001.555
V473

BIBLIOTECA

Escuela Superior Politécnica del Litoral

FACULTAD DE INGENIERIA
EN ELECTRICIDAD

DISEÑO E IMPLEMENTACION DE UN SISTEMA PARA MANEJO INTELIGENTE DE DOCUMENTOS

TESIS DE GRADO

Previa a la Obtención del Título de

INGENIERO EN COMPUTACION

PRESENTADA POR:

DUSYA MARIA VERA BONILLA

Guayaquil - Ecuador

1992

ESPOL
CIB
D-10932

ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD

"DISEÑO E IMPLEMENTACION DE UN SISTEMA PARA MANEJO INTELIGENTE DE DOCUMENTOS"

TESIS DE GRADO

Previa a la Obtención del Título de

INGENIERO EN COMPUTACION

FRESENTADA FOR:

Dusya María Vera Bonilla

Guayaquil - Ecuador

1.992

AGRADECIMIENTO: Mi infinita gratitud al Ing. **Carlos** Valero Delgado, Director de Tesis, por su constante preocupación y excelente disposición a lo **largo** del desarrollo de la presente Tesis de Grado.

Gracias también al Ing. Sixto **Garcia** A., Coordinador del Area Académica Computación, por su generosa colaboración para la feliz culminación de esta Tesis de Grado.

DEDICATORIA: A cada miembro de mi Familia, **por** compartir conmigo todo momento y constituir el motor vital de mi existencia.

En especial, a mis Padres. Su cariño y confianza es el **mayor** compromiso de mi vida.

DECLARACION EXPRESA

"La responsabilidad por los hechos. ideas y doctrinas expuestas en esta Tesis, me corresponden exclusivamente; y, el patrimanio intelectual de la misma a la Escuela Superior Politécnica del Litoral"

(Reglamento de Exámenes y Titulos Profesionales)

Dusya Vera B

Dusya Vera Bonilla

Subdecano de la Facultad de
Ingeniería en Electricidad
Ing. Jorge Flores Macias

Director de Tesis
Ing. Carlos Valero Delgado

Miembro del Tribunal
Ing. Sixto Garcia A.

Miembro del Tribunal
Ing. Jaime Puente P.

INDICE

INTRODUCCION IX

CAPITULO I. HISTORIA DE LAS BASES DE DATOS.

1.1 Generalidades. 12

1.2 Bases de Datos Inteligentes. 16

1.3 Tecnologias Complementarias a las BDI. . . . 17

1.3.1 Sistemas Expertos 21
1.3.2 Hypertext y Hypermedia 25
1.3.3 Orientación a Objeto 29
1.3.4 Manejo de Textos y Recuperación de Información 30

1.4 Arquitectura de una Base de Datos Inteligente. 30

1.4.1 Inteligencia en el Nivel de las Herramientas de Alto Nivel. 33
1.4.2 Inteligencia en el Nivel de las Interfases con el Usuario. 36
1.4.3 Inteligencia en el Nivel de Soporte de la Base de Datos. 38

CAPITULO II. RECUPERACION DE INFORMACION Y MANEJO DE TEXTOS.

2.1 Recuperación de Información. 41

2.1.1 El Desarrollo de la Tecnología de la Información. 42
2.1.2 Caracteristicas de la Recuperación de Información. 43
2.1.3 Definición de la Recuperación de Información. 46

2.1.4 Indexación manual y automatizada. . 50
2.1.5 Intermediario de Búsquedas. 56
2.1.6 Formulación de Búsquedas. 60
2.1.7 Retroalimentación de Relevancia. . 63
2.1.8 Evalución de la Recuperación de Información. 68

2.2 Manejo de Textos. 71

2.2.1 Manejo de Textos vs Recuperación de Información. 72
2.2.2 Indexación Conceptual. 74

2.3 Software Existente para Manejo de Documentos. 86

2.3.1 Metodología de Prueba de los Manejadores de Documentos. 90
2.3.2 Características de los Productos. . 94
2.3.3 Conclusiones. 109

CAPITULO III. SISTEMA PERSONAL DE MANEJO DE DOCUMENTOS.

3.1 Alcance del Sistema. 110

3.2 Descripción General y Componentes Básicos del Sistema. 111

3.2.1 Archivos de la Base de Datos. . . . 129

3.3 Sistema Manejador de la Base de Datos. . . . 131

3.3.1 Lenguaje de Definición de Datos de CDATA. 131
3.3.2. Archivo CDATA.H. 140
3.3.3. Formato de los Archivos de CDATA. . 142
3.3.4. Arquitectura de CDATA. 145

3.4 Descripción e Implementación de Módulos. . . 147

3.5 Manual del Usuario. 192

CAPITULO IV. FUNDAMENTOS DEL LENGUAJE C++.

4.1 La Programación Orientada a Objeto. 231

4.1.1 Objetos. 232
4.1.2 Polimorfismo. 233
4.1.3 Herencia. 233

4.2 Introducción al Lenguaje C++. 234

4.2.1 Entrada y Salida. 235
4.2.2 Clases. 237
4.2.3 Sobrecarga de Funciones y Operadores. 242
4.2.4 Herencia. 245
4.2.5 Constructores y Destructores. . . . 249
4.2.6 Funciones friend. 253

4.3 Implementación de Interfases Gráficas en C++. 258

4.3.1 Conceptos básicos sobre IGUS. . . . 259
4.3.2 Componentes de una IGU. 261

CONCLUSIONES Y RECOMENDACIONES 269

BIBLIOGRAFIA 273

INTRODUCCION

En la actualidad vivimos la "Edad de la Información", caracterizada en primer lugar, por la enorme cantidad de informacion disponible a nosotros en formato electrónico, y en segundo lugar, por las notables facilidades brindadas tanto para su almacenaje como para su recuperación. Sin embargo, el ritmo de generación de nueva información en las distintas areas, parece estar superando al de la evolución de las técnicas de recuperación de esa información.

La obtención de soluciones adecuadas a eate problema se vuelve cada dia mas prioritaria, pues la recuperación de la información contenida en Textos y Publicaciones es una necesidad fundamental de los individuos. A estos les resulta difícil saber o recordar, durante el desarrollo de alguna tarea, dónde se encuentra la información acerca de su problema particular.

En los últimos 30 años, una tecnologia que se ha desarrollado ampliamente y que ha contribuido notablemente a las Tareas de Recuperación de Información, es la de Inteligencia Artificial (IA). Ella tiene el potencial de revolucionar las estrategias de indexación y los métodos

usados comúnmente para la recuperación en línea. La Inteligencia Artificial es capaz de representar los Documentos en términos de Conceptos, y por lo tanto tiene un alcance mucho mayor que el de las técnicas tradicionales. Cuando esta tecnología se combina con la Tecnología de las Bases de Datos se obtiene un producto realmente eficaz en la resolución del problema planteado y conocido simplemente como Bases de Datos Inteligentes.

El presente trabajo de Investigación tiene por objeto describir las tecnicas de Manejo de Textos y de Recuperación de Información, empleadas en las Bases de Datos Inteligentes.

Las tecnicas descritas son ejemplificadas mediante el Diseño e Implementación de un Sistema para Manejo Inteligente de Documentos. El Sistema desarrollado es capaz de accesar en forma inteligente la información contenida en una Base de Datos Central, y puede ser utilizado a nivel de Bibliotecas o simplemente como herramienta de uso personal.

Hemos incluido una descripción detallada de nuestro "Sistema Personal de Manejo de Documentos" que incluye

objetivos, alcance, analisis, diseno. implementación y Manual del Usuario.

Se presenta también, a nivel Tutorial, los fundamentos de la Programación Orientada a Objeto y del Lenguaje de Programación C++. Se incluyen técnicas y procedimientos escritos en Lenguaje C++ que demuestran su poder en la implementación de adecuadas interfases graficas con el usuario.

Finalmente, se anexa un Prototipo que, utilizando procedimizntos en Lenguaje C++, implementa una interfase grafica que simula los Menús del "Sistema Personal de Manejo de Documentos". Este Prototipo constituye el punto de partida para el posterior desarrollo de una versión grafica del nuestro Sistema de Documentos.

CAPITULO I

HISTORIA DE LAS BASES DE DATOS

1.1 GENERALIDADES.-

Las ***Bases de Datos Convencionales*** tuvieron su origen en la necesidad de los Programadores de encontrar una manera conveniente de manejar los datos utilizados por los programas. Para algunas aplicaciones, los datos eran de tal importancia que fue necesario desarrollar programas especializados que los pudieran manejar y actualizar.

Adicionalmente a las **Bases** de Datos que satisfacieron las necesidades de los Programadores. otro tipo de Base de Datos fue desarrollado para almacenar la gran cantidad de información contenida en Textos. Estas son las ***Bases de Datos Textuales***, las cuales representan típicamente a los documentos a través de citas. resiimenes y palabras claves. Constituyen una nueva dirección en la tecnología de la información Primero y obviamente, proveen textos en línea. En contraste con un libro convenciunal. una única copia electrónica de una fuente textual puede ser consultada desde muchas localidades el misrno día. Su segundo efecto es que proporcionan una manera flexible de organizar la información. A diferencia de un Libreria. donde los libros es-

tan ordanados secuencialmente de acuerdo al Tópico, en una Base de Datos Textual, gracias a los palabras claves indexadas. un mismo texto es accesado de diversas maneras.

Dentro del campo de la Recuperación de Información, cuatro tecnologias computacionales han tenido gran implicación [PCKW89]:

A. **Almacenamiento Masivo:** La evolución de la tecnología del disco ha reemplazado los pocos registros de memoria de la primeras computadoras. con Gigabytes de información contenida en dispositivos ópticos de alta velocidad.

B. **Exploración Optica (Optical Scanning) y Publicaciones Electrónicas:** Una barrera para el desarrollo de grandes Bases de Datos ha sido siempre la dificultad de transferir versiones impresas de textos a sus correspondientes versiones electronicas. Esta dificultad se vio superada gracias a dos tecnologias. En primer lugar, la Exploración Optica permite textos impresos ser traducidos a medios electrónicos muy rápidamente. Además, la mayoría de las publicaciones de hoy son editadas rnediante Publicadores Electronicos (Ej: Procesadores de Palabras), de manera yue las versiones

electrónicas de los documentos suelen preceder a las versiones impresas. Asi pues, las Bases de Datos pueden crecer mediante la transferencia electrónica de archivos de documentos desde un Editor al Manejador de Bases de Datos.

C. **Redes:** Una de las mas significativas aplicaciones a las Rases de Datos y la Recuperacidn de Información es la conexión de sistemas computacionales a través de LANs y WANs. La conectividad y compartición de datos entre sistemas de computación ha evolucionado desde el correo electrónico y los servidores de archivos en LANs. hasta las Bases de Datos heterogéneas con multiple:: plataformas y modelos de datos.

D. **Bases de Datos en Línea:** A lo largo de los siglos, los libros representaron el medio mas común para archivar el conocimiento del mundo. Sin embargo. a medida que las capacidades para construir Eases de Datos Textuales han evolucionado junto con los relativamente baratos medios de almacenamiento masivo. largas Rases de Datos en Linea han sido desarrolladas, conteniendo muchos millones de documentos de varios tipos.

Muchas Bases de Datos en Línea sólo incluyen información sumaria (Ej: resúmenes de los documentos). Sin embargo, la creciente disponibilidad de almacenamiento electrónico permite hoy el desarrollo de Eases de Datos basadas en Textos Completos (Full-Text). Esta implementacidn necesita todavia superar algunas dificultades. Los métodos tradicionales de indexación son limitados en cuanto a capacidad y tiempo de acceso de toda esta información. Además, aun con el indexamierto adecuado. saber cómo encontrar la información usando los actuales Sistemas de Recuperación es una tarea difícil. Idealmente. los usuarios dehen ser capaces de decir qué quieren, sin tener que preocuparse por supervisar el proceso de búsqueda de la información. También sería de desear que el Sistema de Recuperación tome en cuenta quién es el usuario y qué pueda estar buscando, de manera que la información recuperada esté enfocada en sus necesidades.

Todo lo nombrado ha dado lugar al desarrollo de una nueva tecnología, la de las ***Bases de Datos Inteligentes (BDI)***.

1.2 BASES DE DATOS INTELIGENTES (BDI).-

Las BDI pueden ser definidas como Bases de Datos que manejan información en forma natural. facilitando su almacenamiento, acceso y utilización [PCKW89]. De esta manera. permiten la recuperación de información en una forma más intuitiva para los usuarios.

Entre sus principales funciones y características se nombran :

A. Manejar grandes cantidades de información.

B. Proveer herramientas de alto nivel para el análisis y control de la integridad de los datos, permitiendo a los usuarios extraer conocimiento ds, y aplicar conocimirnto a, los datos.

C. Permitir a los usuarios interactuar directamente con la información en forma natural, como si se estuviera navegando a través de las páginas de un texto estándard o dialogando con un experto humano.

D. Proveer información a un mayor numero de personas. Más de ellas pueden utilizar el Sistema debido a su facil uso.

E. Recuperar conocimiento y no tan sólo datos puros. Esto es, utilizar métodos de inferencia para determinar qué realmente necesita saber el usuario.

F. Mejorar el proceso de toma de decisiones. Esto se logra, proveyendo información elaborada, en otras palabras: conocimiento. Desde esta perspectiva. las BDI sirven como Sistemas de Soporte de Decisiones.

G. Interrelacionar informaci6n de diversas fuentes usando diferentes medios (Ejs: imágenes, sonido, texto, etc), de manera que la información sea mas facilmente absorbida y utilizada por el usuario.

Asi pues, una Base de Datos Inteligente satisface las necesidades del usuario en cunnto a información relevante, facil de obtener y util en la toma de decisiones.

1.3 TECNOLOGIAS COMPLEMENTARIAS A LAS BDI.-

Las BDI son el resultado de la integración de las Bases de Datos Tradicionales y recientes areas de estudio como son:

- Sistemas Expertos.
- Hypermedia.

- Programación Orientada a Objeto.
- Manejo de Textos Recuperación de Informacion.

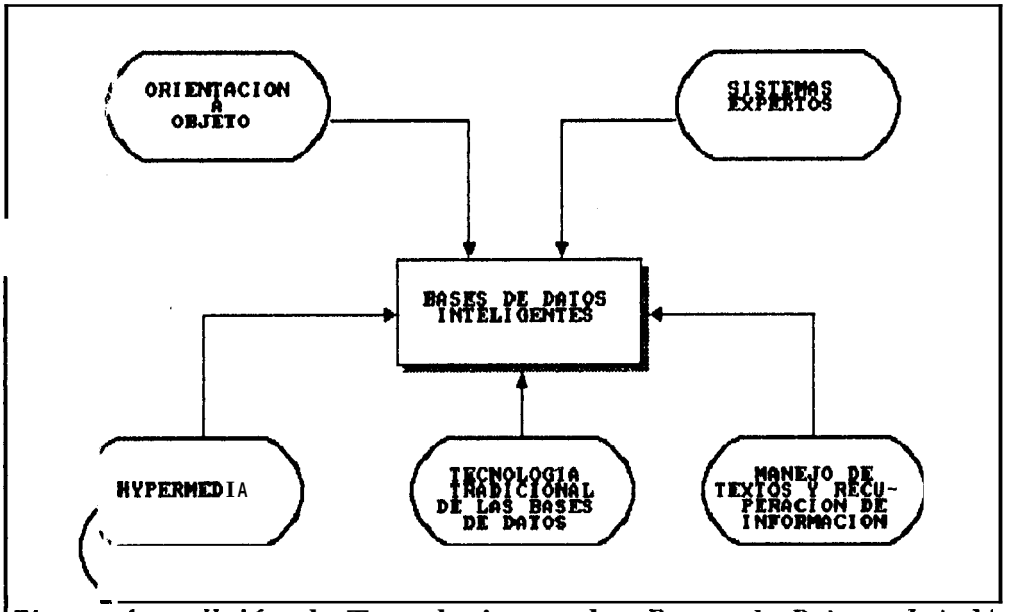


Figura 1. Unión de Tecnologias en las Bases de Datos Inteligentes.

Hasta hace poco, cada una de estas tecnologías habían sido tratadas aisladamente, cada una ligada débilmente a las otras. Esto se debió parcialmente al rápido crecimiento de cada campo. de tal manera que las conexiones y correspondencias para con los otros campos no tuvieron tiempo de formarse. Actualmente. cuando cada tecnología ha alcanzado un cierto estado de madurez, es posible definir una estructura unificadora que ve a cada campo como una parte del mapa general de las Bases de Datos Inteligentes.

Las Figuras 2 y 3 presentan un resumen de la evolución de los campos de estudio nombrados. partiendo de los **Sistemas Tradicionales de Bases de Datos.** Estos. fueron desarrollados en ambientes de mainframes. organizados en registros y típicamente aplicados a datos alfanumericos. En los años 1970 los **Sistemas de Recuperación de Información se** hicieron disponibles y se empezó a trabajar con Rases de Datos Textuales. A mediados de los 70´s la tecnología de **Sistemas Expertos** empezó **a** desarrollarse. llegando a tener prominencia para fines de esa década. Similarmente, a fines de los *70's* y principios de los 80´s. técnicas relacionadas con **Programación Orientada a Objeto** crecieron en las áreas de ingenieria de software. interfases con el usuario y lenguajes de programación de alto **nivel.** Finalmente. a mediados de los 80´s ya estaban disponibles sistemas de **Hypermedia,** los cuales tratan con información en una variedad de formas tales como textos. imágenes y sonido.

A continuación analizaremos Grevemente cada tecnología complementaria a las BDI.

Básicamente, como toda Base de Datos, una BDI incluye: un Modelo de Datos. un Sistema de Indexación. un Lenguaje de Búsqueda, entre otros componentes. Sin embargo. una

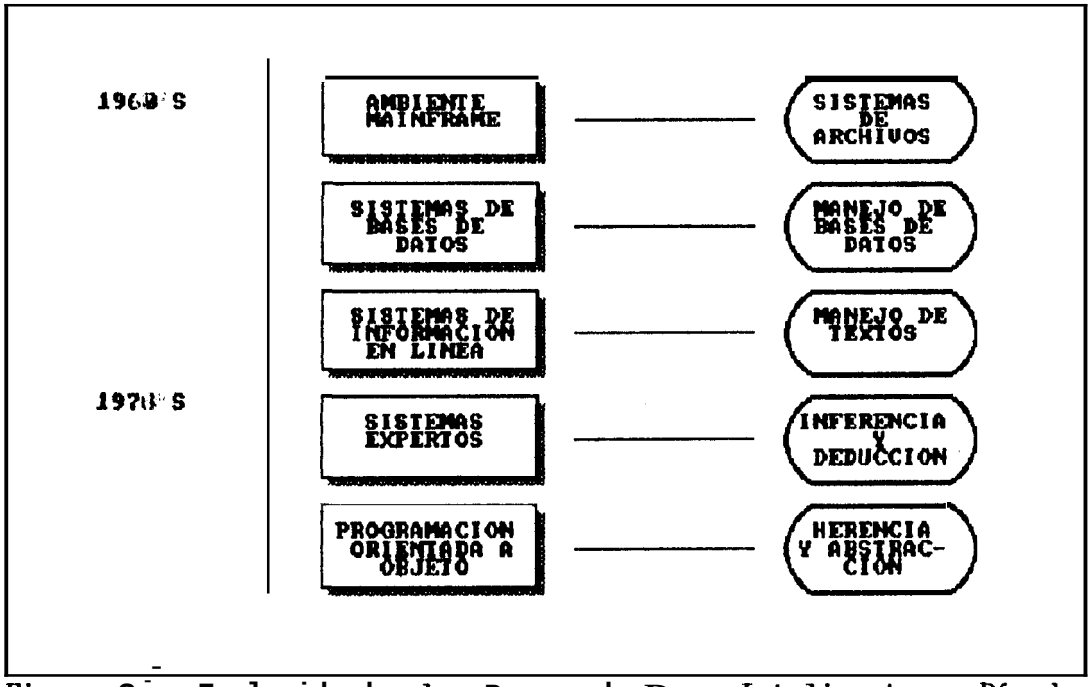


Figura 2. Evolucidn de las Bases de Datos Inteligentes: Décadas 60 *y* 70.

