I=1 TO 3
  @ ROW()=1,16 PROMPT DEVICES(I)
NEXT
MENU TO VOD
DO CASE
  CASE VOD=3
    VFN="PRINT.LST"
    @ 21,1 SAY "PRINT FILE NAME : " GET VFN
    READ
    IF (FN="") .AND. (FN=" ")
      RETURN
    ENDIF
    FP=FCREATE(VFN,0)
  CASE VOD=2
    SET DEVICE TO PRINT
  CASE VOD=1
    SET DEVICE TO SCREEN
ENDCASE
RESTORE SCREEN FROM SCR
RETURN
```

```
/*
PROGRAMA : IGEN.PRG
FUNCION : GENERACION DE UN ARCHIVO INVERTIDO A PARTIR DE UNA
        BASE DE DATOS DBASE
AUTORES : BERTHA CAMPOS MARROQUIN
        DAVID DEL MORAL CORDOBA
FECHA : 1998-01-01

COMENTARIOS : Indexacion por palabra de todos los campos que componen
        la base de datos DIRFAX
```

```
SET EXACT ON
```

```
/* Declaracion de variables */
PUBLIC IDX,P1,PF,I,FLD,SG,ST,Y,CH,KN,KNTMP,KEY,KEYTMP,RN
DECLARE KEYTABL(4096)
DECLARE STOPWORD(50)
```

```
STOPWORD(01)="&"
STOPWORD(02)="A"
STOPWORD(03)="BAJO"
STOPWORD(04)="CABE"
STOPWORD(05)="CONTRA"
STOPWORD(06)="CV"
STOPWORD(07)="DE"
STOPWORD(08)="DEL"
STOPWORD(09)="DESDE"
STOPWORD(10)="EL"
STOPWORD(11)="ELLA"
STOPWORD(12)="ELLAS"
STOPWORD(13)="ELLO"
STOPWORD(14)="ELLOS"
STOPWORD(15)="EN"
STOPWORD(16)="HASTA"
STOPWORD(17)="HAY"
STOPWORD(18)="HAYA"
STOPWORD(19)="LA"
STOPWORD(20)="LAS"
STOPWORD(21)="LE"
STOPWORD(22)="LES"
STOPWORD(23)="LO"
STOPWORD(24)="LOS"
STOPWORD(25)="MAS"
STOPWORD(26)="ME"
STOPWORD(27)="MI"
STOPWORD(28)="NO"
STOPWORD(29)="OD"
STOPWORD(30)="OTRA"
STOPWORD(31)="OTRAS"
STOPWORD(32)="OTRO"
STOPWORD(33)="OTROS"
STOPWORD(34)="PARA"
STOPWORD(35)="POR"
STOPWORD(36)="QUE"
STOPWORD(37)="QUIEN"
STOPWORD(38)="QUIENES"
STOPWORD(39)="SA"
STOPWORD(40)="SE"
STOPWORD(41)="SI"
STOPWORD(42)="SU"
STOPWORD(43)="SUI"
STOPWORD(44)="TAMBIEN"
STOPWORD(45)="U"
STOPWORD(46)="Y"
```

```
&& Inicialización de la tabla de
&& palabras no significativas
```

```

SELECT 1
USE DIRFAX                                && Base de datos principal
                                           && Directorio empresarial (DIRFAX)

SELECT 2
USE KEYLST                                && Base de datos de terminos
                                           && extraidos

ZAP
INDEX ON KEY TO KEY001.NTX                && indice auxiliar

SELECT 3
USE LNKLS1                                && Base de datos de ocurrencias
                                           && (numeros de registro)

ZAP
INDEX ON KEYNUM+KEYREF TO LNK001.NTX     && indice auxiliar

```

```

KN=0
SELECT 1
CLEAR
@ 1,1 SAY "GENERACION DE ARCHIVO INVERTIDO"
@ 2,1 SAY "OSI - GEN VSN 1.0"
@ 3,1 SAY "-----"
FOR I=4 TO 23
@ I,0 SAY "-----"
NEXT
@ 10,30 SAY "INDEXANDO...ESPERE POR FAVOR..."
@ 11,30 SAY "INDEXING ...PLEASE WAIT ..."
@ 13,30 SAY "PROCESANDO/PROCESING REG/REC.#"
DO WHILE .NOT. EOF()
@ 13,62 SAY RECNO()
DO EXTRAELLAVE WITH "NOMBRE"
DO EXTRAELLAVE WITH "DIRECCION"
DO EXTRAELLAVE WITH "COLONIA"
DO EXTRAELLAVE WITH "CIUDAD"
DO EXTRAELLAVE WITH "ESTADO"
DO EXTRAELLAVE WITH "GIRO"
SKIP
ENDDO
DO SORTLNK
DO LNKGEN
CLOSE DATABASE
CLEAR
?
? "PROCESO DE INDEXACION OSI-IFGEN TERMINADO CON EXITO I "
?
RETURN