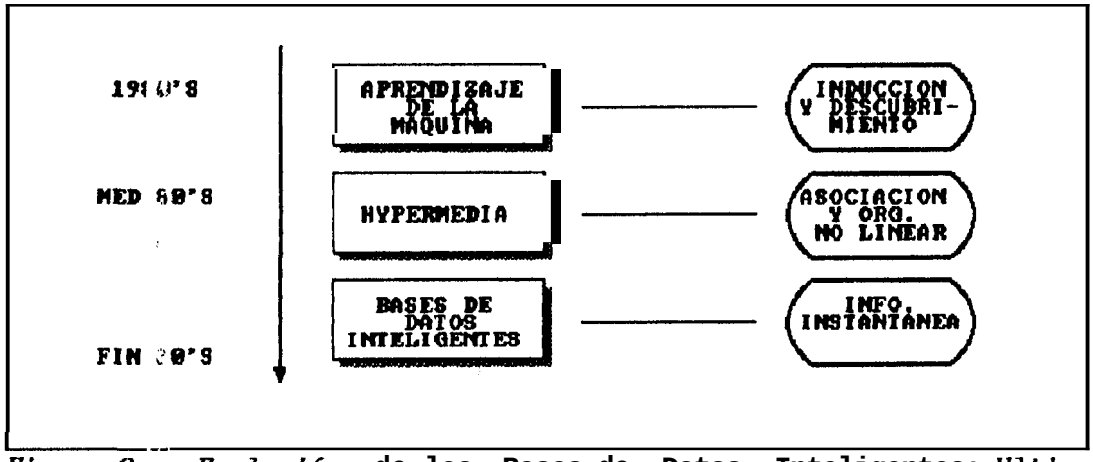


Figura 3. Evolución de las Bases de Datos Inteligentes: Ultima Década.

importante diferencia entre las *Bases de Datos Convencionales y* las Bases de Datos Inteligentes es que las primeras son generalmente estáticas: son actualizadas de tiempo en

tiempo con nueva información, pero este proceso de actualización es externo. no una tarea intrínseca llevada a cabo por el sistema de Bases de Datos mismo.

1.3.1 SISTEMAS EXPERTOS.-

La evolución de los lenguajes de las Bases de Datos está enlazada con la evolución de los lenguajes de programación. Los primeros lenguajes de Bases de Datos utilizaban estructuras de control procedurales, propias de los lenguajee de programación tradicionales como FORTRAN y FASCAL. Fue la tecnología de los Sistemas Expertos la que presentó un conjunto nuevo de paradigmas de programación gracias al desarrollo de nuevas estructuras de datos y nuevas estructuras de control. Vemos así que la siempre creciente necesidad de semántica en los lenguajes de Rases de Datos es satisfecha usando la tecnología de los Sistemas Expertos (SE).

Además, los Sistemas Expertos incrementan el dinamismo de las Rases de Datos. Al proveer la capacidad de razonar e inferir nueva información, permiten la automodificación de las Eases de Datos y la habilidad de analizar qué información interesa a un usuario, aun cuando su necesidad no haya sido expresada de esa manera.

Pero, definamos más exactamente a los Sistemas Expertos, para así poder profundizar en su enlace con las Rases de Datos. Un Sistema Experto (SE) puede ser definido como un programa que, basándose en conocimientos y razonamientos, lleva a cabo tareas dificiles usualmente realizadas por expertos humanos [PCKW89]. Los SE no sólo proveen de capacidad de razonamiento al computador, la cual puede volver inteligente a una Rase de Datos, sino que deben ser vistos como métodos de alto nivel para programación de Bases de Datos.

La arquitectura de un SE tiene tres componentes: la Base de Conocimiento, el Soporte de Inferencia y la Interfase con el usuario. En la representación del conocimiento se utilizan usualmente Framas y Objetos. Los Objetos pueden representarse en jerarquias, haciendo posible la Herencia. Se llama Inferencia al proceso de combinar Hechos y Reglas para deducir nuevos Hechos. Las Reglas son presentadas como combinaciones de premisas y conclusiones y pueden ser aplicadas en los métodos de inferencia de encadenamiento hacia adelante y hacia atras.

Así pues, incorporar la Representación del Conocimiento y la Capacidad de Inferencia, herramientas provistas

por la tecnologia de los SE, puede extender los modelos y sistemas de las Rases de Datos Convencionales de manera que se transformen en BDI.

Pero, aplicar SE al desarrollo de BDI no requiere sólamente representar el conocimiento de una Rase de Datos de manera que pueda ser usado para la recuperación deductiva. Es necesario construir una interfase de software entre los Sistemas Manejadores de Bases de Datos y los SE.

Los SE no sólo proveen una forma de archivar experiencia y hacerla disponible cuando sea necesaria. También proveen construcciones de lenguaje de alto nivel. útiles para la programación de Bases de Datos. Estas construcciones incluyen reglas de encadenamiento hacia adelante y hacia atrás, representación del conocimiento y predicados relacionales. Es asi como surgen los Sistemas de Rases de Datos Expertas, capaces de combinar las mejores capacidades de los Sistemas de Rases de Datos y los Sistemas Expertos.

Existen dos componentes principales en la tarea de enlazar los SE y los Sistemas de Bases de Datos. La primera tarea es proveer un apropiado ambiente de programación. Esto se puede lograr:

A. Enfatizando el rol del Shell de un Sistema Experto, como un ambiente de programación de alto nivel y,

R. Enlazando las relaciones propias de un Sistema Experto, expresadas en términos de Hechos y Reglas, con las relaciones existentes en una Rase de Datos Relacional.

La segunda tarea envuelve el desarrollo de modelos comunes de datos. La Rase de Conocimiento de un SE puede ser usada para extender la semántica de un modelo de Rase de Datos. El secreto es reconocer que tanto las Bases de Datos como los SE se refieren básicamente a Objetos y sus interrelaciones. Asi pues, la integración de los SE y las Rases de Datos Relacionaes recae en el desarrollo de la representación de Objetos. Este es el esqueleto común que permite mapear la semántica de la Base de Datos a los metodos de representación propios de los Sistemas Expertos, y viceversa.

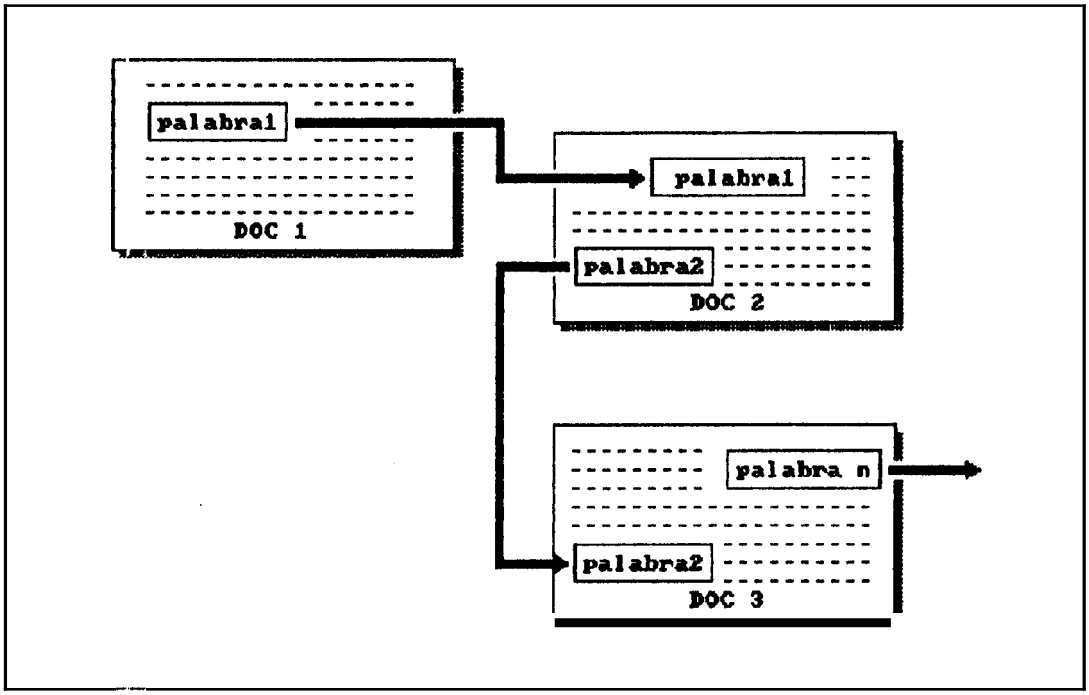
En conclusión, Hechos, Reglas, Objetos y Soporte de Inferencia son las herramientas básicas para la representación de Conocimiento e Inferencia en una BDI. Estos componentes, provistos por los SE, son también útiles en la construcción de modelos de Bases de Datos Convencionales,

Sistemas de Hypertext,, y Sistemas avanzados de Interpretación de Textos y Recuperación de Información.

1.3.2 HYPERTEXT Y HYPERMEDIA.-

Hypertext representa un paso hacia Bases de Datos más dinámicas: esto quiere decir: Bases de Datos en las cuales nuevos hechos o datos pueden ser registrados durante el proceso de computación. Pero también hay otro sentido de dinamism:.:. donde la información puede ser añadida en forma libre sin importar la estructura de la Rase de Datos. Asi pues, una de la metas de las BDI es combinar el dinamismo del Hypertext con la estructura y eficiencia de las Rases de Datos Relacionales.

Hypertext fue ideado como un instrumento **para** construir y usar estructuras de asociación [PCKW89]. Normalmente. la lectura de un documento es lineal y uno intenta leerlo desde el principio hasta el final. Hypertext, permite un final-abierto: el usuario puede saltar de una idea a otra dependiendo de sus intereses particulares. En vez de enlaces entre palabras existen **enlaces** entre Documentos y fragmentos de Textos. Desde **la** perspectiva de las BDI, Hypertext puede ser visto como un sistema de Base de Datos sin restricción de enlaces entre registros y archivos.



igura 4. Enlaces entre Documentos mediante las Palabras de un Hypertext.

La noción de Hypertext se extiende a Hypermedia. donde la información puede ser representada en medios alternativos como gráficos y sonido, además de textos. Redes de información enlazada **con** Hypermedia forman nuevos tipos de documentos conocidos como 'Documentos de Hypermedia'. Estos documentos sirven como Sistemas para representación de Conocimiento y enlace a Sistemas de Inferencia y Recuperación de Información.

El modelo de Hypermedia consiste en un conjunto básico de nodos y enlaces que permiten representaciones múltiples y entrelazadas del conocimiento. Este modelo fue suplementado con herramientas auxiliares y procesos que suministran las capacidades de hojear, representación de conocimiento, inferencia y recuperación de información, todas críticas para el desempeño de las BDI. El Sistema Hypermedia es más flexible que los tradicionales métodos de estructuración de información, pues permite que información, en varios formatos, sea anexada a los nodos. La variedad de nodos y enlaces yue pueden ser definidos en Hypertext contribuye a la flexibilidad de esta herramienta. Son justamente los tipos de enlaces los que definen la estructura de Hypermedia y suministran la capacidad de hojear y explorar los nodos.

Por la manera en que ha evolucionado y el énfasis que se le he dado a su naturaleza desestructurada, Hypermedia ha sido qeneralmente considerada como una manera informal de suministrar información que pueda ser "hojeada". La representación de conocimiento de Hypermedia es análoga en muchas maneras a una red semántica, donde lo importante es estructurar y definir los básicos tipos de nodos y enlaces.

La validez de Hypermedia aumenta por su flexibilidad para almacenar y recuperar conocimientos. Es posible realizar búsquedas y recuperar información, tal y como en las Bases de Datos y en los Sistemas Convencionales de Recuperación de Información. Un Sistema de Recuperación de Información basado en Hypermedia, puede filtrar información para satisfacer las necesidades del usuario. El enlace de los nodos de Hypermedia a las correspondientes representaciones jerárquicas suministra una poderosa forma de indexación. Así pues, **Hypermedia no reemplaza la Recuperación de Información Convencional, más bien la incorpora dentro del amplio modelo de acceso y utilización de la información.**

Hypermedia representa un valioso apoyo en la formulación de búsquedas. Los Documentos de Hypermedia pueden ser usados para crear un sofisticado 'diccionario' en línea, donde las palabras están enlazadas a sus definiciones, a sinónimos y a términos relacionados. De esta manera, Hypermedia asiste la formulación de búsquedas ya que permite que el vocabulario del usuario y del sistema converjan, ántes de realizar la busqueda.

En la evolución de la tecnología de las BDI, Hypermedia ha sido también usada como una interfase del usuario

hacia las poderosas facilidades de manejo de información disponibles en una Base de Datos.

1.3.3 ORIENTACION A OBJETO.-

Una capacidad fundamental de las BDI es que el Modelo Abstracto de Datos de la Base. provisto al usario. se acerque lo más posible al Modelo Real que él concibe. Esta caracteristica es alcanzada incorporando la Orientación a Objeto. al Modelo de Datos de una Base de Datos Inteligente.

Los conceptos fundamentales de la Orientación a Objeto son: Encapsulación, Herencia, y Objetos. Estos. combinados con el Soporte de Transacciones. resultan en poderosos Sistemas llamados Bases de Datos Orientadas a Objeto. Las Bases de Datos Orientadas a Objeto son consideradas una nueva filosofía en el almacenamiento de información y datos. Bien diseñadas. pueden estar integradas a las Bases de Datos Relacionales, y constituir un suplemento a ellas.

Algunas aplicaciones de la Orientación a Objeto en el campo de la representación del conocimiento son: simulación de eventos discretos, ingenieria de diseño e ingenieria de software.

El Capitulo IV de la presente Tesis, profundiza el Tema de la Programación Orientada a Objeto.

1.3.4 MANEJO DE TEXTOS Y RECUPERACION DE INFORMACION.-

Las Bases de Datos Textuales en Linea son el último reto para las técnicas de BDI. La mayor dificultad a la que se enfrentan es el proceso de indexación requerido. Uno de los secretos para el desarrollo de inteligencia en una Base de Datos es el uso de un proceso de indexación que se base en Conceptos más que en Palabras Claves.

El Capítulo II de la presente Tesis, profundiza el Tema del Manejo de Textos y la Recuperación de Información.

1.4 ARQUITECTURA DE UNA BASE DE DATOS INTELIGENTE.-

Los tres principales componentes de una BDI son:

- Inteligencia en las Herramientos de Alto Nivel.
- Inteligencia en las Interfases con el Usuario.
- Inteligencia en el Soporte de la Base de Datos.

Haciendo una analogia entre los Seres Humanos y las Bases de Datos [PCKW89], se puede decir que los primeros utilizan: sus sentidos para obtener información, sus músculos para realizar actividades y sus mentes para grabar

(memorizar) o **recuperar** (recordar) información, razonar y tomar decisiones. Así pues, podemos distinguir entre: los sentidos y los músculos, que actúan como las interfases de las personas con el mundo externo y la mente que internamente procesa información y planea acciones. Esta distinción entre sentidos y músculos por un lado y **la** mente por el otro, es análoga al nivel de **Interfases** con el Usuario y el nivel de Soporte de la Base de Datos (Figura 5).

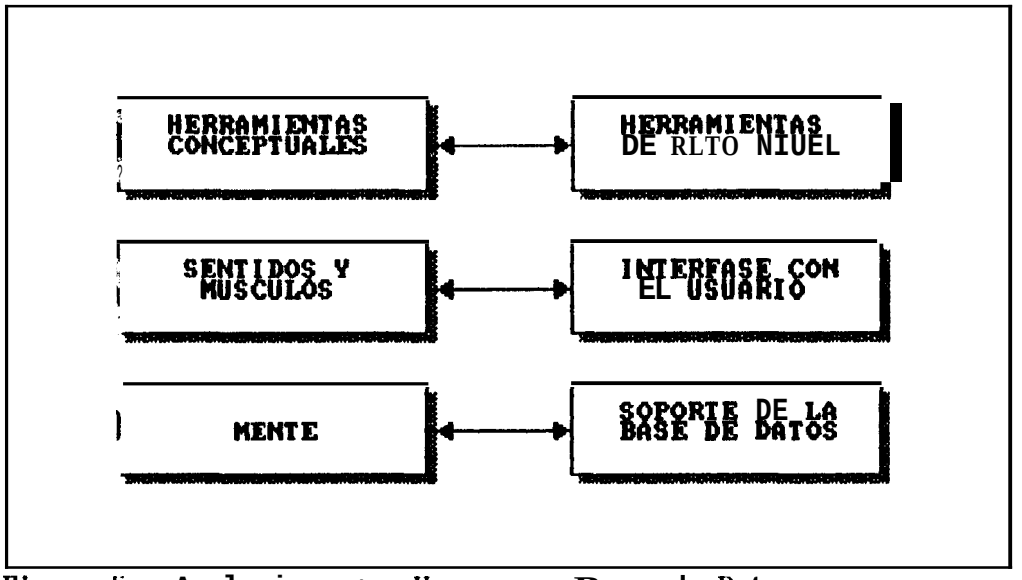


Figura 5. Analogia entre Humanos y Bases de Datos.

En esta arquitectura con niveles escalados, los usuarios y desarrolladores pueden accesar diferentes capas en diferentes momentos (Figura 6).

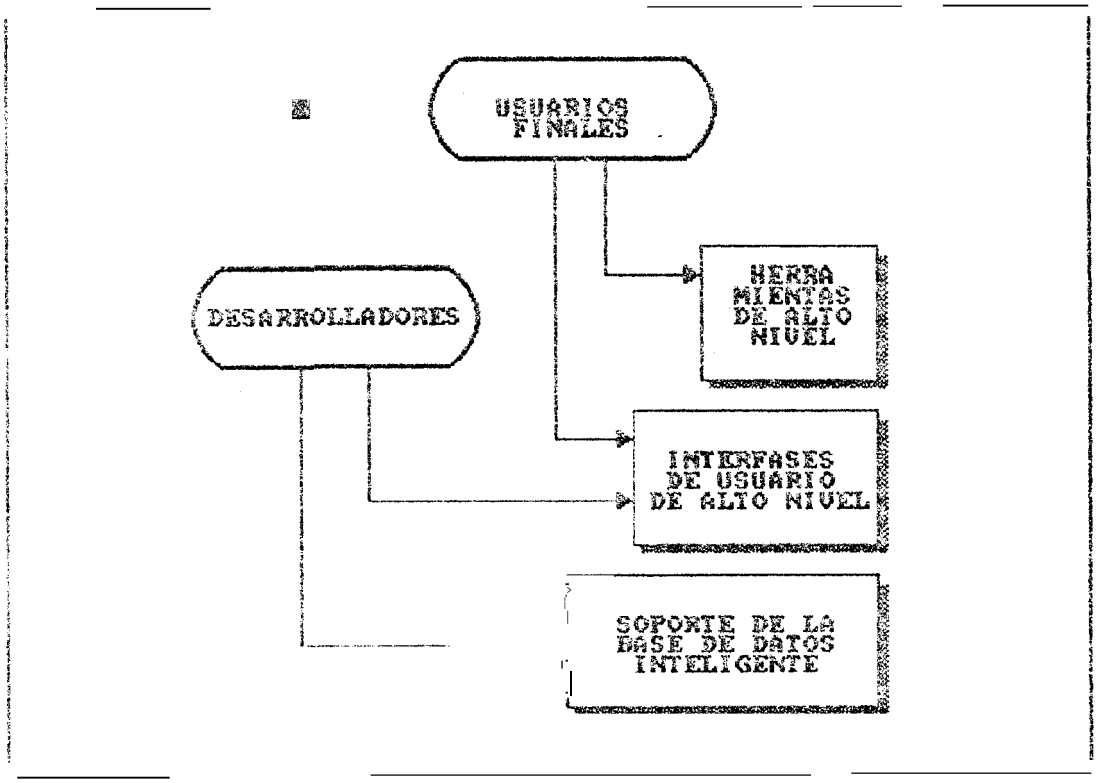


Figura 6. Los tres niveles de inteligencia de una Base de Datos.