/* Extrae terminos de cada palabra que componen los campos
procesados, si existen repeticiones, se anota otra ocurrencia
en el archivo LNKLS1 (entendemos ocurrencia como una palabra
que se repite en uno o mas registros), cada palabra es filtrada
para evitar que ingresen al indice palabras no significativas */

PROCEDURE EXTRAELLAVE
PARAMETERS FLDNAME
LOCAL STW
FLD=UPPER(ALLTRIM(TRIM(&FLDNAME)))
PF=LEN(FLD)
IDX=0
RN=STRZERO(RECNO(),8)
DO WHILE PF>0
PF=AT(CHAR(1),FLD)
LN=LEN(FLD)
IF (PF=0)
IDX=IDX+1
ST=SUBSTR(FLD,1,PF-1)

```

```

DO DEPURAR
KEYTBL(IDX)=SG
FLD=SUBSTR(FLD,PF+1,LN)
ELSE
  IF LN>0
    IDX=IDX+1
    ST=FLD
    DO DEPURAR
    KEYTBL(IDX)=SG
  ENDIF
ENDIF
ENDDO

FOR I=1 TO IDX
  SELECT 2
  KEYTMP=ALLTRIM(SUBSTR(KEYTBL(I),1,30))
  DO CHECKSTOPWORD WITH KEYTMP,STW

  IF !STW
    IF !SEEKEXACT(KEYTMP)
      KN=KN+1
      APPEND BLANK
      REPLACE KEY WITH KEYTMP
      REPLACE KNO WITH STRZERO(KN,6)
      SELECT 3
      APPEND BLANK
      REPLACE KEYNUM WITH STRZERO(KN,6)
      REPLACE KEYREF WITH RN
    ELSE
      KNTMP=KNO
      SELECT 3
      SEEK(KNTMP+RN)
      IF !FOUND()
        APPEND BLANK
        REPLACE KEYNUM WITH KNTMP
        REPLACE KEYREF WITH RN
      ENDIF
    ENDIF
  ENDIF
  SELECT 1
NEXT
RETURN

/* Filtro de palabras no significativas, se utiliza un algoritmo
de búsqueda binaria */
PROCEDURE CHKSTOPWORD
PARAMETERS WORD,FOUNDSTW
LOCAL LI,LF,AP
FOUNDSTW=.F.                && Palabra encontrada=.T. o no=.F.
LF=45                       && Límite máximo de la tabla
LI=1                         && Límite mínimo
AP=ROUND((LI+LF)/2,0)
DO WHILE LI<=LF .AND. !FOUNDSTW
  DO CASE
    CASE STOPWORD[AP]=WORD
      FOUNDSTW=.T.
      EXIT
    CASE STOPWORD[AP]>WORD
      LF=AP-1
    CASE STOPWORD[AP]<WORD
      LI=AP+1
  ENDCASE
  AP=ROUND((LI+LF)/2,0)
ENDDO
RETURN

```

```

/* Elimina caracteres de puntuación */
PROCEDURE DEPURA
  SG=""
  LS=LEN(ST)
  FOR Y=1 TO LS
    CH=SUBSTR(ST,Y,1)
    IF CH<>" " .AND. CH<>"." .AND. CH<>"(" .AND. CH<>")"
      SG=SG+CH
    ENDIF
  NEXT
RETURN

/* Ordenación de archivos */
PROCEDURE SORTLNK
  SELECT 2
  SORT TO KEY001 ON KEY
  CLOSE
  RUN COPY KEY001.DBF KEYLST.DBF >NULL
  RUN DEL KEY001.DBF >NULL
  USE KEYLST
  SELECT 3
  SORT TO LNK001 ON KEYNUM,KEYREF
  CLOSE
  RUN COPY LNK001.DBF LNKLST.DBF >NULL
  RUN DEL LNK001.DBF >NULL
  USE LNKLST
  INDEX ON KEYNUM+KEYREF TO LNK001
RETURN