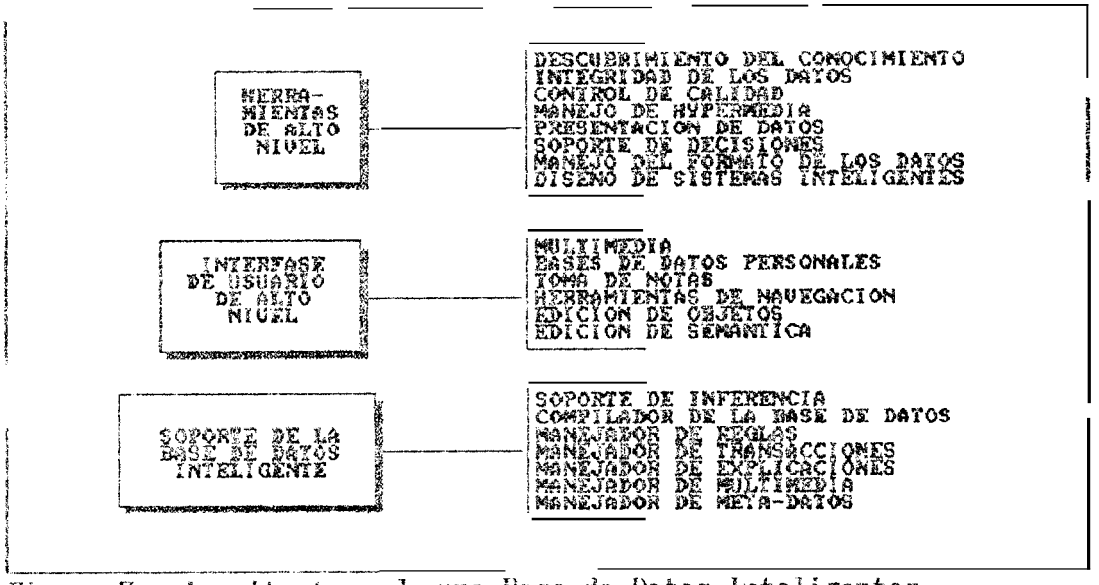


Figura 7. Arquitectura de una Base de Datos Inteligentes

1.4.1 INTELIGENCIA EN EL NIVEL DE LAS HERRAMIENTAS DE ALTO NIVEL.-

Además de la integración de las tecnologías tratadas previamente, existen herramientas de alto nivel que suplementan y complementan la funcionalidad de una BDI. Estas herramientas pueden ser utilizadas por usuarios y desarrolladores de Bases de Datos y representan una librería externa de gran poder. Son orientadas a objeto y su estructura básica refleja los métodos de representación de objetos del modelo inteligente de la Base de Datos.

Existen 7 Herramientas de Alto Nivel [PCKW89]:

A. **Herramientas de Descubrimiento de Conocimientos:** Incluyen análisis de datos, aprendizaje de la máquina y análisis estadístico. Permiten al usuario descubrir dentro de una gran Base de Datos, relaciones y enlaces escondidos e inesperados.

B. **Herramientas para Control de la Integridad y Calidad de Los Datos:** Son necesarias debido a los indeseables efectos del incremento del número y tamaño de las Bases de Datos. Tomando en cuenta la creciente dependencia del ser humano a los datos grabados en una Base, es imperativo contar con herramientas que auto-

máticamente detecten o señalen errores en ellas. Estas herramientas van desde aquellas que usan el conocimiento de un experto en la detección de errores, hasta otras que señalan estadísticamente desviaciones en los registros.

C. **Herramientas para Manejo de Hypermedia:** Permiten a los desarrolladores y usuarios construir Sistemas de Información Hypermedia, combinando textos y enlazándolos libremente con datos, imágenes, sonido. etc. Esta categoría refleja el hecho de que la información puede ser expresada en muchas maneras a medias diferentes y que son necesarios métodos para organizar y accesar estas diferentas formas de información.

D. **Herramientas para Presentación de Datoo:** Estas herramientas proveen graficos, y otros tipos y formas de presentación de datos y sumarios. En la práctica constituyen una extensión de las herramientas de manejo de Hypermedia.

E. **Herramientas para Soporte de Decisiones y Análisis del Entorno:** Permiten una fusión entre hojas electrónicas. Bases de Datos, Sistemas de Finanzas, etc. Par-

ten de la visión del Soporte de Decisiones como una clase especial de Recuperación y Utilización de la información, donde esta es valorada dentro de las particulares actividades de toma de decisiones.

F. **Herramientas para Formato de Datos:** Estas herramientas permiten a los usuarios transformar datos de un fornato a otro. Ej: fusionar un archivo ASCII de datos en formato DBASE. con datos obtenidos en una búsqueda SQL. Son indispensables para aplicaciones que se relacionan con análisis de datos reales.

G. **Herramientas para Diseño de Sistemas Inteligentes:** Permiten la integración del diseño de Bases de Datos, diseño de Sistemas de Información y diseño de Sistemas Expertos. Mediante estas herramientas los desarrolladores y administradores de sistemas obtienen facilidades para diseñar y mantener BDI.

Aunque todas estas herramientas de alto nivel parecen ser periféricas a la idea central y al funcionamiento de una Base de Datos, algunas son indispensables para que las BDI trabajen en ciertas aplicaciones.

1.4.2 INTELIGENCIA EN EL NIVEL DE LAS INTERFASES CON EL USUARIO.-

Este es el nivel con el que los usuarios interactúan directamente. Trata tanto de cómo el usuario piensa acerca de las Bases de Datos y el Manejo de Información, así como de la manera en que el Soporte de la Base de Datos efectivamente opera.

Las interfases del usuario constan de dos aspectos:

- Existe un modelo esencial presentado al usuario. Este consiste de una representación (orientada a objeto) de la información y de un conjunto de herramientas básicas que permiten al usuario crear nuevos tipos de objetos, realizar búsquedas y consultas, etc.
- Adicionalmente, existe un conjunto de herramientas de alto nivel que aumentan la funcionalidad del Sistema de BDI para ciertos tipos de usuarios.

Se distinguen dos niveles de interfase con el usuario:

- Nivel Físico: Consiste en los dispositivos de entrada y salida como el mouse, el teclado y el monitor.
- Nivel Cognocitivo: Consiste en el modelo utilizado para presentar la información, la interpretación que

el usuario hace de ella y las intenciones que en él se formulan.

Las primeras interfases parecían confundir ser "inteligentes" con ser "esotéricas". Uno no sólo tenia que ser inteligente para utilizarlas, sino también aprender una sintaxis compleja y arbitraria. Las interfases eran prácticamente mudas y la inteligencia residia totalmente en la capacidad del usuario. Algunos usuarios se adaptaron a esta situación, otros prefirieron simplemente desechar el computador.

Una interfase debe ser transparente, fácil y natural de usar. El usuario tipico debe ser capaz de sentarse frente al computador a ejecutar una tarea y en cada paso del programa debe serle obvio qué hacer a continuación. La naturalidad debe ser vista como un aspecto de inteligencia en las interfases. Ellas utilizan razonamientos para interpretar lo que el usuario realmente quiere o para deducir información que no puede ser obtenida directamente.

Algunas estrategias para incrementar la naturalidad de una interfase son:

- Usar, además de texto, mediov como sonidos y gráficos.

Permitir a los usurios pasar de vistas cercanas a lejanas de tópicos (Zoom).

Permitir a los usuarios llevar a cabo búsquedas asociativas: pasar de un tópico a otros relacionados.

- Donde sea posible, diseñar la interfase de manera que contenga objetos y conceptos familiares.
- No dar información bruta. Las personas quieren respuestas, resúmenes, sumarios de sus necesidades.
- Permitir a los usuarios operar directamente mediante objetos representados por la interfase, más que a través de algún tipo de lenguaje de comando.

1.4.3 INTELIGENCIA EN EL NIVBL DE SOPORTE DE LA BASE DE DATOS.-

El nivel básico del sistema es el Soporte Inteligente de la Base de Datos. Este consiste en un conjunto básico de capacidades que permiten a un Sistema Manejador de Base de Datos o a una Aplicacion de Manejo de Información, funcionar como lo hace. Muchas de las actividades de Soporte dependen de las especificaciones de Hardware y Software del ambiente en el que la BDI es implementada.

Toda la inteligencia de un Sistema de Bases de Datos puede ser desarrollada en terminos de ciertas características [PCKW89]:

- **Modclo de Datos Orientado a Objeto y basado en Conocimiento:** Este tipo de Modelo de Datos permite la representación de informacidn en una forma que refleje la percepción del usuario del mundo real. Una vez formulado, puede servir como la Base de Conocimiento de un Sistema Experto.

- **Soporte de Inferencia en las Bases de Datos:** Permite la recuperación deductiva, pues la búsqueda de informacidn y la inferencia están integradas.

- **Búsquedas sensibles al contexto o estructuradas sensitivamente:** La búsqueda sensible al contexto encuentra informacidn relevante basandose en "contenido". Por otro lado, la búsqueda de estructrura sensitiva recupera información en base a "forma" (similitud de sonido o de escritura).

- **Soporte de múltiples dispositivos de almacenamiento:** Permite grabar informacidn de varios tipos (mapas,

pinturas, gráficos, sonidos, etc) y recuperarla por medio de Bases de Datos.

- **Manejo inteligente de versiones, recuperación y elasticidad:** El Manejo inteligente de versiories asegura que versiones corrientes y previas de la Base de Datos puedan ser recuperadas eficientemente. Recuperación y elasticidad son elementov que permiten a la Base de Datos manejar las demandas operacionales que expe[illegible] menta, incluyendo acceso simultáneo de varios usuarios y fallas de equipos.

- **Optimización de Búsquedas:** Provee respuestas adecuadas a complejas búsquedas que envuelven Rases de Conocimiento Orientadas a Ohjeto.

- **Soporte transacciones concurrentes.**

CAPITULO II

MANEJO DE TEXTOS Y RECUPERACION DE INFORMACION

2.1 RECUPERACION DE INFORMACION.-

La Recuperación de Información concierne la representación, almacenamiento, organización y acceso de items de información. El advenirniento de las computadoras y el desarrollo de económicos y eficientes dispositivos de almacenamiento de datos, han llevado a un marcado desbalance en la Ciencia de la Información. Las actuales capacidades de almacenamiento de información exceden grandemente la cantidad de información que puede ser organizada y recuperada eficientemente. Son neceaarios metodos para representar, organizar y accesar información en forma compatible con las capacidades de almacenamiento de información.

Trataremos los métodos de Recuperación de Información, desarrollados para tratar con grandes Bases de Datos Textuales con millones de documentos y extractos de ellos. Estos Sistemas de Recuperación en Línea han sido siempre considerados del dominio de la Ciencia de la Información y no se ha dado mucha integración entre las Bases de Datos Convencionales y ellos. Esta separación se debe, en parte,

a los diferentes tipos de datos que manejan las Bases de Datos y los Sistemas de Recuperación en Linea: Registros vs Documentos, respectivamente.

2.1.1 EL DESARROLLO DE LA TECNOLOGIA DE LA INFORMACION.-

Los conceptos claves para la representación y recuperación do información fueron desarrollados en el ambiente del papel.. A pesar de que las paredes de cuevas. tablas de cera y pieles de animales se podian usar para representar información, fueron el papel y la imprenta los que convirtieron a la información en un producto masificado.

Junto con la producción masiva de informacidn. vino la necesidad de organizarla. Se desarrollaron sistemas de Cátalogos para referenciar documentos indirectamente: el Catálogo por si sólo se convirtid en una fuente de información. A su vez, tecnologias como la Microficha simplificaron el almacenamiento de mayores volúmenes de información. Esta podia ser almacenada en una forma muy compacta y reproducida en papel. cuando fuera necesario. La Microficha ofreció almacenamiento conveniente, pero no permitía procesamiento directo de la información.

El impacto de los medios electrónicos en la Recuperación de Información es incalculable. Frimeramente presentaron nuevas maneras de representar y organizar informacidn. En segundo lugar, permitieron procesar la información, incluyendo el manejo de Bases de Datos y Lenguajes de Búsqueda. En tercer lugar. una vez solucionados los problemas de programación simbólica y representación de con(:)-cimiento, desarrollaron la habilidad del raconamiento electrónico. Sin embargo, hasta tiempos recientes, las tecnologías de la Recuperación de Información y el Razonamiento Electrónico han estado disociadas. Existen pocos casos donde Sistemas Expertos pueden utilizar directamente información de grandes Bases de Datos. Mas Lien, las Bases de Datos de Conocimientos fueron especialmente diseñadas para aplicaciones especificas de Sistemas Expertos.

2.1.2 CARACTERISTICAS DE LA RECUPERACION DE INFORMACION.-

El modelo basico de la Recuperación de Información es muy simple: la información es almacenada, y posteriormente, cuando es necesaria, recuperada. Pero, cómo precisamente debe ser definida y almacenada la información? Es necesario un lenguaje que describa la información y la etiquete para su posterior recuperación.

Los Sistemas de Recuperación en Línea existentes consisten tipicamente de una gran Ease de Datos con resúmenes de documentos. Palabras y frases que describen el contenido de los documentos (términos de indexación) son seleccionados, ya sea de un vocabulario controlado de terminos estándard o libremente del titulo y resúmen del documento. La información necesitada por el usuario es representada con un requerimiento de búsqueda, parafraseado en la forma de: "Que documentos se tienen que cumplan las siguientcs condiciones?". Las condiciones son ingresadas como una serie de palabras claves, combinadas con un lenguaje de búsqueda basado comunmente en Lógica Rooleana. El uso apropiado de los operadores lógicos permite realizar entre los documentos, operaciones de intersección, unión, exclusión, etc, y recuperar aquellos cuyos términos indexados cumplen la combinación de condiciones.

Se puede resumir el paradigma de la Recuperacidn de Información como un proceso de *coincidencia*, donde la relevancia de los documentos se determina por la *similitud* entre sus términos indexados y la busqueda planteada.

La manera en que la información es almacenada e indexada, es determinada por los métodos de recuperación que

vayan a ser usados. En un Sistema de Bases de Datos Convencionales, la factura de un cliente necesita un registro, cuyos campos de nombre del cliente y número de factura estén indexados. Esto es suficiente para almacenar este tipo de datos. Sin embargo, el problema de los Sistemas de Manejo de Documentos no es directamente análogo a los Sistemas de Recuperación de Datos, pues los primeros tratan con textos mas que con registros. Asi pues. indexadores de un artículo de 50 páginas deben encontrar la manera de representar la información e ideas contenidas en el documento. Esta puede ser una tarea dificil y controversial.

Se ha demostrado que ni siquiera un par de indexadores coinciden en el conjunto de términos de indexación para un documento. Muchas veces, los indexadores no están de acuerdo en el significado del texto, peor aún en la manera en que los terminos de indexación deben ser asignados para representar un particular concepto contenido en él. Aún en casos de Compañías de Rases de Datos. donde se usa un vocabulario estricto y controlado, siguen habiendo una cantidad sorpresiva de desacuerdos entre los indexadores. Todo esto lleva a la inseguridad de cómo expresar un requerimiento de busqueda, para tratar de encontrar información en una Base de Datos indexada inconsistentemente. . Vemos

asi que la Recuperación de Información en Bases de Datos Textuales es un proceso mucho mas relativo que la Recuperación de Datos en otros tipos de Rases.

2.1.3 DBFINICION DE RECUPERACION DE INFORMACION.-

La palabra ´Información´ puede ser definida mediante dos aspestos [PCKW89]:

A. La Información es un recurso necesario para llevar a cabo ciertas tareas. Consiste generalmente en datos o textos que expresan hechos y reglas. Tambien puede ser expresada mediante gráficos y sonidos.

B. La cantidad de información que se recibe en una comunicacion depende de las personas involucradas y corresponde a la diferencia entre lo que ellas saben ántes y después de escuchar la información.

Puesto que la información es un recursu vital para tomar decisiones y aumentar la productividad. uno de los criterios para el diseño de una BDI es que la información entregada a los usuarios sea relevante. De la habilidad para definir esta utilidad o relevancia de la información depende que el usuario reciba sólo lo que necesita saber y no información general e inutil.

Se requieren Búsquedas para procesar la inforrnación en una cierta forma. La tarea de procesar información en Bases de Datos Textuales es generalmente llamada ***Recuperación de Xnformacidn,*** y los Sistemas que llevan a cabo esta tarea son llamados ***Sistemas de Recuperación de Información o Sistemas de Recuperación en Línea.***

El término 'Recuperación de Información' fue popularizado en 1950 y puede ser definido como la búsqueda y recuperación de información, desde un almacenamiento, de acuerdo a un tema especificado [PCKW89]. La Recuperación de Información increment6 las capacidades no sólo de las Bibliotecas, sino de todos los lugares donde la información pueda ser catalogada e indexada. Lop especialistas en Recuperación de Información entienden su rol, como aquel de informar a los usuarios dónde está la información. Un Sistema de Recuperación de Información no aumenta el conocimiento de un usario sobre un tema en especial, simplemente le indica existencia o inexistencia y qué documentos contienen la información deseada.

Así pues, la meta de la recuperación de información no es proveer información directamente, sino indicar dónde la información relevante puede ser encontrada. En una Base de

Datos Relacional, esto es análogo a decirle a una persona dónde encontrar un registro, sin decirle qué está dentro de él. Existen modelos alternativos de recuperación de información basada en conocimiento. que buscan proveer información directamente, más que citas o nombres de documentos donde se espera encontrar la información.

Asociados al modelo tradicional de Recuperación de Información (Figuras 8 y 9) existen dos tipos de personas. Por un lado están los usuarios. con sus requerimientos o pedidos de información. Por el otro lado están: los indexadores, que almacenan la información, y las personas que traduciendo los requerimientos del usuario, realizan efectivamente las búsquedas y recuperan la información. La existencia de usuarios, indexadores y buscadores crea una serie de problemas potenciales de comunicación. pues puede existe incompatibilidad entre los lenguajes y la terminologia utilizadas por cada grupo.

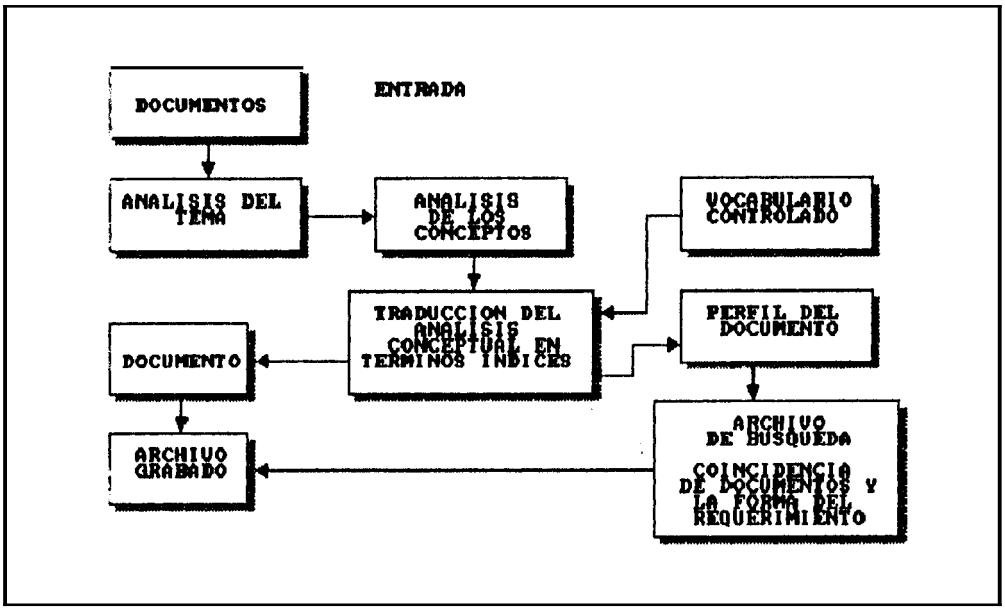


Figura 8. Entrada: Modelo Tradicional de Recuperación de Información.

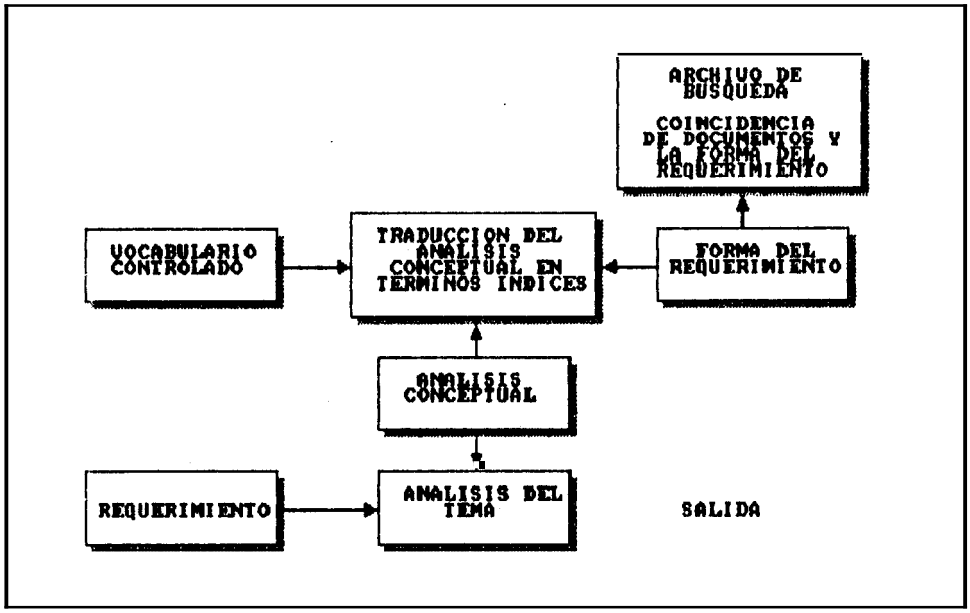


Figura 9. Salida: Modelo Tradional de Recuperación de Información.

2.1.4 NDEXACION MANUAL Y AUTOMATIZADA.-

La recuperación de los documentos es iniciada por un requerirniento de busqueda que especifica las propiedades que los items deseados deben tener. Idealmente, las propiedades especificadas para la busqueda. coinciden con la manera en que los documentos han sido etiquetados (indexados) en la Rase de Datos. Ya que el objetivo de un proceso de Recuperación de Información es formular una búsqueda que permita obtener documentos relevantes. el éxito de este proceso depende de la manera en que los documentos hayan sido indexados y la manera en que su relevancia pueda ser asignada en base al requerimiento de búsqueda (Figura 10).

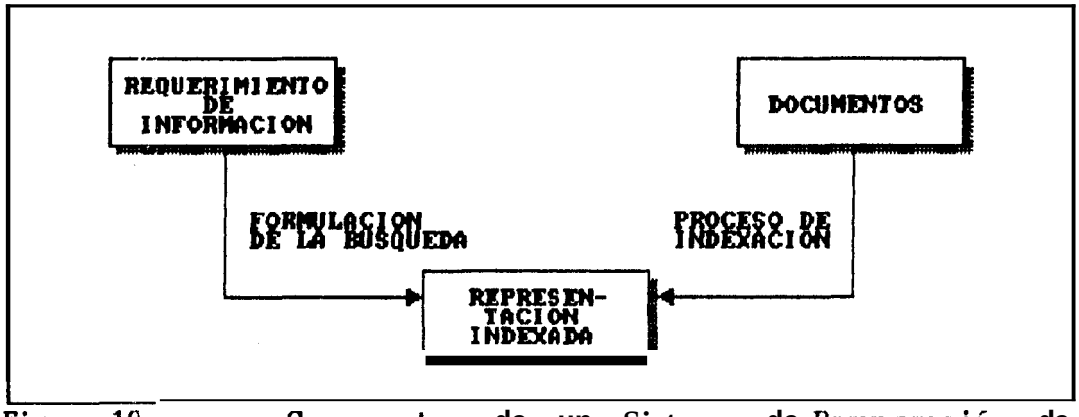


Figura 10. Componentes de un Sistema de Recuperación de Información.

El proceso de indexación crea un Archivo de Indices. tambien llamado Archivo Invertido. Cuando se realiza una

búsqueda usando un archivo invertido, el indice es examinado para determinar qué items satisfacen el requerimiento. Esto es mucho mas rapido que buscar por medio de los documentos originales. Los archivos directos e invertidos son sólo dos maneras diferentes de presentar la relación entre los documentos y sus términos indexados. En el caso de los archivos directos, estos proveen los terminos de indexación asociados a cada documento (Figura 11). Al contrario. los archivos invertidos identifican, para cada término. todos los documentos indexados por medio de él (Figura 12).

En el modelo tradicional, la indexación es vista como un proceso de clasificación donde los documentos son etiquetados de acuerdo al tema que tratan. Este tipo de índice es similar al que se encuentra en la parte posterior de los libros, excepto en el hecho de que es construído con un vocabulario controlado de términos, escogidos especialmente para cubrir el espacio de posibles conceptos. sin incluir términos superpuestos ni sinónimos. El vocabulario es "controlado" en la medida en que nuevos términos pueden ser añadidos sólo si representan nuevos conceptos.

El proceso de indexación de documentos puede ser llevado a cabo manual o automaticamente.

ARCHIVO DIRECTO

	TERM 1	TERM 2	TERM 3	TERM 4
DOC 1	1	0	0	1
DOC 2	0	1	1	0
DOC 3	1	1	0	0

Figura 11. Archivo Directo.

ARCHIVO INVERTIDO

	DOC 1	DOC 2	DOC 3
TERM 1	1	0	1
TERM 2	0	1	1
TERM 3	0	1	0
TERM 4	1	0	0

Figura 12. Archivo Invertido,

A. INDEXACION MANUAL.-

La indexación manual es generalmente vista como una tarea difícil. Como resultado de esto, las Compañías de Bases de Datos aplican frecuentemente una serie de reglas para asistir a los indexadores y forzar la consistencia. Ej: La "Regla de Tres o Más" para indexar componentes químicos: si un articulo menciona tres o mas miembros de la misme familia química, el nombre de la familia es usado en el índice, y no los nombres de las sustancias. Asi pues, si, un articulo nombra al mercurio, cadmio y plomo, "metales pesados" sera usado como término de indexación.

En general, existen cuatro componentes de la tarea de indexación manual [PCKW89]:

1. Escoger una Lista de Autoridad con términos de conceptos relevantes.
2. Utilizar reglas para construir términos de tópicos complejos.
3. Examinar un documento para determinar qué conceptos y temas son referidos en él.
4. Utilizar una variedad de reglas para controlar el uso de la Lista de Autoridad en la asignación de los términos de indexación. Estas reglas son usadas general-

mente para reducir la ambiguidad y forzar la consistencia entre indexadores.

Como vemos. en el modelo de. indexacion manual los indexadores no sólo deben leer y comprender muchos documentos. sino tambien recordar el vocabulario controlado y las reglas de asignación de terminos. Entre los métodos que asisten la indexacion manual se incluyen diccionarios jerarquitos y sistemas de referencia cruzada.

Si bien la indexación manual permite detallar los términos de indexación que reflejen realmente el contenido de los documentos, en la practica. el proceso es difícil y los resultados pueden ser decepcionantes.

B. INDEXACION AUTOMATIZADA.-

La indexación automatizada esta comúnmente baaada en un análisis estadistico de las palabras y frases usadas en el texto de los documentos. Un método es el siguiente [PCKW89]

1. Calcular la frecuencia de cada término unico. para todns los documentos de la Rase. La frecuencia **F** de un término **k** en un documento **i** es denotada como: F_{ik}.

2. Determinar la frecuencia total **FT** de cada palabra de la Base de Datos: FT_{ik}.

3. Ordenar las palabras en orden descendente de frecuencia en la Base de Datoe. Eliminar las palabras de mucha ocurrencia.

4. Remover las palabras de poca ocurrencia. Esto remueve términos que ocurren tan infrecuentemente que su ausencia del índice no afecta la calidad de la recuperación de información.

5. Utilizar las palabras de frecuencia mediana como términos de indexación de la Rase de Datos.

Este método basico de indexación automatizada se extiende de muchas maneras. Algunos métodos envuelven "limpiar" los conjuntos de términos, eliminando plurales y palabras que tengan la misma raíz y difieran sólo en el sufijo. Métodos más complejos examinan la coocurrencia estadística de palabras y remueven aquellas que aparentan ser sinónimos o conceptos superpuestos. Por otro lado, los metodos automatizados de indexación pueden ser combinados con los métodos manuales y así constituir métodos interactivos de indexación.

2.1.5 INTERMEDIARIO DE BUSQUEDAS.-

Usualmente, se requiere experiencia especializada para poder entender las políticas de indexación y formular búsquedas en el lenguaje de la Rase de Datos. Como la mayoría de los usuarios no tienen evta experiencia. ellos necesitan una especie de intermediario que realice las tareas de búsqueda. En la vida diaria, los bibliotecarios son intermediarios tipicos, pues organizan las fuentes de información y asisten a los usuarios en sus necesidades.

La interacción entre el cliente. con un requerimiento de búsqueda, y el intermediario, que realiza efectivamente la búsqueda en la Base de Datos, no es tan fácil. En la situación ideal. el cliente es un experto en su campo y tiene un conocimiento práctico de los recursos de información. Por su parte, el intermediario debe tener por lo menos un conocimiento práctico del campo general de estudio del cliente, aunque ésto se dificulta si el tópico a tratar es técnico o especializado. De la interacción entre el cliente v el intermediario nace el proceso de ***Negociación de la Búsqueda.*** Este, representa un puente entre el entendimiento de cada uno, sobre el tópico específico y busca transformar el requerimiento de información en comandos de búsqueda reconocibles por la Base de Datos.

El intermediario representa una tercera entidad inteligente entre el usuario y el Sistema de Recuperación en Línea. Si reconocemos dos canales de comunicación durante la ejecución de una búsqueda, podemos decir que el primer canal es usado por el usuario y el intermediario en la formulación del requerimiento de búsqueda, y el segundo por el intermediario en su interacción con la Base de Datos.

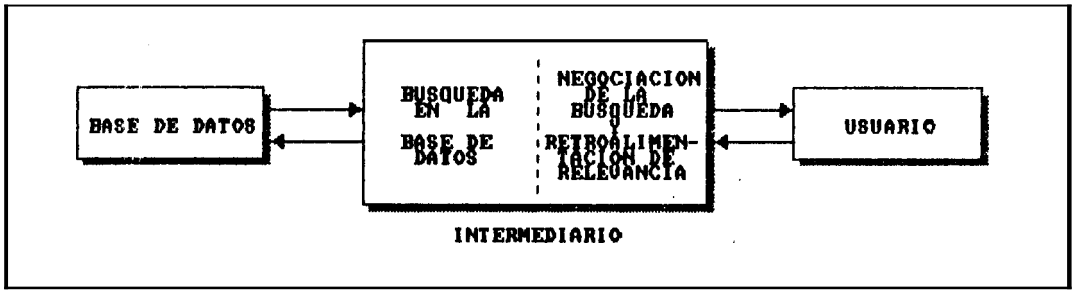


Figura 12. El rol del Intermediario en las búsquedas en línea.

Como ya vimos, las demandas de la Recuperación de Información requieren que usuarios e intermediarios conozcan, tanto de los procesoe de recuperación de información como de los contenidos de la información misma. Algunas clases necesarias de conocimiento son [PCKW89]:

A. Un reinimo de Experiencia Procedural en el uso de Sistemas de Computación.

B. Terminología de los diversos tópicos o dominios de conocimiento, sus conceptos y la relación entre ellos.

C. Estrategias para la formulación de búsquedas, tanto generales o específicas H una Rase de Datos. Ej: facilidad para combinar operadores lógicos con palabras claves.

D. Conocimientos sobre la construcción específica de una Base de Datos y sus politicas de indexación: tienen especial implicación en la decisión de qué estrategia de búsqueda es más promisoria para una Rase de Datos particular.

A pesar de que un intermediario de búsquedas es una parte esencial del paradigma de la Recuperación de Información, existen una serie de problemas asociados con el trabajo a través de el:

A. El rol del intermediario es una labor costosa. A medida que las tecnologias de Recuperación de Información maduran, el costo de los servicios de los intermediarios constituye una proporción cada vez mayor del costo total del proceso.

R. Muchos de los que conducen búsquedas bibliográficas accesan los Sistemas de Recuperación de Información por medio de terminales remotos. Asi pues, no tienen acceso real a intermediarios calificados.

C. Los intermediarios, por ser humanos, no pueden tener un conocimiento completo sobre todos los tópicos contenidos en las Bases de Datos. For la simple inseguridad y ambiguidad en el uso del lenguaje, matices críticos de los tópicos de los usuarios finales pueden perderse o distorsionarse durante la interacción con el intermediario.

Es claro que los usuarios preferirían interactuar directamente con los Sistemas de Recuperación en Línea, pues esto les da acceso directo y control. Asi pues. la necesidad de intermediarios es indicativa de una inapropiada interfase de software para la Recuperación en Línea.

El método predominante para la realización de búsquedas implica indentificar conceptos por medio de comandos booleanos. Sin embargo, esta comprobado que la formación y comprensión de expresiones lógicas excede las limitaciones de procesamiento de información del hombre promedio.

Pueden existir otras dificultades para el usuario. Este debe, en ciertos casos, anticipar los varios sinónimos y abreviaciones que pueden ser usadas para representar una palabra y preveer que una cadena especifica tenga mas de un significodo. Además de ésto, en algunas Bases de Datos los documentos están indexados sólo en el nivel más específico y asi, usuarios interesados en un tópico amplio como 'LENGUAJES DE PROGRAMACION' deben ingresar este término junto con todos los términos que representan un específico lenguaje. En caso contrario, sólo serán recuperados documentos específicamente indexados bajo 'LENGUAJES DE PROGRAMACION' y no aquellos indexados bajo 'PASCAL'.

Una efectiva búsqueda en línea debe, ya sea proveer asistencia al usuario en la construcción de los requerimientos de busqueda o, simplemente hacer el proceso totalmente transparente a él. Como hemos visto, existen razones de sobra para desarrollar Sistemas de Recuperación de Información basados en conocimiento que no requieran la traslación de conceptos a lógica Booleana.

2.1.7 FORMULACION DE BUSQUEDAS.-

Como ya vimos, la Recuperación de Información es un proceso en el que las necesidades de información del usua-

rio son comparadas con la información disponible en la Base de Datoe. Esto nos llevó al ***Paradigma de la Coincidencia.*** donde la información necesitada es expresada como una busquedn y la información potencial, los documentos. se representa como una coleccidn de términos indexados. Aquellos documentos cuyas representaciones indexadas estén más cercanas a las características de la búsqueda, son asumidos como relevantes y presentados al usuario.

El proceso de coincidencia puede ser determinístico o basado en alguna medida de distancia o similitud [PCKW89]. En la Coincidencia Determinística, se construye una cadena de búsqueda Booleana y los documentos son recuperados sólo si cumplen precisamente las condiciones expresadas en ella.

A su vez, es posible desarrollar representaciones del requerimiento de búsqueda y del espacio de los documentos que permitan calcular medidas de similitud entre la búsqueda y los documentos. De esta manera se podrían ordenar los documentos de acuerdo a su relevancia frente a la búsqueda. Sin embargo, la difundida adopción de "Coincidencia basada en Similitud" se obstaculiza por la serie de desacuerdos sobre cómo las medidas de similitud deben construirse y cómo las grandes Bases de Datos deben ser indexadas.

El **segundo** Paradigma de la Formulación y Evaluación de Búsquedas envuelve la ***Exploración*** más que la coindicencia. El usuario explora la información directamente y escoge lo que le parece apropiado. Este **Paradigma** se ha vuelto más popular ha medida que proliferan los Sistemas Hyper-text.

Ideslmente, los Paradigmas de Exploración y Coincidencia, dentro de la Recuperación de Información. pueden combinarse en las **RDI** y así permitir tanto el hojeo como la utilización de estructuras de búsqueda.

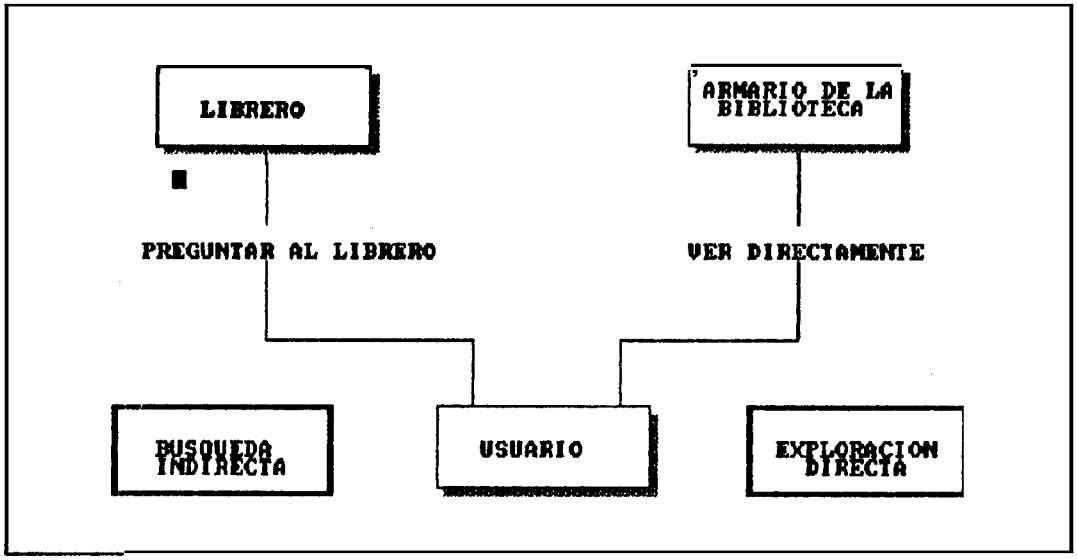


Figura 14. Dos Paradigmas para la Recuperación de Información.