/* Generación del archivo de ligas, LMKLST el cual contiene los números
de registro de donde fueron extraídas cada una de las palabras indexadas */
PROCEDURE LNKGEN
LOCAL TRMNO,LNKNO,POSTING,POSIN,RECN,KEY
  SELECT 2
  GO TOP
  DO WHILE .NOT. EOF()
    SELECT 2
    TRMNO=KNO
    SELECT 3
    POSTING=0
    SEEK(TRMNO)
    IF FOUND()
      POSIN=RECN()
      KEY=KEYREF
      RECN=1
      DO WHILE TRMNO=KEYNUM
        POSTING=POSTING+1
        IF KEY<=KEYREF
          KEY=KEYREF
          RECN=RECN+1
        ENDIF
        SKIP
      ENDDO
    SELECT 2
    REPLACE PIN WITH POSIN
    REPLACE POS WITH POSTING
    REPLACE REC WITH RECN
  ENDIF
  SKIP
ENDDO
RETURN

```

```
/* Rutina de búsqueda exacta */  
FUNCTION SEEKEXACT(VLLAVE) /*  
  SEEK PADR(VLLAVE,30)  
  RETURN(FOUND())
```

```
"
" GENERACION DE ARCHIVO INVERTIDO
" DAVID DEL MORAL C.
" 19940401
"
SET EXACT ON
PUBLIC RNO,I,FLD,RN
PUBLIC NOMNO,DIRNO,COLNO,CODNO,ESTNO,GIRNO
DECLARE KEYTBL(4096)
DECLARE LNKTB(4096)
SELECT 1
USE DIRFAK

SELECT 2
USE NOMLST
ZAP
INDEX ON KEY TO NOMLST.NTX
CLOSE
SELECT 3
USE NOMLNK
ZAP
INDEX ON KEYNUM+KEYREF TO NOMLNK.NTX
CLOSE

SELECT 4
USE DIRLST
ZAP
INDEX ON KEY TO DIRLST.NTX
CLOSE
SELECT 5
USE DIRLNK
ZAP
INDEX ON KEYNUM+KEYREF TO DIRLNK.NTX
CLOSE

SELECT 6
USE COLLST
ZAP
INDEX ON KEY TO COLLST.NTX
CLOSE
SELECT 7
USE COLLNK
ZAP
INDEX ON KEYNUM+KEYREF TO COLLNK.NTX
CLOSE

SELECT 8
USE CIULST
ZAP
INDEX ON KEY TO CIULST.NTX
CLOSE
SELECT 9
USE CIULNK
ZAP
INDEX ON KEYNUM+KEYREF TO CIULNK.NTX
CLOSE

SELECT 10
USE ESTLST
ZAP
INDEX ON KEY TO ESTLST.NTX
CLOSE
SELECT 11
USE ESTLNK
ZAP
INDEX ON KEYNUM+KEYREF TO ESTLNK.NTX
```

CLOSE

```
SELECT 12
USE GIRLST
ZAP
INDEX ON KEY TO GIRLST.NTX
CLOSE
SELECT 13
USE GIRLNK
ZAP
INDEX ON KEYNUM+KEYREF TO GIRLNK.NTX
CLOSE
```

```
NOMNO=0
DIRNO=0
COLNO=0
CLINO=0
ESINO=0
GIRNO=0
KN=0
SELECT 1
```

```
CLEAR
@ 1,1 SAY "GENERACION DE ARCHIVO INVERTIDO"
@ 2,1 SAY "OSI - GEN VSN 1,0"
@ 3,1 SAY " "
FOR RN=4 TO 23
@ RN,0 SAY " "
NEXT
```

```
@ 10,30 SAY "INDEXANDO...ESPERE POR FAVOR..."
@ 11,30 SAY "INDEXING ...PLEASE WAIT ..."
@ 13,30 SAY "PROCESANDO/PROCESING REG/REC.#"
DO EXTRAELAVE
DO SORTIDX
DO SETPOSTING
```

```
CLEAR
?
? " PROCESO DE INDEXACION OSI-IFGEN2 TERMINADO CON EXITO 1 ."
?
RETURN
```

```
PROCEDURE EXTRAELAVE
DO WHILE IEOF()
SELECT 1
I=0
DO WHILE IEOF() .AND. I<=4000
RNO=STRZERO(RECHO(),8)
@ 13,60 SAY RNO
DO CHECALLAVE WITH "ESTADO", "06",RNO
DO CHECALLAVE WITH "NOMBRE", "01",RNO
DO CHECALLAVE WITH "DIRECCION", "02",RNO
DO CHECALLAVE WITH "COLONIA", "03",RNO
DO CHECALLAVE WITH "CIUDAD", "04",RNO
DO CHECALLAVE WITH "GIRO", "09",RNO
SELECT 1
SKIP
IF I>=4000 .OR. IEOF()
DO GRABALLAVE
ENDIF
```

```
ENDDO
ENDDO
RETURN
```

```
PROCEDURE CHECALLAVE
```

```

PARAMETERS FLDNAME,FLDNUM,RNO
FLD=UPPERCALLTRIM(TRIM(&FLDNAME))
IF LEN(FLD)=0
  I=I+1
  KEYTBL[I]=FLDNUM-RNO=FLD
ENDIF
RETURN

```