Existe un tercer Paradigma de la Recuperación de Información, probado únicamente ***con*** Rases de Datos peque-

ñas. Se conoce generalmente como la ***Recuperación Conceptual.*** En él, el usuario expresa su necesidad de información a manera de un concepto. mas que como una colección de palabras **claves** conectadas por operadores lógicos. La Recuperación Conceptual es un tipo **sofisticado** de Paradigma de Coindicencia. **Ejs:** "Deseo documentos sobre la Contaminación del **Estero** Salado". "Deseo artículos sobre el encuentro en Guayaquil de Bólivar y San Martín". En la Recuperaci6n Conceptua1. e1 sistema recupera documentos en base a inferencias sobre los conceptos que éstos contienen. aún si esos **conceptos** no están explícitamente indexados en la forma del requerimiento original de búsqueda.

2.1.7 RETROALIMENTACION DE RELEVANCIA.-

La Recuperación de Información fue concebida como un proceso interactivo donde el usuario establece una búsqueda, reviza los resultados obtenidos y construye luego una nueva búsqueda. Como el proceso de construir búsquedas es dificultoso, tiene sentido considerar la manera en que una búsqueda anterior puede ser refinada. en base al juicio de relevancia de los documentos obtenidos de ella. Asi pues, la Retroalimentación de Relevancia es un proceso iterativo de búsqueda, donde los requerimientos son refinados en base a la retroalimentación desde la **Base** de Datos [PCKW89].

Una forma simple de Retroalimentación de **Relevancia** es examinar los documentos obtenidos de una búsqueda. escoger **en** base a sus títulos o resumenes, aquellos **que** eon relevantes y añadir al requerimiento de búsqueda los términos de indexación **asociados** a ellos. Al mismo tiempo se remueven del requerimiento los términos que no aparecen en los documentos relevantes. **Este** método informal da una gran carga al usuario o intermediario. pues ellos valoran manualmente **la** efectividad de los términos de indexación.

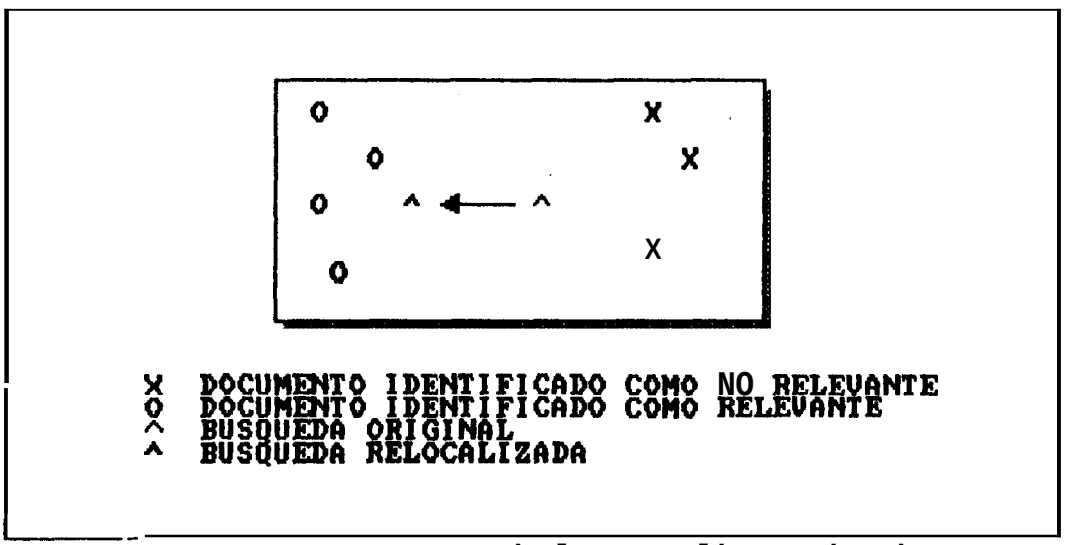


Figura 15. **Ilustración de la Retroalimentacion de Relevancia para relocalizar una Búsqueda en el eapacio de los Documentos.**

Un método mas complejo, pero mas facil de usar. es desarrollar un Sistema automatizado de Retroalimentación de **Relevancia** como el del Sistema SMART (citado en Salton y

McGill, 1983) [PCKW89]. En él, la **relevancia** de cada término de **una** búsqueda es determinada por un peso. Este peso es ajustado según el juicio de relevancia hecho sobre los documentos recuperados de la búsqueda y mediante el uso de la Ecuación 1.

$$Q' = \alpha Q + \beta\left(\left(1/R'\right)\cdot\sum DOC_i\right) - \gamma\left(\left(1/N'\right)\cdot\sum DOC_k\right)$$

Ecuacidn 1. Retroalimentación de Relevancia

	Term 1	Term 2	Term 3	Term 4
Busqueda	3	1	0	2
Documento 1	2	1	1	2
Documento 2	0	3	2	0

JUICIO DE RELEVANCIA : *"EL DOC. 2 ES RELEVANTE, EL DOC. 1 NO"*

REFINACION DE LA BUSQUEDA

Busqueda	a	2	.5	1

Figura 13. Proceso de Retroalimentación de Relevancia donde las búsquedas están expresadas como términos con pesos.

La letra **i** denota los **R´** documentos juzgados como relevantes y la letra j.. los **N´** documentoe juzgados como no

relevantes. La búsqueda actualizada se especifica como el vector suma de la busqueda anterior más la diferencia de pesos entre el promedio de los términos relevantes y el promedio de los terminos no relevantes. Alfa, Beta y Gama son constantes determinadas empíricamente, de manera que produzcan una satisfactoria Retroalimentación de Relevancia. Q es el vector original de busqueda. DOC(i) representa el vector de pesos de los términos de un documento particular. Los pesos de los términos para cada documento son asignados de acuerdo a la importancia de cada término en el documento. Esto requiere un paso adicional en el proceso de indexacidn pues normalmente se considera únicamente la presencia o ausencia de términos en los documentos y no su importacia relativa en él. El vector DOC(i) representa todos los terminos de la Base de Datos, aparezcan o no en los documentos recuperados. La calificación de documentos como relevantes o no. es hecha por el usuario.

El efecto del segundo término en la Ecuación 1 es añadir términos al conjunto de términos relevantes o incrementar los pesos de los terminos ya juzgados como relevantes. A su vez, el efecto del tercer término de la ecuación es remover términos de la lista de términos relevantes o por lo menos reducir sus pesos.

En el sistema SMART, los documentos son representados gráficamente como vectores, en un espacio definido por los pesos de los términos asignados ellos (Figura 17). El requerimiento de busqueda es también un vector del espacio y la relevancia de los documentos se juzga de acuerdo a su distancia a él. El proceso de Retroalimentación de Relevancia corresponde a empujar el vector de requerimientos lo más cercano posible al centroide definido por los vectores de los documentos relevantes.

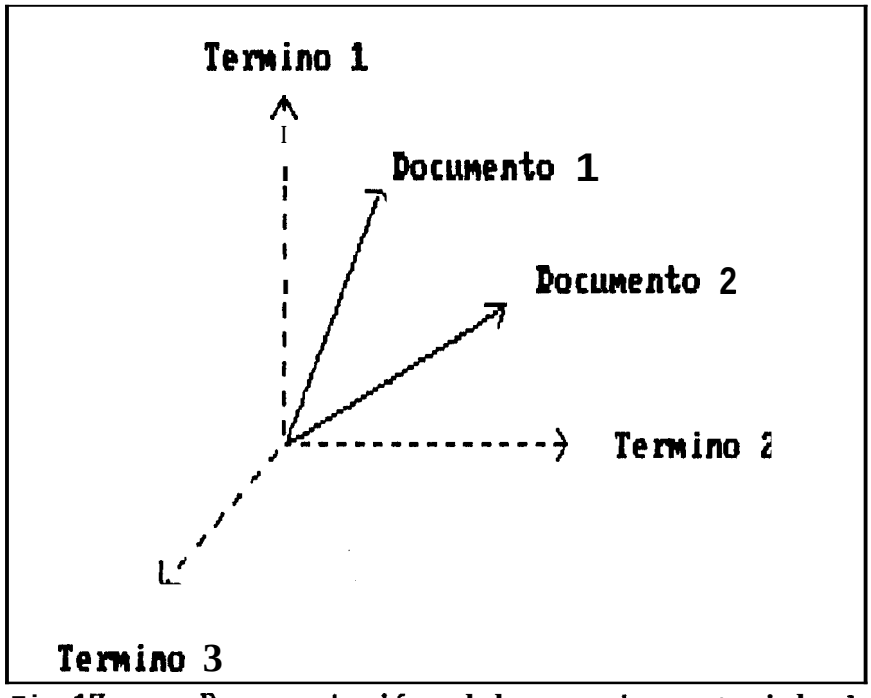


Fig 17. Representación del espacio vectorial de los documentos.

2.1.8 EVALUACION DE LA RECUPERACION DE INFORMACION.-

Los factores que pueden afectar el desempeño de una Búsqueda incluyen: la instalación, el usuario, el requerimiento de Gusqueda, la Base de Datos, el Sistema de Recuperación, el Proceso de Búsqueda, el intermediario, el resultado de la búsqueda, etc.

Tradicionalmente, el desempeño de la recuperación de información ha sido evaluado por rnedio de dos medidas: la Recolección y la Precisión. Estas medidas están típicamente basadas en juicios de relevancia realizados por los usuarios, a medida que inspeccionan los resúmenes de los documentos obtenidos. Un tercer criteria de evaluación es la Utilización [PCKW89].

A. **RECOLECCION:** Se define como la proporción de documentos relevantes de una Rase de Datos, que fueron recuperados. Es una medida un tanto dificil de estimar, pues el numero total de documentos relevantes en una busqueda particular, es usualmente desconocido.

B. **PRECISION:** Se define como la proporción de documentos que resultaron relevantes, dentro del conjunto recuperado. Esta medida es importante en una BDI.

pues ningún usuario quiere ser distraido con información irrelevante. El sistema debe poder restringir las decisiones de información y selección y determinar qué es relevante en base a las capacidades. talentos. background y experiencia del usuario y la naturaleza del trabajo que esta llevando a cabo.

En las Bases de Datos Textuales, métodos para incrementar la Precisión incluyen el uso de terminos de indexación más específicos ("PASCAL" en vez de "LENGUAJES DE PROGRAMACION") y la combinación de términos de indexación en conceptos más claros gracias a la utilización del operador lógico Y. Si bien se recomienda también el uso del operador NO para eliminar información no deseada, en la práctica es difícil eliminar información indeseable sin eliminar también información relevante.

Las Figuras 18 y 19 hacen una distinción entre: métodos que incrementan la Precisión durante la Exploración (hojeo) y métodos que incrementan la Precisión durante Búsquedas Estructuradas. En la exploración, un aumento de precisión envuelve filtrar los enlaces asociativos que emanan de cada documento. En la búsqueda estructurada,

aumentar la precisión significa modificar el algoritmo para el cálculo de la relevancia de cada documento.

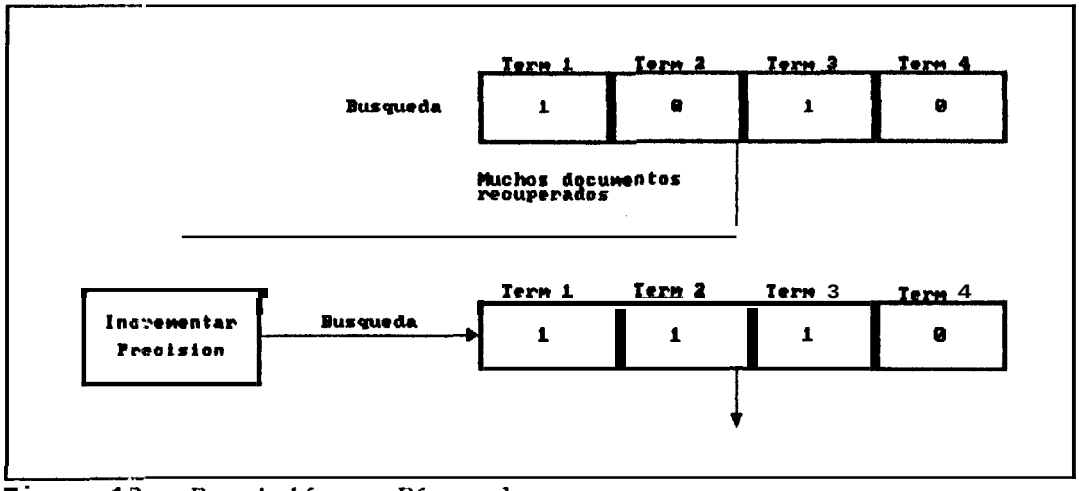


Figura 18. Precisión en Búsqueda.

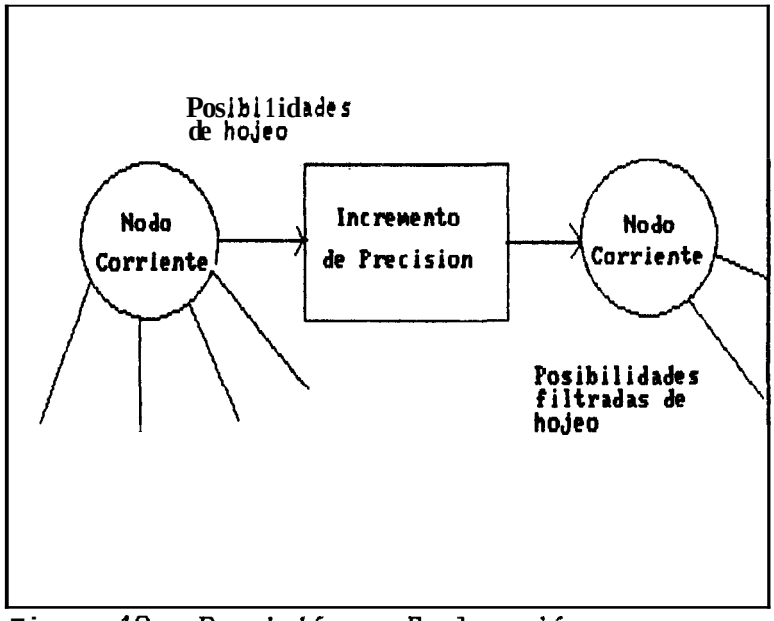


Figura 19. Precisión en Exploración.

C. **UTILIZACION:** La Utilizacibn se define como el número de documentos obtenidos en la búsqueda, que realmente fueron utilizados por el usuario. Es que un documento supuestamente relevante, no es bueno si el usuario no lo usa.

La Utilización puede expresarse en diferentes formas:

- Cuántos documentos fueron leidos por el usuario?
- Cuántos documentos fueron posteriormente citados por el usuario en artículos o reportes?

La efectividad de los Sistemas de Recuperación es un tópico controversial. En la practica las medidas de precisión y recolección son cercanas a la perfección. La recolección es, en la mayoría de los casos, inferior a la precisión. Las medidas de utilización son aún menores.

2.2 MANEJO DE TEXTOS.-

Los sistemas convencionales de Recuperación de Información están siendo beneficiados por el enfoque basado en conocimiento. Idealmente, un sistema de Recuperación de Información debe funcionar como un experto en conocimientos y un intermediario de búsqueda, además de llevar a cabo el almacenamiento y la recuperación de la información. Debe

basarse en conocimientos y especializarse en un dominio específico y en estrategias de búsqueda tales como usar un documento relevante para identificar los terminus de indexación que pueden ser usados para encontrar otros documentos relevantes.

2.2.1 MANEJO DE TEXTOS VS RECUPERACION DE INFORMACION.-

La Recuperación de Información, como es llevada a cabo corrientemente, es un proceso de interacción entre un Humano activo y una Base de Datos pasiva: la Base de Datos tiene la información, pero para conseguirla, el usuario debe hacer las preguntas correctas. Como ya vimos, existen dos canales de comunicación: uno entre el usuario y el intermediario, y otro entre el intermediario y la Base de Datos. A su vez, cada canal tiene asociadas doe direcciones de flujo de información.

Las cuatro combinaciones de canal y dirección del flujo representan cuatro tareas criticas para el desempeño del proceso de Recuperación de Información. Las dos primeras constituyen una negociación de la pregunta, mientras las dos últimas conciernen al acceso a la información de la Base de Datos en base a una búsqueda [PCKW89]:

A. El intermediario debe entender la búsqueda.

B. El Intermediario debe hacer al usuario preguntas inteligentes que le permitan refinar la búsqueda inicial.

C. El intermediario traduce el requerimiento de búsqueda en comandos del lenguaje del Sistema de Recuperación en Linea.

D. La totalidad del Sistema de Recuperación debe asegurar que el intermediario reciba buenas respuestas del requerimiento de búsqueda y que las entienda.

En los Sistemas convencionales de Recuperación de Información, la última de las tareas es ignorada. La salida de la búsqueda es simplemente un conjunto de documentos. completamente determinados por la forma del requerimiento. Pero, en un sistema más flexible de Manejo de Textos. esta cuarta tarea se vuelve críticamente importante. La Rase de Datos, asumiendo que tiene inteligencia, debe pasar al intermediario información relacionada con la disponibilidad de diversos tipos de conocimiento. y no simplemente listar un conjunto de documentos relevantes. Además, la Rase de Datos debe poder modificar el requerimiento de búsqueda, en base al conocimiento actualmente disponible en ella. Asi pues, en un Sistema de Manejo de Textos mucha de la distinción entre el intermediario y la Base de Datos desaparece y hay una migración de inteligen-

cia desde el intermediario hacia estructuras de conocimiento en la Rase de Datos.

Para muchos, el rol central de la inteligencia está en la experioncia. En este sentido, el desarrollo de sistemas de recuperación de textos debería ser facilitado por la gran cantidad de información existente en las Rases de Datos. El gran desafio de la ingenieria es determinar cómo transformar toda esa información en conocimientos útiles. La tarea no es facil, pues la información de las Bases de Datos consiste fundamentalmente en hechos y palabras claves que repreuentan a los documentos en una forma relativamente desestructurada. Es necesaria una estructura, una organización, que permita transformar la información en conocimiento. Así pues, la ventaja que representa la cantidad de información de las Bases de Datos es contrapuesta por la complejidad y variedad de esa misma información.

2.2.2 INDEXACION CONCEPTUAL.-

Muchas tareas dentro de los campos de Rases de Datos e Inteligencia Artificial enfatizan la importancia de representar el conocimiento en una forma compatible con el comportamiento inteligente. No es de interés el lenguaje en el que el conocimiento es almacenado (Ejs: predicados

de cálculo, reglas de producción, estructuras o dependencias conceptuales), sino las operaciones que pueden ser llevadas a cabo desde las estructuras de conocimiento.

Algunas Bases de Datos representan a los documentos por medio de un resumen, palabras claves y términos de indexación. En este caso, los términos de indexación consisten de: términos de un vocabulario controlado y sentencias no controladas asociadas a ellos. Ej:

Termino de Indexación: *MATRICES*
Sentencia asociada: *Suma de.. de igual dimensión*

Traducción: "Suma de Matrices de igual dimensión"

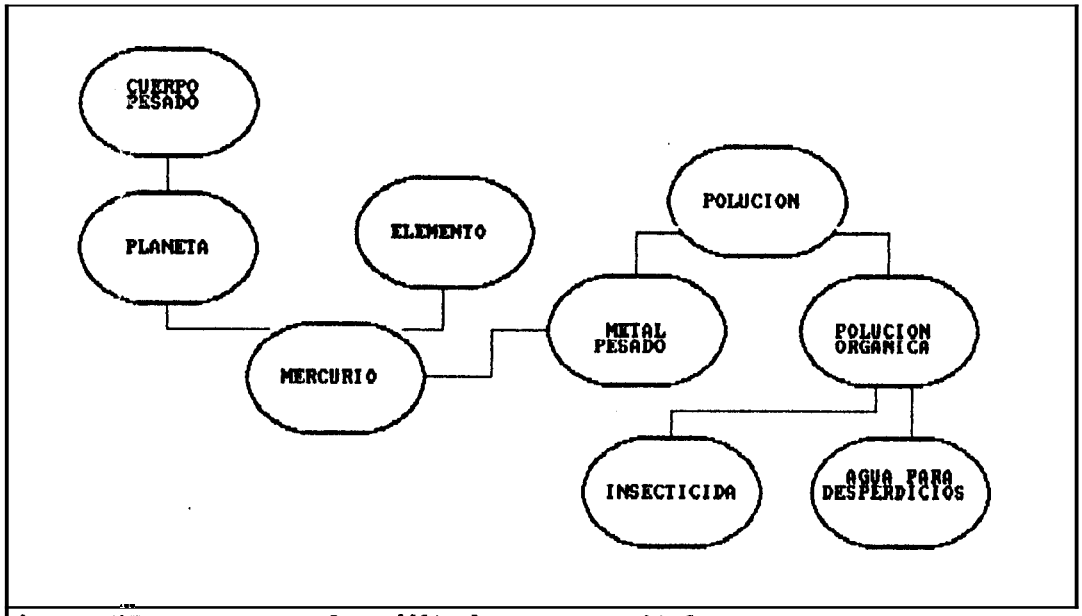
Para este ejemplo, el término de indexación representa una declaración proposicional sobre el conocimiento contenido en un documento. Estos términos de indexación pueden ser convertidos en framas de una Base de Datos Relacional y constituir asi un compromiso entre las palabras claves y el lenguaje natural. Los framas limitan la representación del conocimiento de manera que éste sea accesible, a un costo razonable de procesamiento.

Sin embargo, la creación de una Base de Datos Relacional no garantiza por sí sola la accesibilidad al conoci-

miento/información. Es necesario traducir el requerimiento de búsqueda del usuario en una forma reconocible por la Base de Datos. Esta traducción requiere reorganizar apropiadamente el conocimiento, dentro de la Rase de Datos Relacional, de manera que ésta sea capaz de crear relaciones entre, y clasificaciones de, los términos que contiene.

Una rnanera de realizar esta reorganización es mediante la construcción de una jerarquia [PCKW89]. Los que defienden que el uso de jerarquias es restrictivo, argumentan efectivamente que muchos conceptos pueden aparecer en un numero diverso de jerarquias, dependiendo de la perspective de la organización. La solución a este problema es reconocer que un concepto no se relaciona a una única organización jerárquica y que organizaciones jerárquicas diferentes pueden uuperponerse a lo largo de los niveles de una estructura compleja de red, tal y como se sucede en los Documentos de Hypermedia.

Las Figuras 20 y 21 ilustran un método para pasar de una perspectiva jerárquica a otra. Como se puede apreciar, cada nodo comprende un número de jerarquias diferentes. Movilizándose hacia arriba o hacia abajo de la jerarquia, el usuario puede escoger: tratar conceptos relacionados,



igura 20. Red: Método para switcheo cntre perspectivas jerárquicas.

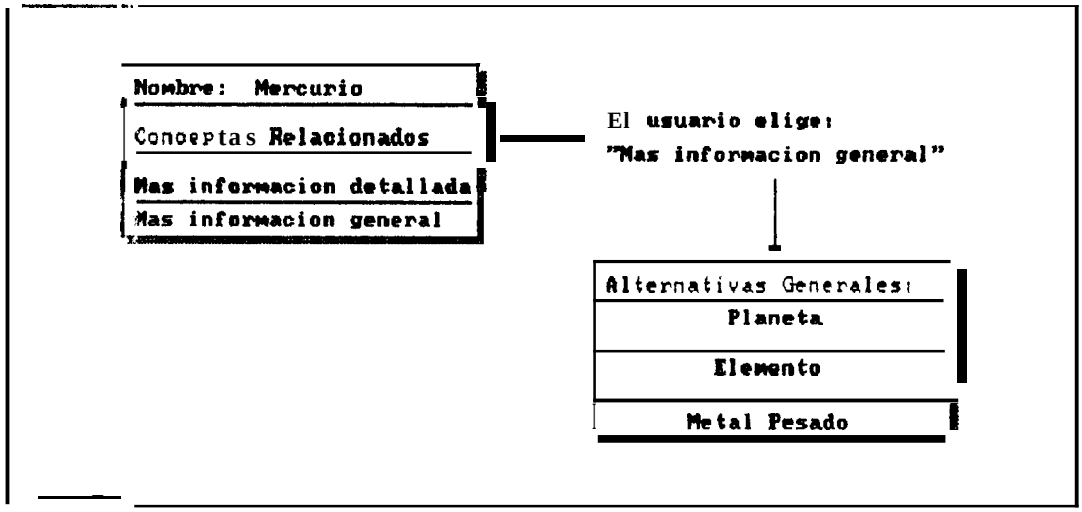


Figura 21. Selección: Método para switcheo entre perspectivas jerárquicas.

explorar versiones más detalladas del concepto actual o pasar a versiones mas generales de él.

A. INDEXACION CONCEPTUAL CON ´FRAMES´ JERARQUICOS.-

Una jerarquía conceptual puede ser construida de muchas maneras. Un método es proponer a una serie de expertos en un tema que ordenen una colección de términos dentro de una jerarquia. Estos términos se pueden ohtener de fuentes de referencia estándard. Durante el proceso de actualización y expansión de la jerarquía, los expertos humanos pueden utilizar interactivamente técnicas *en* Gase a frecuencias estadísticas y coocurrencia de palabras de la Base de Datos.

Un método alternativo es decidir previamente cuáles son los conceptos claves y luego encontrar la manera de reconocerlos dentro del texto. Este es un problema difícil, pero no equivalente a la terrible complejidad de tener que entender el lenguaje natural.. El sistema sólo estaría buscando en el texto, pistas que le indiquen la presencia de conceptos generales; no seria necesaria una interpretación detallada del significado del documento.

Este tipo de indexacibn conceptual requiere una representación del dominio del conocimiento. de la manera típica de los Sistemas Expertos. Asi pues, muchos de los conceptos usados para indexar una Base de Datos sobre ´Redes

Locales', perteneceran también a la Base de Conocimiento de un Sistema Experto que trate el mismo tema. Idealmente. la Base de Conocimiento de un indexador debe representar el conocimiento en una forma completa. Además, los conceptos indexados deben ser representados de tal manera que se permita su fácil acceso durante las búsquedas.

Una forma de representación que cumple estos requerimientos es la jerarquía de framas. Cada nodo de 1a jerarquía es una instancia particular del concepto que define la estructura total: los nodos subordinados y superordinados constituyen a EU vez instancias mas especificas o más generales. respectivamente. Implementando cada nodo como una frama, las propiedades de cada instacia pueden ser almacenadas localmente y también propagadas hacia abajo. a los sucesores de la jerarquía. Mas importante aún. las relaciones semanticas entre instancias son preservadas y pueden ser usadas para estructurar la búsqueda.

Las framas contienen campos con los datos de las instancias especificas. En general, uno de estos campos o atributos apunta A una categoria superordinada a la cual la frama corriente pertenece. Otros campos apuntan a framas subordinadas que constituyen instancias especializadas de

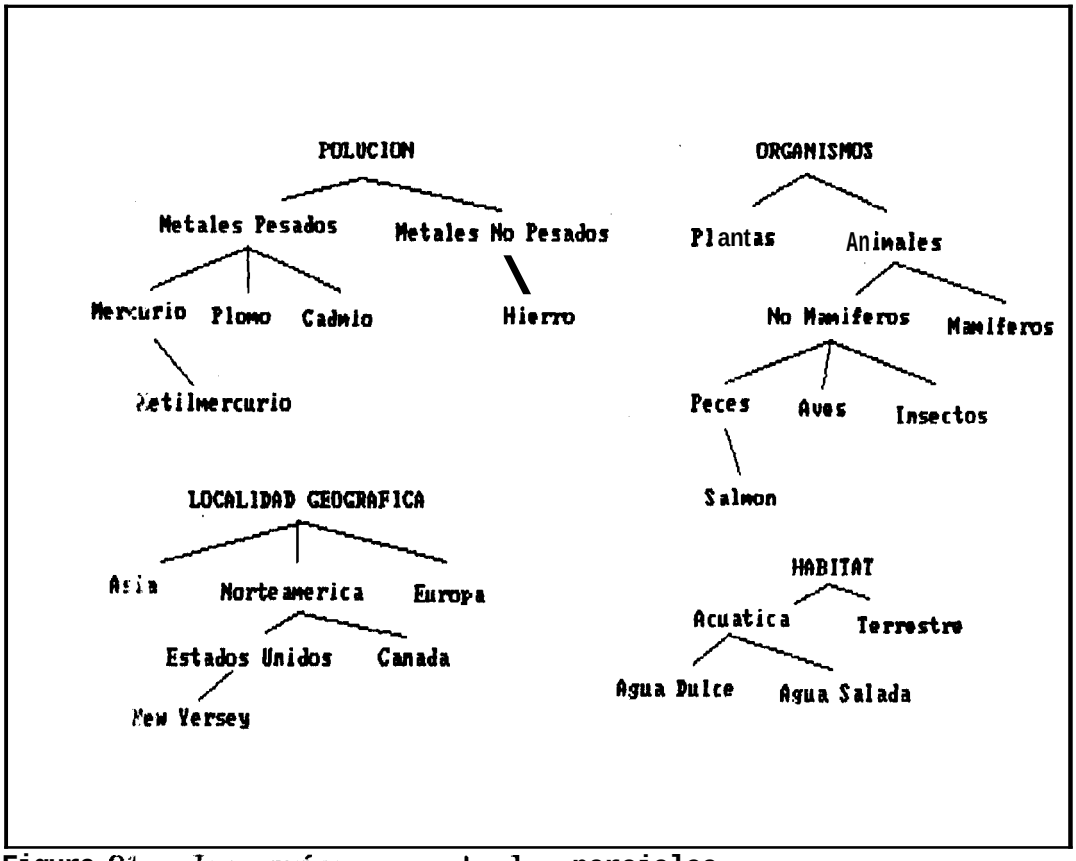


Figura 21. Jerarquías conceptuales parcialea.

la frama corriente. Estos campos especiales, llamados campos padres y campos hijos, permiten definir categorías y subcategorías de las framas dentro de una jerarquía. En el ejemplo de la Figura 22, los Metales Pesados son subordinados a la frama de Polución.

Utilizando la siguiente sintaxis para las framas [PCKW89]. la frama de un documento sobre Contaminación Ambiental. se veria asi:

´Nombre de la frama´		*Documento N*	
Campo 1:	*Dato 1*	*Polución :*	*Cadmio*
Campo 2:	*Dato 2*	*Localidad:*	*Grandes Lagos*
		Organismo:	*Bacilo X*
Campo N:	*Dato N*		

Adicionalmente a los datos que se colocan en los campos, las **framas** pueden contener información referente a procedimientos vinculados. Estos procedimientos actuan como relaciones y especifican cómo los atributos o campos de una frama son afectados por los valores de los campos de otras **framas**. Ej: Un procedimiento "si .. entonces". que actúe como regla, diciéndole al sistema qué hacer cuando información ha sido ingresada a cierto campo.

En una jerarquía, los objetos forman grupos de acuerdo a sus propiedades comunes. Esto da lugar a la Herencia. Ej: En el campo de la Quimica sabemos que Cadmio es un tipo de Metal **Pesado**, por lo tanto esperamos que retenga todas las propiedades básicas de los Metales y de los Metales Pesados, aunque con algunas restricciones.

Así pues, la combinación de framas, procedimientos vinculados y herencia permite desarrollar una estrategia flexible **para** construir un sistema de indexación basado en conceptos. La gran ventaja de las framas es **que** por ser

estructuras de datos muy flexibles, soportan una diversidad de demandas. **Para** expresar las propiedades que una frama debe tener para ser la instancia de un concepto, sólo es necesario un procedimiento vinculado que valide cada requerimiento.

Por otro lado, la preservación de relaciones semánticas entre framas elimina algunas de las dificultades de la interacción con Sistemas tradicionales de Recuperación el Linea. Ej: En vez de tener que nombrar todas las especies de mamíferos, un sistema basado en conocimiento, con jerarquias de **conceptos**, puede usar relaciones semánticas e inferir que al ingresar la palabra ´mamiferos´, el usuario está interesado en todas sus especies.

Finalmente, si además de propiedades, se almacenan en cada frama de la jerarquía, una serie de sinónimos, se estaria resolviendo el problema de **conceptos** que puedan ser expresados con diferentes palabras o frases. Cada nodo seria visto como un término de indexación.

B. INDEXACION CONCEPTUAL INTERACTIVA.-

Uno de los argumentos en contra de la indexación conceptual en el Manejo de Textos, es que es un proceso dema-

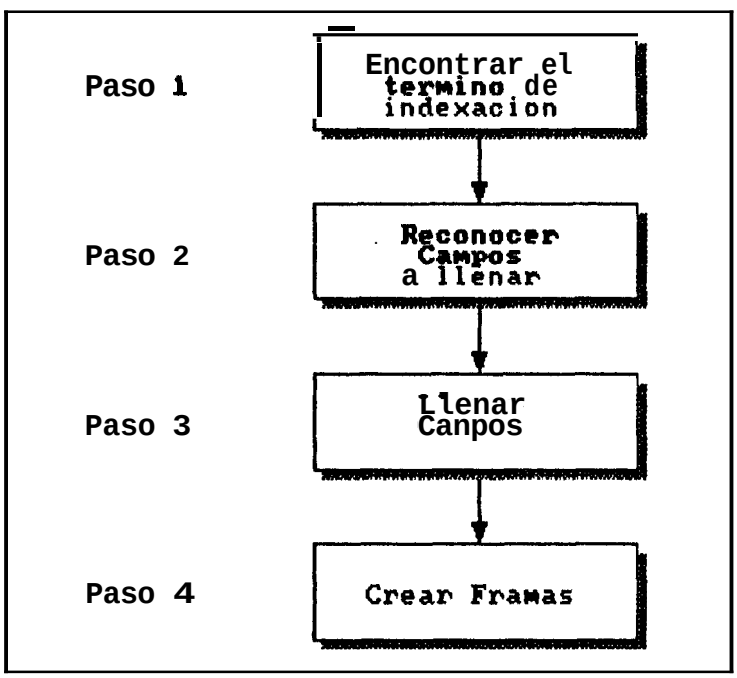
	INDEXAMIENTO MANUAL	INDEXAMIENTO INTERACTIUO	INDEXAMIENTO AUTOMATIZADO
HUMANO	1. Revision del Documento 2. Asignacion do Terminos	3. Asistir oon el indexa-miento dificil 4. Monitorear y Editar el indexamiento	Ninguna tarea humana
MAQUINA	Ninguna tarea de maquina	1. Revision del Documento 2. Asignacion dr Terminos	1. Revision del Documento 2. Asignacion de Terminos

Figura 23. Tres aproximaciones a la indexación.

siado laborioso para ser llevado a cabo en grupos de muchos documentos. Los indices conceptuales hechos a mano aumentan la dificultad de la tarea de indexación y por otro lado, los métodos corrientes para entender el lenguaje natural no permiten la automatización completa de la indexación conceptual. La mejor solución parece ser una forma interactiva, que haga un balance entre la necesidad de eficiencia y la necesidad de supervisión humana en el análisis de textos y el proceso de indexación conceptual. Es que, si bien la indexación automatizada es mas consistente que la manual, también es mas compleja y conceptos sensibles al contexto son mejor manejados por indexadores humanos.

La Base de Datos CAS en Línea [PCKW89] desarrolló un método para convertir términos de indexación en framas

conceptuales, todo dentro de un conjunto limitado de 874 documentos y tratando un tema específico. Primeramente, se identifica el término de indexación relevante. Luego, usando análisis sintactico, el término de indexación y sus sentencias asociadas son relacionados y se determina qué campos necesita la frama. Finalmente se reconocen los conceptos que llenan cada campo y se construye la representación correspondiente de la frama, para el documento.



igura 24. Método para convertir términos de indexación en frames conceptuales.

Para Bases de Datos con gran cantidad de documentos, se pueden utilizar primeramente los métodos convencionales

de Recuperación de Información y así, indentificar un conjunto de unos pocos miles de posibles documentos relevantes. Luego, en vez de aplicar los métodos estándard de Retroalimentación de Relevancia para refinar la búsqueda e incrementar la precisión, el grupo de documentos es pasado a un sistema Manejador de Textos. Este construye un índice conceptual para ese tópico.

En general, podemos distinguir cuatro diferentes modos de indexación interactiva dependiendo del tipo de relación que exista entre el indexador humano y el indexador máquina [PCKW89]:

1. El indexador humano refina el resultado de la indexación automatizada.
2. La máquina refina el trabajo del indexador humano.
3. La máquina es una herramienta de ayuda y un supervisor del indexador humano.
4. El liumano es un supervisor, un maestro y un consultor técnico para el indexador máquina.

A continuación, abandonamos los fundamentos teóricos de la implementación de Sistemas de Recuperación de Información y Manejo de Textos y Documentos y pasamos a analizar su realidad en el Mercado.

2.3 SOFTWARE EXISTENTE PARA MANEJO DE DOCUMENTOS.-

Si bien no toda empresa necesita los servicios de un Sistema de Manejo de Documentos, cualquier organización que genere muchos papeles puede beneficiarse de un buen Sistema de este tipo. No es necesario tener mucha imaginación para ver cómo la oficina de un medico podría ahorrar tiempo y dinero dejando a la computadora manejar las fichas de los pacientes: oficinas de ingenieria podrían organizar sus gráficos y planos; revistas y periódicos que manejan gran cantidad de artículos, podrían accesar un texto sobre "BUSES MCA" publicado "en algun momento del año pasado".

Recientes encuestas revelan que una significativa minoría de usuarios de redes usan **Software para** Manejo de Documentos [DUNC91]. Esto se debe a que el manejo de multiples documentos dentro de una red es una rama relativamente nueva de la industria misma de las redes. Estudiando los diversos productos del mercado, se descubre que existen santas maneras **para** manejar documentos, como compañías los producen. Una explicación parcial para estas observaciones es que la industria de los documentos es muy Joven, con menos de 10 años de existencia, en los cuales no se han establecido estandards.