```

PROCEDURE GRABLLAVE
LOCAL J,K,FLDNUM,LLAVE,KNOTMP,RN
SELECT 2
USE NOMLST INDEX NOMLST.MTX
SELECT 3
USE NOMLNK INDEX NOMLNK.MTX
FOR J=1 TO 1
  FLDNUM=SUBSTR(KEYTBL[J],1,2)
  IF FLDNUM="01"
    SELECT 2
    USE NOMLST INDEX NOMLST.MTX
    LLAVE=SUBSTR(KEYTBL[J],11,LEN(KEYTBL[J])-10)
    @ 15,10 SAY "KEY:"+LLAVE
    SEEK(LLAVE)
    IF !FOUND()
      NOMNO=NOMNO+1
      KNOTMP=STRZERO(NOMNO,6)
      APPEND BLANK
      REPLACE KEY WITH LLAVE
      REPLACE KNO WITH KNOTMP
    ELSE
      KNOTMP=KNO
    ENDIF
  ENDIF
  SELECT 3

  RN=SUBSTR(KEYTBL[J],3,8)
  SEEK(KNOTMP+RN)
  IF !FOUND()
    APPEND BLANK
    REPLACE KEYNUM WITH KNOTMP
    REPLACE KEYREF WITH RN
  ENDIF
ENDIF
NEXT
SELECT 2
CLOSE
SELECT 3
CLOSE
SELECT 4
USE DIRLST INDEX DIRLST.MTX
SELECT 5
USE DIRLNK INDEX DIRLNK.MTX
FOR J=1 TO 1
  FLDNUM=SUBSTR(KEYTBL[J],1,2)

  IF FLDNUM="02"
    SELECT 4
    LLAVE=SUBSTR(KEYTBL[J],11,LEN(KEYTBL[J])-10)
    SEEK(LLAVE)
    IF !FOUND()
      DIRNO=DIRNO+1
      KNOTMP=STRZERO(DIRNO,6)
      APPEND BLANK
      REPLACE KEY WITH LLAVE
      REPLACE KNO WITH KNOTMP
    ELSE

```

```

      KNOTMP=KNO
    ENDIF
    SELECT 5
    RM=SUBSTR(KEYTBL[J],3,8)
    SEEK(KNOTMP+RM)
    IF 1FOUND()
      APPEND BLANK
      REPLACE KEYNUM WITH KNOTMP
      REPLACE KEYREF WITH RM
    ENDIF
  ENDIF
NEXT
SELECT 4
CLOSE
SELECT 5
CLOSE
SELECT 6
USE COLLST INDEX COLLST.NTX
SELECT 7
USE COLLNK INDEX COLLNK.NTX
FOR J=1 TO I
  FLDNUM=SUBSTR(KEYTBL[J],1,2)
  IF FLDNUM="03"
    SELECT 6
    LLAVE=SUBSTR(KEYTBL[J],11,LEN(KEYTBL[J])-10)
    SEEK(LLAVE)
    IF 1FOUND()
      COLNO=COLNO+1
      KNOTMP=STRZERO(COLNO,6)
      APPEND BLANK
      REPLACE KEY WITH LLAVE
      REPLACE KNO WITH KNOTMP
    ELSE
      KNOTMP=KNO
    ENDIF
  ENDIF
  SELECT 7
  RM=SUBSTR(KEYTBL[J],3,8)
  SEEK(KNOTMP+RM)
  IF 1FOUND()
    APPEND BLANK
    REPLACE KEYNUM WITH KNOTMP
    REPLACE KEYREF WITH RM
  ENDIF
ENDIF
NEXT
SELECT 6
CLOSE
SELECT 7
CLOSE
SELECT 8
USE CIULST INDEX CIULST.NTX
SELECT 9
USE CIULNK INDEX CIULNK.NTX
FOR J=1 TO I
  FLDNUM=SUBSTR(KEYTBL[J],1,2)
  IF FLDNUM="04"
    SELECT 8
    LLAVE=SUBSTR(KEYTBL[J],11,LEN(KEYTBL[J])-10)
    SEEK(LLAVE)
    IF 1FOUND()
      CIUNO=CIUNO+1
      KNOTMP=STRZERO(CIUNO,6)
      APPEND BLANK
      REPLACE KEY WITH LLAVE
      REPLACE KNO WITH KNOTMP
    ENDIF
  ENDIF
ENDIF
NEXT

```

```

ELSE
  KNOTMP=KNO
ENDIF
SELECT 9
RN=SUBSTR(KEYTBL(J),3,8)
SEEK(KNOTMP=RN)
IF !FOUND()
  APPEND BLANK
  REPLACE KEYNUM WITH KNOTMP
  REPLACE KEYREF WITH RN
ENDIF
ENDIF

NEXT
SELECT 8
CLOSE
SELECT 9
CLOSE

SELECT 10
USE ESTLST INDEX ESTLST.WTX
SELECT 11
USE ESTLTK INDEX ESTLTK.WTX
FOR J=1 TO 1
  FLDNUM=SUBSTR(KEYTBL(J),1,2)
  IF FLDNUM="06"
    SELECT 10
    LLAVE=SUBSTR(KEYTBL(J),11,LEN(KEYTBL(J))-10)
    SEEK(LLAVE)
    IF !FOUND()
      ESTNO=ESTNO+1
      KNOTMP=STRZERO(ESTNO,6)
      APPEND BLANK
      REPLACE KEY WITH LLAVE
      REPLACE KNO WITH KNOTMP
    ELSE
      KNOTMP=KNO
    ENDIF
    SELECT 11
    RN=SUBSTR(KEYTBL(J),3,8)
    SEEK(KNOTMP=RN)
    IF !FOUND()
      APPEND BLANK
      REPLACE KEYNUM WITH KNOTMP
      REPLACE KEYREF WITH RN
    ENDIF
  ENDIF
NEXT
SELECT 10
CLOSE
SELECT 11
CLOSE

SELECT 14
USE GIRLST INDEX GIRLST.WTX
SELECT 15
USE GIRLTK INDEX GIRLTK.WTX
FOR J=1 TO 1
  FLDNUM=SUBSTR(KEYTBL(J),1,2)
  IF FLDNUM="09"
    SELECT 14
    LLAVE=SUBSTR(KEYTBL(J),11,LEN(KEYTBL(J))-10)
    SEEK(LLAVE)
    IF !FOUND()
      GIRNO=GIRNO+1
      KNOTMP=STRZERO(GIRNO,6)

```

```

APPEND BLANK
REPLACE KEY WITH LLAVE
REPLACE KNO WITH KNOTMP
ELSE
  KNOTMP=KNO
ENDIF
ENDIF
SELECT 15
RN=SUBSTR(KEYTBL(JJ,3,8)
SEEK(KNOTMP*RN)
IF FOUND()
  APPEND BLANK
  REPLACE KEYNUM WITH KNOTMP
  REPLACE KEYREF WITH RN
ENDIF
ENDIF
NEXT
SELECT 14
CLOSE
SELECT 15
CLOSE
SELECT 1
RETURN

```

```

PROCEDURE SORTIDX
SELECT 2
USE NOMLST
SORT TO NOM001.DBF ON KEY
CLOSE
RUN COPY NOM001.DBF NOMLST.DBF >NULL
SELECT 3
USE NOMLNK
SORT TO NOM002.DBF ON KEYNUM,KEYREF
CLOSE
RUN COPY NOM002.DBF NOMLNK.DBF >NULL

SELECT 4
USE DIRLST
SORT TO DIR001.DBF ON KEY
CLOSE
RUN COPY DIR001.DBF DIRLST.DBF >NULL
SELECT 5
USE DIRLNK
SORT TO DIR002.DBF ON KEYNUM,KEYREF
CLOSE
RUN COPY DIR002.DBF DIRLNK.DBF >NULL

SELECT 6
USE COLLST
SORT TO COL001.DBF ON KEY
CLOSE
RUN COPY COL001.DBF COLLST.DBF >NULL
SELECT 7
USE COLLNK
SORT TO COL002.DBF ON KEYNUM,KEYREF
CLOSE
RUN COPY COL002.DBF COLLNK.DBF >NULL

SELECT 8
USE ESTLST
SORT TO EST001.DBF ON KEY
CLOSE
RUN COPY EST001.DBF ESTLST.DBF >NULL
SELECT 9
USE ESTLNK
SORT TO EST002.DBF ON KEYNUM,KEYREF

```

```
CLOSE
RUN COPY EST002.DBF ESTLNK.DBF >NULL

SELECT 10
USE CIULST
SORT ON KEY TO CIU001.DBF
CLOSE
RUN COPY CIU001.DBF CIULST.DBF >NULL
SELECT 11
USE CIULNK
SORT ON KEYNUM,KEYREF TO CIU002.DBF
CLOSE
RUN COPY CIU002.DBF CIULNK.DBF >NULL
SELECT 14
USE GIRLST
SORT TO GI001.DBF ON KEY
CLOSE
RUN COPY GI001.DBF GIRLST.DBF >NULL
SELECT 15
USE GIRLNK
SORT TO GI002.DBF ON KEYNUM,KEYREF
CLOSE
RUN COPY GI002.DBF GIRLNK.DBF >NULL
RETURN
```

PROCEDURE SETPOSTING