En el momento de comprar un software para Manejo de Documentos es necesario realizar las siguientes preguntas:

A. **Es el producto de tipo "front-end" o de tipo "back-end'?**

Un producto "front-end" se encuentra entre el usuario y el documento. El usuario debe llenar ciertos campos (como cliente o autor) para llegar a tener acceso sobre el archivo. Un aspecto positivo de lor productos "front-end" es que forzan al usuario a acatar una serie de convenciones para nombrar A los archivos y directorios. Como consecuencia. los beneficios de este sistema supervisor son obvios: todos los archivos de texto tienen extensiones similares y residen en subdirectorios con nombres similares, dentro del directorio personal del usuario.

Para usuarios con fobia hacia las computadoras, el enfoque "front-end" proporciona una interfase consistente. Pero por el lado negativo, existe una leve curva ascendente de aprendizaje, pues el usuario se tiene que enfrentar de repente a una pantalla de ingreso del perfil del documento.

Un producto "back-end" no requiere que los usuarios llenen un perfil hasta el momento de abandonar el documen-

- Pueden los usarios realizar búsquedas "Full-Text" de cualquier palabra del documento? Esta es la tarea más usada de un Manejador de Documentos. Los usuarios pueden olvidar los nombres de los archivos, pero siempre son capaces de recordar al menos una palabra. en algún lugar del documento.
- Pueden los usuarios buscar por medio de palabras claves predefinidas?
- Pueden los usuarios refinar sus búsquedas, usando las técnicas booleanas ("Y", "O". ">". "<". etc) ?
- Puede el producto buscar "por similitud de sonido"?
- Si el usuario no recuerda haber escrito la palabra "cien" o usar los digitos "100". Podrá el producto encontrar cualquiera de las dos variables?

D. Cuál es la relación entre el tamaño del lndice y el de la Ease de Datos?

Como ya vimos, la indexación es la tarea de los Sistemas de Manejo de Documentos que analiza toda la Rase de Datos y crea un índice sobre qué palabras están en qué documentos. Asi pues, un índice es algunas veces más pequeño que la Base de Datos original. pues cada palabra diferente es grabada solo una vez y sus posiciones en los varios documentos son referenciadas. Diferentes esquemas

de indexación producen indices de diferentes tamaños, siendo su tamaño óptimo entre cinco y diez por ciento de la Base de Datos total. Obviamente, ya que las búsquedas se realizan verdaderamente en los índices, aún en los sistemas con capacidad de búsqueda "Full-Text", mientras mas pequeño el indice, más rapidamente se encuentran los documentos.

E. Existe Control de Versiones?

Esta capacidad trata si el sistema acostumbra mantener en el servidor, versiones antiguas del **documento** y permite accesarlas de **ser** neceeario. Este servicio puede ser considerado un desperdicio de la memoria del servidor. Sin embargo, permite accesar párrafos que fueron eliminados de la versión corriente.

2.3.1 METODOLOGIA DE PRUEBA DR LOS MANEJADORES DE DOCUMENTOS.-

La revista LAN TIMES presentó en su sección de "Comparación de Productos" un analisis sobre cinco Manejadores de Documentos existentes en el Mercado [DUNC91]. Estos productos pertenecen a las compañías: Soft Solutions, Intel, Interpreter, Internet **Software** y PC-Docs. A continuación, explicaremos las pruebas realizadas por los Laboratorios

LAN TIMES sobre estos paquetes y las observaciones y resultados obtenidos por ellos.

En primer lugar se tomó en cuenta, que el desempeño de un producto no depende sólamente de la velocidad de uno de sus procesos. Ej: Un proceso de una aplicación, una vez ejecutado, puede ser dos veces mas rápido que el mismo proceso en otra aplicación. Pero si el usuario tiene que pasar por cuatro submenús para poder llegar a ejecutar el proceso de la primera aplicación, esto significa que ha invertido una cantidad de tiempo, sin ejecutar nada. Asi pues, LAN TIMES utilizó las medidas de desempeño como indicadores de la eficiencia del código del programa.

La primera prueba de desempeño tabula el tiempo que le toma a la aplicación importar un conjunto de documentos ya existentes. Esta prueba toma en cuenta el hecho de que la mayoría de las personas que empiezan a trabajar con un Manejador de Documentos, tienen ya una serie de documentos que necesitan cargar en él. Se utilizaron 11 MB de datos, contenidos en 1130 archivos Wordperfect. Cada perfil de documentos contenia la siguiente información: nombre del archivo, descripción del archivo, fecha y hora de creación,

nombre del autor y espacio para un comentario del archivo.

Dentro de la prueba de importación de documentos, se incluyó también el análisis del tamaño resultante de la Base de Datos, como una consideración de la utilización del disco. Se tomó el tamaño de la Rase de Datos, we lo dividió para el número total de documentos y se obtuvo una medida de 'caracteres por documento'.

La segunda prueba de desempeño se refiere a la indexación "Full-Text". Este proceso puede consumir mucho tiempo y llegar a causar una degradación innecesaria en el desempeño de la red. Los Laboratorios LAN TIMES tomaron el tiempo al proceso de indexación de un pequeño grupo de diez archivos con un total de 259 KB. Paralelamente analizaron el tamaño del índice resultante.

Una capacidad estándard de los Manejadores de Documentos, es la habilidad de cargar aplicaciones automáticamente, mediante la simple selección de un documento para edición. Así pues, como tercera prueba de desempeño se midió, para cada producto, el tiempo que le toma cargar automáticamente Wordperfect y abrir el documento seleccionado. Se encontró una diferencia de cinco segundos entre el programa

2.3.2 CARACTERISTICAS DE LOS PROWCTOS.-

A continuación presentamos un resumen de las observaciones hechas por los Laboratorios LAN TIMES. al respecto de cada uno de los Productos analizados [DUNC91]. Esto nos permite tener una idea de las características y posibilidades que ofrecen los Manejadores de Documentos existentes en el mercado.

A. INTERPRETER: DOCUMENT ADMINISTRATOR

A.1 CAPACIDADHS: DOCUMENT ADMINISTRATOR realiza las tareas estándard de manejo de documentos en forma bastante efectiva. Perfiles y Búsquedas "Full-Text" son disponibles. Existe además una opción de Búsquedas Próximas, utilizada cuando se tienen documentos muy similares y se necesita identificar uno específicamente.

Otra característica valorable es un Manejador de Tareas, que permite automatizar tareas del sistema y procedimientos para tareas del hogar. Esta capacidad hace facil planificar actividades para las noches o fines de semana.

A.2 TRABAJO EN REDES: DOCUMENT ADMINISTRATOR se destaca en este campo. Primeramente, soporta diversos Sistemas Operativa para Redes: Banyan. 3Com. IBM PC Net. Nestar

Nex. y otros basados en DOS 3.1. En segundo lugar, ofrece una gran integración can Netware y Novell. Ej: incorpora usuarios y niveles de seguridad definidos en Netware.

A.3 INSTALACION Y USO: DOCUMENT ADMINISTRATOR es placentero de utilizar. Tiene un sistema intuitivo de menus que permiten movilizacion rápida y fácil. Destaca la facilidad de saltar de un menu a otro, aún si los submenus del primer menu están todavia listados en la pantalla. Esto es más eficiente que los programas que obligan a un usuario a retroceder hasta salir de un menú. **para** luego poder entrar en otro. El sistema incluye además diversos mensajes de ayuda que faoilitan el trabajo de los usuarios principiantes.

La instalación es confiable, pero podría ser mejorada automatizando los cambios al archivo Config.sys y no requiriendo que el Administrador utilice Syscon para crear un usuario.

A-4 DOCUMENTACION: **La** Documentación **no** es de gran calidad. El manual mezcla los tutoriales y el material de referencia, de una manera frustrante **para** los principiantes. El ideal de un usuario es seguir un tutorial y avanzar en él paso a paso hasta comprender una tarea. El

manual de DOCUMENT **ADMINISTRATOR** saltea la información, que aunque es buena, deberia estar en los lugares apropiados. Por esta misma razón, **es** fácil perderse a través de las instrucciones de configuración. Algunas veces se termina un capítulo y el usuario todavía tiene un menú de configuración en la pantalla. Otras veces, el manual asume que el usuario está viendo un cierto menú, pero él nunca recibió estas instrucciones específicas.

A.5 DESEMPEÑO: El programa es muy eficiente. En la prueba de Importación, DOCUMENT ADMINISTRATOR fue el ganador. También tuvo buenos tiempos en la indexación y en la carga de la aplicación.

B. INTERNET SOFTWARE: NETADVANTAGE

B.1 CAPACIDADES: NETADVANTAGE almacena la información sobre los archivos de documentos en una Rase de Datos llamada Docuhase´. Probablemente la información más valiosa de ´Docubase´ es la historia de los archivos, que registra los eventos especiales y los cambios realizados sobre ellos. ´Docubase´ puede listar cuándo y quién modificb un srchivo y lleva records de las últimas veces que el archivo fue impreso, copiado, eliminado o respaldado.

Antes de que NETADVANTAGE permita crear un documento, el usuario debe llenar dos pantallas de información sobre el. Esta información sirve a 'Docubase' para registrar el documento. A continuación, NETADVANTAGE carga la aplicación y crea un documento. Soporta diversos tipos de archives: Lotus, Wordperfect, Word, OfficeWriter, y ASCII.

Las búsquedas se pueden realizar en una de dos maneras. La primera es la búsqueda rápida, la cual permite al usuario colocarse en un determinado campo de búsqueda, escribir en él la información deseada y presionar la tecla <ENTER>. Ej: Si posicionado en el campo de Nombre del Autor, el usuario escribe un nombre y presiona <ENTER>. NETADVANTAGE presenta en pantalla el primer documento con ese nombre de autor.

La segunda y más flexible búsqueda implica llenar una pantalla de selección. En esta pantalla se pueden construir complejos criterios de búsqueda, por medio de la combinación de fechas de archivos, campos de 'Docubase' y eventos de los documentos.

NETADVANTAGE permite abandonar un archivo inactivo, pero mantener en línea información sobre él, como por ej:

el lugar dónde está almacenado. También es posible mover grupos de archivos de un directorio a otro y dejar a NETAD-VANTAGE actualizar automáticamente la ´Docubase´.

NETAPVANTAGE lleva cuenta de los tiempos invertidos en cada documento así como el número de páginas que de él han sido impresas. Estas capacidades son útiles si los costos de crear un documento deben ser cobrados a algun cliente en particular.

Durante la instalación, NETADVANTGE permite al usuario definir hasta seis **campos** y añadirlos a la estructura de ´Docubase.. Estos campos permiten adecuar NETADVANTAGE a las propias necesidades.

El punto en contra de NETADVANTAGE es nu soportar la indexación "**Full-Text**".

B.2 TRABAJO EN HEDES: NETADVANTAGE incluye un menú con los once, más comúnmente usados, comandos de **Red.** De esta manera, los usuarios no tienen que abandonar NETADVANTAGE para ejecutar un comando.

Como medio de seguridad, NETADVANTAGE usa el esquema de derechos de Netware para asi mantener la privacidad de los archivos; Si no se puede ingresar a un directorio desde la linea de comando, entonces NETADVANTAGE tampoco permitirá esa entrada. Se podrá observar el árbol del directorio, pero si se trata de ver algún archivo de éste sin la autoridad necesaria. un mensaje de "Directorio Inválido" aparece en pantalla.

B.3 INSPALACION Y **USO:** Instalar el software en el sistema es extremadamente facil: Se utiliza el comando INSTALL. desde el prompt **del** Sistema Operativo y luego se ejecuta por primera vez la rutina de Setup.

Si se decide cambiar la configuraci3n de 'Docubase.. se perderá toda la información de descripción de los documentos ya ingresados. Es mejor **leer** todo el manual y luego probar diversas instalaciones. **Luego** de decidir cómo se quiere grabar la información sobre los archivos, se puede configurar la verdadera instalación de NETADVANTAGE.

NETRDVANTAGE es muy facil de usar, pues cuenta con una interfase intuitiva **para** el usuario.

B.4 DOCUMENTACION: La documentación consiste en un unico manual, por el cual se puede navegar fácilmente mediante el uso del Indice, el Glosario y la Tabla de contenidos. Sin embargo, usar el manual para instalar el programa es toda una faena, pues éste direcciona diferentes secciones. En general, la documentación tiene más información de lo que uno puede desear saber sobre NETADVANTAGE.

B.5 DESEMPEÑO: Los tiempos de importación y la utilización del disco, si bien no fueron los más altos, están cercanos a la cima. El desempeño general está en los límites aceptables.

C. PC-DOCS: PC-DOCS

C.1 CAPACIDADES: Además de las capacidades que se esperan de todo Manejador de Documentos, PC-DOCS tiene algunas innovaciones. En la Búsqueda por Perfil del Documento existe la opción de presentar una lista de los documentos que cumplen por lo menos uno de los criterios de búsqueda. Una vez presentada esta lista, se puede: recuperar cualquiera de los documentos, buscar documentos adicionales o, aplicar a la misma lista otro criterio de búsqueda creando así un subconjunto de la lista previa.

El programa permite manejar hasta 99 versiones de un mismo documento. PC-DOCS crea también subversiones, cuando se graba a una versión previa de un documento. De ser necesario, se mantienen hasta subversiones de subversiones.

PC-DOCS puede marcar diferencias entre dos documentos similares. Esta capacidad es util para comparar dos versiones o subversiones de un mismo documento.

C.2 TRABAJO EN REDES: PC-DOCS fue diseñado para trabajar en Netware, y lo hace eficientemente. Un usuario de PC-DOCS se conecta en una red y puede inmediatamente buscar y recuperar documentos en una área lejana (WAN), sin tener que conectarse con servidores separados de archivos o Map Drivers. Sin embargo, todo paquete de software debe soportar diversos Sistemas Operativos para redes y PC-DOCS sólo soporta Netware.

C.3 INSTALACION Y USO: PC-DOCS utiliza menús de tipo Netware. Estos presenta familiaridad para usuarios Netware, pero son algo **arcaicos** para los estándards actuales. Entrar y **salir** de un submenú es algo engorroso y uno no tiene siempre seguridad de dónde está. A pesar de esto, el programa es amigable y no toma mucho tiempo sentirse con-

fortable con **61.** El manejo de errores ofrece explicaciones detalladas y claras de los mensajes de **error.**

El programa se instala fluídamente y **hay** una gran facilidad para importar usuarios Netware.

C.4 DOCUMENTACION: El manual está dividido en tres secciones lógicas y convenientes. Existen frecuentes encabezados y un buen índice, de manera que el usuario puede obtener ayuda rápidamente. Por el lado negativo, la sección de tutorial explican a veces qué hacer, sin asegurarse en qué menú se encuentra el usuario.

C.5 DESEMPEÑO: PC-DOCS fue el segundo en la importación de textos y el ganador en la indexación, haciendo el trabajo dos veces más rápido que los demás. Necesita mejorar en la carga de la aplicación, pues los usuarios de Wordperfect deben esperar algunos segundos ántes de poder entrar a un documento.

D. INTEL CORP: REFERENCEPOINT

D.1 CAPACIDADES: Los documentos se organizan en una librería principal, compartida por todos los usuarios, y en librerías personales para cada usuario de la red. REFE-

RENCEPOINT trabaja unicamente con documentos situados en una librería personal. Así pues, para modificar archivos de la librería principal. estos deben pasar a una librería personal Ningún otro usuario podrá utilizar un archivo sacado de la librería principal. hasta que éste regrese a ella. Por otro lado. es obvio que los documentos persona-les de un usuario se hallan en su propia librería.

REFERENCEPOINT incluye una interfase de menús con soporte de mouse. La pantalla está dividida en ventanas de tamaño ajustable, las cuales pueden presentar diversa información, al mismo tiempo.

El nombre del documento, su título y autor son grabados en una 'Referencecard'. compuesta por diversos campos. REFERENCEPOINT permite que 16 de estos carnpos sean definidos por el usuario. de manera que éste pueda adecuar esas entradas y grabar en ellas información especifica para su Compañía. REFERENCEPOINT ofrece más campos definibles que cualquier otro paquete probado.

Los archivos nuevos son creados en el espacio personal de trabajo. Seleccionar la Opción New del Menu File. crea una nueva 'Referencecard'. Se pueden llenar entonces los

campus, yroporcionando información relevante sobre el archive. REFERENCEFOINT mantiene una lista *de* entradas aceptables **para** cada campo. Si un campo es requerido, REFERENCEPOINT forza su ingreso, escogiendo de una lista de alternativas. Existen campos automatizados. donde el Sistema puede ingresar datos predeterminados, como nombre del autor y nombre del archivo, sin intervención del usuario.

REFERENCEPOINT soporta indexación "Full-Text". pero **esta** opción debe ser **adquirida** separadamente. Tomando en cuenta la importancia de esta tarea, debería ser incluída en el paquete de Manejo de Documentos.

REFEHENCEFOINT, como todo Manejador de Documentos estándard, invoca automáticamente una aplicación como Wardperfect: cuando se selecciona un documento **para** edición. Esto lo hace mediante archivos tipo batch, fácilmente modificables por el usuario.

D.2 TRABAJO EN REDES: REFERENCEFOINT es un sistema abierto que corre en la mayoría de Sistemas Operativos para red. Es compatible con cualquier red compatible con DOS incluyendo Novell Netware, 3Com 3+. Banyan Vines. y LAN

Manager. Las estaciones de trabajo requieren 640KB y DOS 3.0 o mayor.

A pesar de que REFERENCEPOINT es bueno reconociendo redes Novell, no es tan bueno sacando información de ellas. Uno debe de recordar todos los nombres de usuarios y el mapeo de drives. Este es un proceso tedioso considerando que otros programas permiten escoger de una lista de usuarios de la red.

Además de los derechos de seguridad establecidos por el Sistema Operativo de la red. REFERENCEPOINT añade otro nivel de seguridad. El administrador del sistema puede otorgar a cada usuario derechos sobre los documentos. Estos van desde la simple capacidad de leer. hasta completos privilegios administrativos.

D.3 INSTALACION Y USO: El proceso de instalación se lleva a caho en dos pasos. El primer paso envuelve: copiar todo el software en el drive de la red. crear las necesarias estructuras de directorios e ingresar el comando INSTALL. El segundo paso de instalación toma más esfuerzo. Envuelve configurer el ambiente de REFERENCEPOINT para alcanzar específicas necesidades. Se debe ingresar en la red como

supervisor, definir usuarios, adecuar los campos de las 'Referencecards' e incluir aplicaciones. Este proceso de instalación es demasiado complicado.

En cuanto a la interfase de menus. aunque el soporte del mouse es bueno, no es tan amigable como otras.

D.4 DOCUMENTACION: La documentación es buena. Sin embargo, las explicaciones de los comandos necesitan ejemplos que las ilustren.

D.5 DESEMPEÑO: El proceso de importación de documentos fue el más lento de todos. Esto se debe a la gran cantidad de campos de la 'Referencecard' yue son definibles por el usuario. El desempeño es promedio.

E. SOFT SOLUTIONS: SOFT SOLUTIONS

E.1 CAPACIDADES: SOFT SOLUTIONS tiene la mejor capacidad de indexación. Con un método especial de compactación, asegura que puede hacer sus indices cada vez más pequeños. Los índices se cargan en caché, de manera que los tiempos de respuesta de las búsquedas "Full-Text" son muy pequeños.

SOFT SOLUTIONS tiene las capacidades más completas. Además de búsquedas "Full-Text". hace aproximaciones y búsquedas por sonidos similares.