```
SELECT 3
CLOSE
USE NOMLNK
SELECT 2
CLOSE
USE NOMLST
DO LOADPOSTING
```

```
SELECT 3
CLOSE
USE DIRLNK
SELECT 2
CLOSE
USE DIRLST
DO LOADPOSTING
```

```
SELECT 3
CLOSE
USE COLLNK
SELECT 2
CLOSE
USE COLLST
DO LOADPOSTING
```

```
SELECT 3
CLOSE
USE CIULNK
SELECT 2
CLOSE
USE CIULST
DO LOADPOSTING
```

```
SELECT 3
CLOSE
USE ESTLNK
SELECT 2
CLOSE
USE ESTLST
DO LOADPOSTING
```

```
SELECT 3
CLOSE
USE COOLMK
SELECT 2
CLOSE
USE COOLST
DO LOADPOSTING
```

```
SELECT 3
CLOSE
USE GIRLNM
SELECT 2
CLOSE
USE GIRLST
DO LOADPOSTING
RETURN
```

```
PROCEDURE LOADPOSTING
LOCAL FLD, POSIC, RECNU, FND, POSMU, KEYREFTMP
SELECT 2
GO TOP
POSMU=0
RECNU=0
DO WHILE 'EOF()'
  FLD=KNO
  SELECT 3
  FND=.F.
  DO BSEARCH WITH FLD, FND, POSIC
  IF FND
    GO POSIC
    POSMU=0
    RECNU=1
    KEYREFTMP=KEYREF
    DO WHILE KEYNUM=FLD
      POSMU=POSMU+1
      IF KEYREF<>KEYREFTMP
        RECNU=RECNU+1
        KEYREFTMP=KEYREF
      ENDIF
      SKIP
    ENDDO
    SELECT 2
    REPLACE PIN WITH POSIC
    REPLACE POS WITH POSMU
    REPLACE REC WITH RECNU
  ENDIF
  SELECT 2
  SKIP
ENDDO
RETURN
```

```
PROCEDURE BSEARCH
PARAMETERS TERM, FND, AP
LOCAL TMP
SELECT 3
GO BOTTOM
LF=RECNO()
GO TOP
LI=1
FND=.F.
AP=ROUND((LI+LF)/2, 0)
GO AP
DO WHILE LI<=LF .AND. !FND
  DO CASE
    CASE KEYNUM=TERM
```

```
FND=.T.
EXIT
CASE KEYNUM<TERM
  LF=AP-1
CASE KEYNUM<TERM
  LI=AP+1
ENDCASE
AP=ROUND((LI+LF)/2,0)
GO AP
ENDDO
IF FND
DO WHILE .T.
  TMP=AP-1
  GO TMP
  IF KEYNUM=TERM
    AP=TMP
  ELSE
    EXIT
  ENDF
ENDDO
ELSE
GO TOP
DO WHILE .NOT. EOF()
  IF KEYNUM=TERM
    FND=.T.
    AP=ROUND()
    EXIT
  ENDF
  SKIP
ENDDO
ENDIF
RETURN
```

```

1
2
3
4 Archivo
5 Búsqueda
6 Opciones
7 Archivo
8 Impresión
9 Salvar
10 Salir
11 Cartas
12 Etiquetas
13 Reportes
14 Búsqueda
15 Todos
16 Nombre
17 Dirección
18 Colonia
19 Ciudad
20 Estado
21 Giro
22 *Inicializando el Sistema, por favor espere ...
23 Opciones
24 Idioma
25 Colores
26 Inglés
27 Español
28 Operador
29 Registros procesados:
30 Aciertos última búsqueda:
31 Localizar por:
32 F9 - Historia      F10 - Mostrar
33 Número Hits      Op Llave      Campo
34 Llave
35 Occ
36 FB - Imagen
37 Imprimiendo el registro #: ...
38 Nombre de archivo a imprimir?:
39 Grabando el registro #:
40 Precaución: Imposible retornar al modo texto, código:
41 Precaución: Imposible mostrar gráficos, código:
42 OPCIONES: M=Archivo nuevo      L=Archivo Existente      X= salir
43 Nombre del archivo/carta (máx. 8 caracteres):
44 Ingrese el nombre del formato de carta <Enter> Salir
45 Archivo no encontrado
46 EDICION DE CARTAS
47 Seleccione una de las siguientes opciones:
48 M = Pág. siguiente B = Pág. anterior X = Salvar y Salir C = Cancelar
49 Salir (S/M):
50 Ingresar Texto:
51 <PgDown> = Menú      <ESC> = Salir
52 EDITAR OPCIONES:
53 F1 = Mostrar Ayuda
54 F2 = Mostrar Menú de Campos
55 F10 = Grabar el texto en disco
56 MOVIMIENTO DEL CURSOR:
57 <- -> Desplaza el Cursor una posición
58 <Ctrl> and <- -> (Ctrl and flechas) = Desplazamiento por palabra
59 <Pg Down> = Avanza una página
60 <Pg Up> = Retrocede una Página
61 MACROS:
62 Debe encerrar el campo entre "[...]" para indicar una macro Ejem. [nombre]
63 o presionar F2 para mostrar lista de campos
64 Nombre de Archivo (máx 8 caracteres):

```

65 Archivo existente, sobrescribir (S/N) ? :  
66 PRECAUCION : Error al crear el archivo  
67 Seleccionar un campo de la lista - Use las flechas para desplazarse  
68 Seleccionar el conjunto a imprimir  
69 #CONJ NITS OP LLAVE CAMPO  
70 Error: No se ha seleccionado el conjunto a imprimir...  
71 ETIQUETAS  
72 Ingrese los siguientes datos :  
73 Ancho (en columnas) :  
74 Líneas por etiqueta :  
75 Margen izquierdo :  
76 Espacios entre cada etiqueta :  
77 Etiquetas por línea :  
78 Error : parametro invalido(s) - <ESC> Salir <Enter> - Modificar parametros  
79 Tamaño de la etiqueta :  
80 Seleccione los campos a imprimir  
81 Oprima <F2> para mostrar la lista de campos  
82 ERROR : imposible abrir archivo de mensajes  
83 Inicializando el sistema, por favor espere ...  
84 Inicio  
85 ERROR : imposible abrir archivo de especificaciones de impresión  
86 Res  
87 No se ha seleccionado el conjunto a grabar...  
88 <free>  
89 <free>  
90 <free>  
91  
92 <Avilable>  
93 <Avilable>  
94 <Avilable>  
95 <Avilable>  
96 <Avilable>  
97 <Avilable>  
98 <Avilable>  
99 <Avilable>  
100 <Avilable>

```

1
2
3
4 File
5 Search
6 Options
7 File
8 Print
9 Save
10 Exit
11 Letters
12 Mail Tags
13 Reports
14 Search
15 All
16 Name
17 Address
18 Area
19 City
20 State
21 Business
22 *Starting environment, please wait ...
23 Options
24 Language
25 Colors
26 English
27 Spanish
28 Operator
29 Processed records :
30 Last Search Hits :
31 Find for :
32 F9 - History      F10 - Show hits      Field
33 Set Hits Op Key
34 Key
35 Posting
36 FB - Imager
37 Printing record # :
38 Print file name ? :
39 Writing record # :
40 WARNING : Unable to return to text mode, code :
41 WARNING : Unable to show graphics, code :
42 OPTIONS : N=New file L=Load a file X=Exit
43 Letter file name (max. 8 characters) :
44 Type letter file name or <Enter> to exit
45 file not found
46 LETTER EDITION
47 Select one of the next options :
48 N = Next page B = Previous Page X = Save and Exit A = Abort C = Cancel
49 Abort (Y/N) :
50 Type your text :
51 <PgDown> = Menu      <ESC> = Abort
52 EDIT OPTIONS :
53 F1 = Show this help
54 F2 = Show fields menu
55 F10 = Save your text to disk
56 CURSOR MOVEMENT :
57 <- -> Cursor arrows one position each
58 <Ctrl> and <... -> (Ctrl and cursor arrows) = one word each
59 <Pg Down> = Scroll a page
60 <Pg Up> = Previous page
61 MACROS :
62 The field to macro must be with "M" before each e.g. @nombre
63 or press F2 to show the field list
64 File name (max 8 characters) :

```

65 File exist, overwrite (Y/N) ? :  
66 WARNING : file creation error  
67 Select a field from the list - use arrows to move  
68 Select a set to print  
69 SET HITS OP KEY  
70 No SET to print !  
71 MAIL TAGS  
72 Type the next parameters :  
73 Wide (in columns) :  
74 Lines per tag :  
75 Left margin :  
76 Spaces between each tags :  
77 Tags per line :  
78 Error : invalid parameter(s) - <ESC> Abort <Enter> - Edit parameters  
79 The tag size is as is show ;  
80 Select the fields to print  
81 Press <F2> to show the fields list  
82 FATAL ERROR : I can't open message file  
83 Starting environment, please wait ...  
84 Start  
85 ERROR : it can't to open specification print file  
86 Rec  
87 No records to save....  
88 <Free>  
89 <Free>  
90 <Free>  
91  
92 <Available>  
93 <available>  
94 <Available>  
95 <Available>  
96 <Available>  
97 <Available>  
98 <Available>  
99 <Available>  
100 <Available>

## BIBLIOGRAFÍA

## BIBLIOGRAFIA

---

- **BYTE**  
Guide to CD-ROM  
Osborne/Mc Graw Hill  
De. 1994, United States of América  
ISBN = 0-07-881982-2
- **CONACYT SECOBI**  
Introducción a los sistemas recuperadores de información (IRAI).
- **Curiel, Alberto**  
Discos Compactos  
Revista Micromonitor.  
Ed. mayo/junio 1993 año 2 No. 13, México, D.F.
- **Dialog Information Services Inc. U.S.A.**  
Dialog Curso Básico  
Ed. 1985.
- **Miller, Rofley**  
CD-ROM facts & figures 95  
TFPL Publishing  
Ed. 1995, Washington, DC U.S.A.  
ISBN 1 870889 55 X
- **Folk, Michael; Zoellick, Bill**  
File Structures  
Addison-Wesley Publishing Company, Inc.  
2nd, Ed. 1992, U.S.A.  
ISBN 0-201-55713-4
- **García, José Javier -Badell**  
Clipper 5.01 a su alcance. Serie Mc. Graw Hill de Informática  
Mc. Graw Hill  
1a. Ed. 1993, México, D.F.  
ISBN 84-7615-918-8
- **González, Fernando; Dominguez, Javier**  
Automatización de Bibliotecas - Sistemas Disponibles en México  
Serie Manuales 12  
1a. Ed. 1990, México  
UNAM-CUIB  
ISBN 968-36-1213-X

## **BIBLIOGRAFIA**

---

- **Norris, Mark**  
Ingeniería de Software Explicada  
Noriega-Megabyte  
1a. Ed. 1989, México, D.F.  
ISBN=968-18-5028-9
- **Kirby, Steven**  
Frequently Asked Questions List for CDROMLAN  
University of Georgia  
Ed. feb 1995,  
Localizado en el WWW [saturn.uaamath.alaska.edu/gibbsg/cdromlan\\_fa.html](http://saturn.uaamath.alaska.edu/gibbsg/cdromlan_fa.html)  
University of Alaska
- **Kruse, Robert L.**  
Data Structures & Program Design  
Prentice Hall  
3a. Ed.
- **Langberg, Mike**  
CD-ROM attracts heavy hitters; creative types dabble in new worlds created by computer.  
(Compact Disc-Read Only Memory)  
San Jose Mercury News March 2, 1994 p1F(1)
- **Swartz, Jhon**  
Rental plan aims to make CD-ROM a blockbuster hit with consumers -  
(Blockbuster Entertainment Corp. will demonstrate and rent CD-ROM players and programs)  
MacWEEK Nov 15, 1993 v7 n45 p44(2)  
Coastal Associates Publishing L.P. 1993
- **Tiley, Edward**  
Using Clipper  
Programming Series  
3a. Edition  
Que Corporation
- **Thunderstone Inc.**  
Overview of a Full Text Search an Retrieval Database Manager  
Localizado en el WWW [www.thunderstone.com](http://www.thunderstone.com)  
Actualización al 23 de febrero 1996
- **(UNESCO) División of software Development and Applications Office of Information Programs and Services**  
Mini-Micro CDS/ISIS Pascal (Versión 2.3)  
UNESCO

## **BIBLIOGRAFIA**

---

1a. Ed. 1989, París, Francia  
ISBN 92-3-102-605-7

- **Veronis, Suhler**  
Interactive media is major growth sector (Digital Information Group Associates Inc., and Cunningham Communications)  
Digital Media Jan 31, 1994 v3 n8 p24(2)  
Seybold Publications Inc. 1994
  
- **Zugasti, Maritza; Salazar, Edgar.**  
Sistemas Recuperadores de Información Bibliográfica  
Universidad Nacional Autónoma de México  
Facultad de Ingeniería  
Tesis Profesional  
1987, México
  
- **Información: Producción, Comunicaciones y Servicios.**  
Año 3. Número 1. 1993  
Infoconsultores S.C. México, D.F.  
ISSN 0188-5847  
Pag. 8-14; 15-23
  
- **Información: Producción, Comunicaciones y Servicios.**  
Año 3. Número 2. 1993  
Infoconsultores S.C. México, D.F.  
ISSN 0188-5847  
Pag. 26-32.