La importacion de documentos permite tres alternativas que demuestran gran versatilidad: copiar todos los archivos existentes a una nueva localidad. mantenerlos en el directorio corriente, o moverlos a una nueva localidad.

El diseño de los perfiles de los documentos es demasiado estructurado; cuenta con sólo dos campos para definición del usuario.

E.2 TRABAJO EN REDES: SOFT SOLUTIONS es independiente a la red. En red Novell sería de desear que tomara los nombres de los usuarios y asumiera los niveles de seguridad de la red. A pesar de esto, el matrimonio entre el producto y Netware es intachable.

E.3 INSTALACION Y USO: La instalación corre bien, pero requiere muchas revisiones en el manual. El menú principal es poco usual, y poco intuitivo para los usuarios Netware. Todas las alternativas se originan de una pantalla con un perfil de documento vacio.

E.4 DOCUMENTACION: La documentación es decepcionante. El sólo proceso de instalación requiere saltar varias localidades del manual. Existe además una sección de implementación y luego una sección de **Setup**. lo cual complica **la** situación. La sección de Referencia pudo ser mejor si existiera un orden alfabetico y no un orden por funciones. La estructura existente requiere **que** uno lea toda una seccidn o que utilice el indice para encontrar la infox-mación que se necesita.

E.5 DESEMPEÑO: SOFT SOLUTIONS estuvo en la mitad de los cinco productos en cuánto **a la** importación de documentos. Sin embargo, después de la importación tuvo los tamaños más pequeiios de indices con un promedio de 46 carácteres **por** documento. Su competidor mas cercano tuvo 473 caracteres por documento. SOFT SOLUTIONS fue el más rápido en cuanto a **carga** del programa. Toma un promedio de 5 segundos desde el menu hasta el programa.

El hecho de poseer los **indices** más pequeños, puede convertir a SOFT SOLUTIONS en el lider del desempeño. Aún el índice más grande puede cargarse en caché y eliminar la interferencia con el disco. De esta manera, sus tiempos de búsqueda podrían ser aún menores [DUNC91].

2.3.3 CONCLUSIONES.-

Los Laboratorios LAN TIMES decidieron recomendar al Manejador de Documentos SOFT SOLUTIONS [DUNC911.

Quedó claro que la capacidad de indexación "Full-Text" es indispensable para un software Manejador de Documentos y yue es imperdonable que NETADVANTAGE no la provea.

Se concluyó también que una vez encontrado un documento, la entrada a el debe ser lo más rápida posible. SOFT SOLUTIONS. REFERENCEPOINT, y DOCUMENT ADMINISTRATOR tuvieron tiempos aceptables de ingreso a los documentos. Los 8 segundos de NETADVANTAGE y 10 de PC-DOCS fueron inaceptables, aún en una estación 386SX de 16MHz.

La interfase de PC-DOCS fue la mas parecida a Netware, aunque esto no se considera una virtud del todo. La interfase más agradable fue la de DOCUMENT ADMINISTRATOR con su sistema de menus Pull-Down. La interfase de SOFT SOLUTIONS es poco usual, pero una vez acostumbrados, trabaja bien.

Se destaca la importancia de la consistencia y organización de la Documentación de todo Producto Manejador de Documentos.

CAPITULO III

SISTEMA PERSONAL DE MANEJO DE DOCUMENTOS

3.1 ALCANCE DEL SISTEMA.-

El "Sistema Personal de Manejo de Documentos" satisface las siguientes necesidades:

- Almacenamiento de información referente a diversos tipos de Documentos Impresos como son: Articulos de Revistas o Periódicos, Libros, Capítulos de Libros, Tomos de Enciclopedia o Colección, Tesis, Proyectos, Conferencias, Seminarios, etc.

- Consulta, Actualización y Eliminación de los Documentos almacenados.

- Acceso a los Documentos almacenados, mediante Búsquedas "Inteligentes".

- Almacenamiento de información referente a las Fuentes de los Documentos, tales como: Editoriales, Instituciones, Revistas o Periódicos, etc.

- Consulta, Actualización y Eliminación de las Fuentes almacenadas.

- Mantenimiento de Archivos Secundarios con información relacionada a los Documentos almacenados. Existirán Archivos de: Idiomas, Tipos de Documentos y Tópicos.

- Mantenimiento de la Base de Datos mediante tareas de: Creación Original de la Base de Datos, Reindexación e Inicialización de Archivos.

- Generación de Reportes de los Archivos de la Base de Datos.

3.2 DESCRIPCION GENERAL Y COMPONENTES DEL SISTEMA.-

El "Sistema Personal de Manejo de Documentos" (SPMD) está conformado por una serie de Módulos Principales y Submódulos, todos presentados en las Figuras 25. 26. y 27. A su vez, la Figura 28 presenta un esquema simplificado de las capacidades y tareas soportadas por el Sistema.

A continuación procederemos a nombrar los Componentes Básicos y ciertas Definiciones y Conceptos que serán utilizados a lo largo de la Documentación del Sistema.

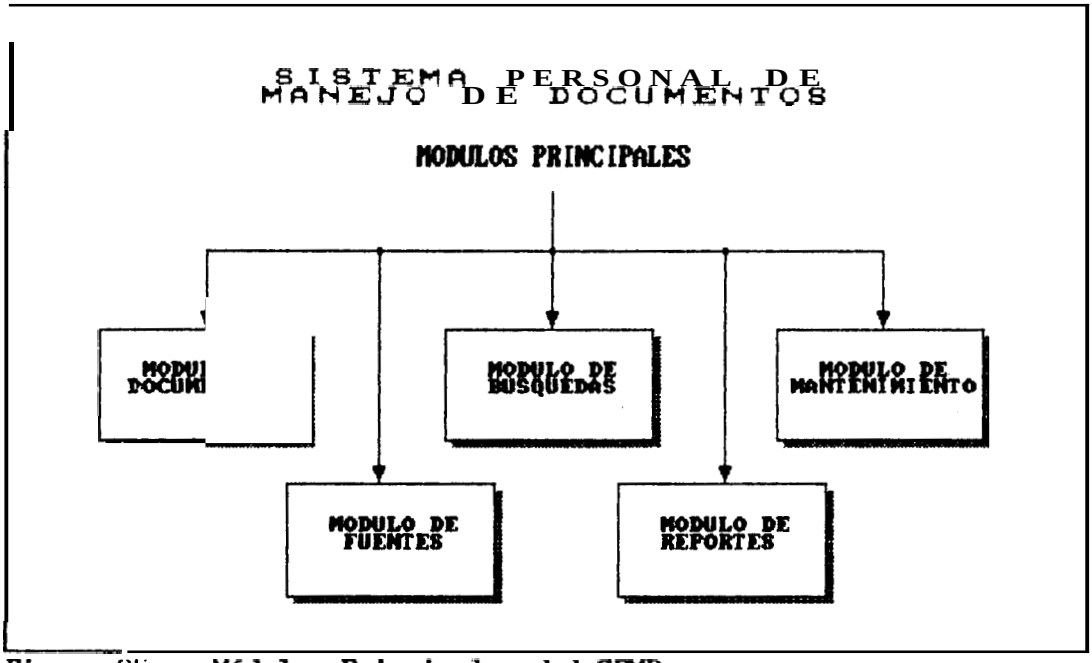


Figura 25. Módulos Principales del SPMD.

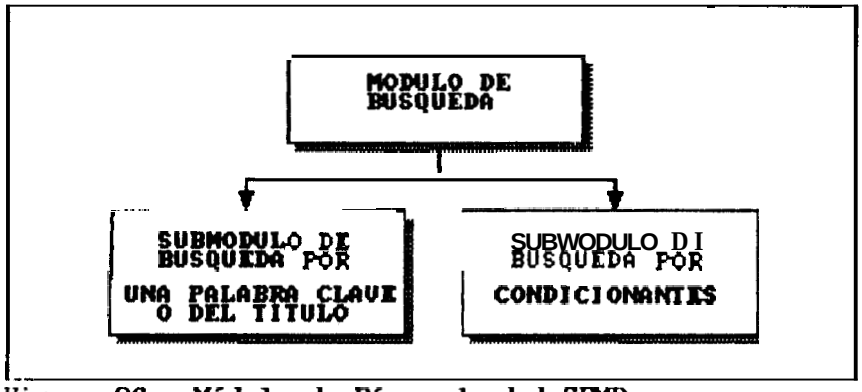


Figura 26. Módulo de Búsqueda del SPMD.

Se llama *DOCUMENTO* a cerdo texto impreso que suministra información. Los Documentos considerados por el "Sistema Personal de Manejo de Documentos" son:

- **Articulcrs** de Revista.
- Artículos de Periódicos.

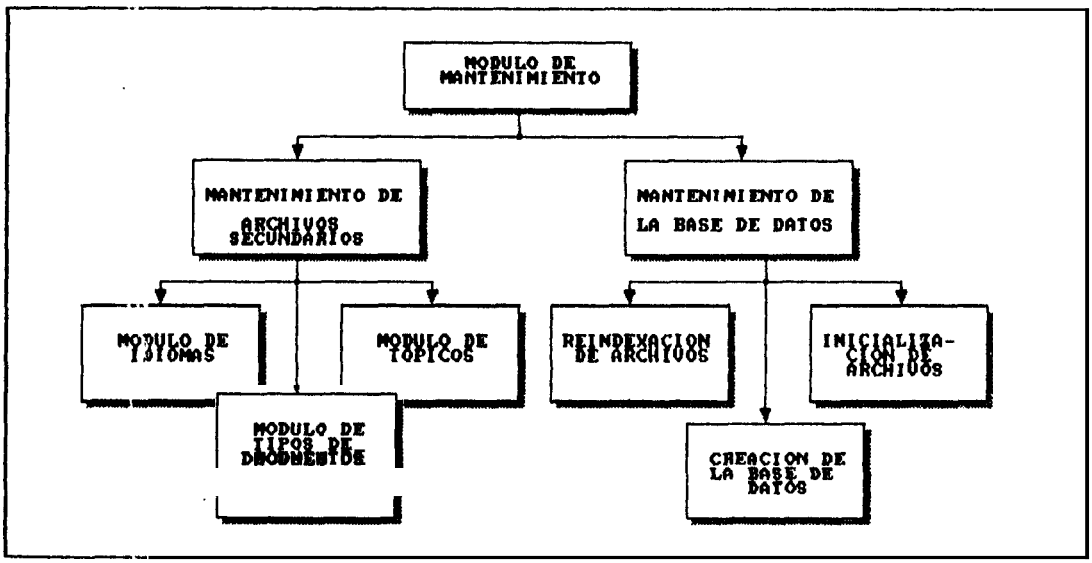


Figura 27. Módulo de Mantenimiento del SPMD.

- Libros.
- **Capitulos** de Libros.
- Tomos de Enciclopedig.
- Tomos de Colección.
- Tesis.
- Proyectos.
- Seminarios.
- Conferencias.

Far:: cada uno de estos Documentos el usuario ingresa al Sistema una serie de datos que representan e identifican al Documento, ej : Código, Titulo, Autor. Fuente. etc. Entre estos datos se distinguen las *"Palabras Claves": palabras o frases que representan el contenido del Documento.*

1. *REGISTRO DE DOCUMENTOS:* *Ingreso, Actualización, Consulta y Eliminación.*

2. *BUSQUEDA DE DOCUMENTOS:*
   2.1. *BUSQUEDA EN BASE A UNA PALABRA CLAVE O DEL TITULO.*
   Opciones:
   - *Ver Respuesta Previa o realizar Búsqueda Nueva.*
   - *Impresión de Resultados.*

   2.2. *BUSQUEDA EN BASE A CONDICIONANTES.*
   Condicionantes:
   - *Una/Dos/Tres Palabras Claves o del Titulo.*
   - *Autor: Se permite el uso del '*' como comodín de fin de cadena.*
   - *Código de la Fuente.*
   - *Código del Idioma.*
   - *Código del Tipo de Documento.*
   - *Mes de Publicación: Uso de Rango >= y <=.*
   - *Año de Publicación: Uso de Rango >= y <=.*

   Opción: - *Impresión de Resultados.*

3. *REGISTRO DE FUENTES:* *Ingreso, Actualización, Consulta y Eliminación.*

4. *GENERACION DE REPORTES:* *Imprimir el contenido de los Archivos de la Base de Datos.*

5. *MANTENIMIENTO.*
   5.1. *REGISTRO DE TIPOS DE DOCUMENTOS:* *Ingreso, Actualización, Consulta y Eliminación.*

   5.2. *REGISTRO DE IDIOMAS:* *Ingreso, Actualización Consulta y Eliminación.*

   5.3. *REGISTRO DE TOPICOS:* *Ingreso, Actualización Consulta y Eliminación.*

   5.4. *CREACION DE LA BASE DE DATOS.*

   5.5. *REINDEXACION DE ARCHIVOS.*

   5.6. *INICIALIZACION DE ARCHIVOS.*

Figura 28. Capacidades del Sistema Personal de Manejo de Documentos.

Estas Palabras Claves, junto con las Palabras que conforman el Título del Documento, son utilizadas por los algoritmos de Búsqueda para lograr la recuperación y el acceso de Documentos requeridos por el usuario en un cierto momento.

La Rase de Datos está compuesta por diversos Archivos de Documentos. Los Documentos se localizan en uno u otro Archivo. en base al Tópico o Materia que tratan. Es asi como pueden existir: Archivos de Documentos de Computación, Electrónica, Medicina, Arte. etc. Existen en la **Base** de Datos tantos Archivos de Documentos como Tópicos hayan sido definidos por el usuario. **Cada** Archivo de Documentos tiene tantos registros como Documentos **hayan** sido ingresados sobre ese Tópico respectivo. A continuación ee presenta el diseño del Registro de los Archivos de Documentos.

TABLA 1. DISEÑO DEL REGISTRO DE DOCUMENTOS

	CAMPO	TIPO	DESCHIPCION
1	IDENT	char[10]	Código o Identificador alfanumérico y único para cada Documento.
2	TIT	char[60]	Titulo del Documento.
3	AUTOR	char[40]	Autor del Documento.
4	C_FUENTE	char[8]	Código de la Fuente del Documento.
5	ANO	char[4]	Año de publicación del Documento.

6	MES	char[2]	Mes de publicación del Documento.
7	IDIOMA	char[21	Código del Idioma de publicscción del Documento.
8	FAGS	char[91	Número de Página Inicial y Final del Documento. [nnnn-mmmm]
9	REF	char[40]	Persona que posee, o lugar donde se puede encontrar el Documento.
10	C_TIPO	char[21	Código del Tipo de Documento: AR = Artículo de Revista/Periódico CL = Capítulo de Libro LE = Libro/Tomo de Enciclopedia PT = Proyecto/Tesis SC = Seminario/Conferencia
11	VOLUMEN	char[4]	Número de Tomo: Se aplica a Revistas, Enciclopedias o Libros compuestos por más de un Tomo.
12	COLECCION	char[30]	Nombre de Colección o Enciclopedia: Se aplica a Libros o Tomos de Colecciones o Enciclopedias.
13	LIBRO	char[30]	Para un Capítulo de Libro, contiene el nombre del Libro al que pertenece.
14	CLAVES	char[2831	Palabras Claves que representan el contenido del Documento.
15	RESUMEN	char[703]	Resumen del Documento.

Como se puede observar, tres datos sobre los Documentos son ingresados mediante códigos: el Tipo de Documento, la Fuente y el Idioma de Publicación. Para cada uno de es-

tos campos existe un Archivo, que almacena respectivamente los Tipos de Documentos. las Fuentes y los Idiomas existentes en la Base. Como lo muestra la Figura 28, el SPMD provee un Módulo de Fuentes y Submódulos de Idiomas y Tipos de Documentos que permiten el ingreso, actualización, eliminación y consulta de los registros de estos archivos.

Existen en el Sistema 5 ***TIPUS DE DOCUMENTOS***, codificados de la siguiente manera:

AR = Artículo de Revista o Periódico

CL = Capitulo de Libro

LE = Libro o Tomo de Enciclopedia

PT = Yroyecto o Tesis

SC = Seminario o Conferencia

TABLA 2. DISEÑO DEL REGISTRO DE TIPOS DE DOCUMENTOS

	CAMPO	TIPO	DESCRIPCION
1	C_TIPO	char[2]	Código alfanumérico y único para cada Tipo de Documento.
2	TIPO	char[25]	Nombre del Tipo de Documento.

De todos los campos que conforman el Registro de Documentos hay algunos que sólo tienen sentido para determinados Tipos de Documentos. La siguiente Tabla describe

qué campos del Registro de Documentos son válidos para cada Tipo de Documento (s=SI , n=NO).

TABLA 3.
TIPOS DE DOCUMENTOS VS CAMPOS DEL REGISTRO DE DOCUMENTOS

		AR	CL	LE	PT	SC
1	Identificador	s	s	s	s	s
2	Título	s	s	s	s	s
3	Autor	s	s	s	s	s
4	Cod. de Fuente	s	s	s	s	s
5	Año	s	s	s	s	s
6	Mes	s	s	s	s	s
7	Cod. del Idioma	s	s	s	s	s
8	Páginas	s	s	s	s	s
9	Referencia	s	s	s	s	s
8	Cod. de Tipo de Documento	s	s	s	s	s
9	Enciclopedia	n	s	s	n	n
10	Libro	n	s	n	n	n
11	Colección	s	s	s	s	s
13	Claves	s	s	s	s	s
14	Resumen	s	s	s	s	s

El Sistema llama ***FUENTES*** a las Editoriales o Instituciones que generan o publican los Documentos. Existe un único Archivo de Fuentes en toda la Base de Datoe. En él se almacenan todas las Fuentes, sin importar el Tópico de los Documentos. Existen una serie de datos que se ingresan

para cada Fuente, ej: **Ciudad**. País, Dirección, etc. En el caso de Articuloe de Revistas o Periódicos, se llama "**Fuente**" al nombre de la Revista o Periódico al que pertenece el artículo y se hace uso del campo llamado "Editorial" para especificar quién edita la Publicación.

TABLA 4. DISEÑO DEL REGISTRO DE FUENTES

	CAMPO	TIPO	DESCRIPCION
1	C_FUENTE	char[8]	Código alfanumérico y único para cada Fuente.
2	FUENTE	char[40]	- Nombre de Revista o Yeriódico: aplicado a Articulos. - Institución: aplicado a Conferencias, Seminarios, Tesis, Proyectos. - Editorial: aplicado a Libros, Tomos de Enciclopedias o Colecciones.
3	EDITORIAL	char[40]	Sólo utilizado para el caso de Articulos de Revistas o Periódicos, contiene el nombre de la Editorial.
3	CIUDAD	char[15]	Ciudad de ubicación de la Fuente.
4	PAIS	char[15]	País de ubicación.
5	DIREC	char[30]	Dirección de la Fuente.
6	POST	char[10]	Número de la Casilla Postal
7	TELEFONO	char[14]	Código de Area + Número.

El Sistema mantiene también un Archivo de *IDIOMAS*. Los registros de este Archivo agrupan los variados idiomas en los que se han publicado Documentos.

TABLA 5. DISERO DEL REGISTRO DE IDIOMAS

	CAMPO	TIPO	DESCRIPCION
1	C_IDIOMA	char[2]	Código del Idioma de Publicación.
2	IDIOMA	char[15]	Nombre del Idioma de Publicación.

Existe un cuarto tipo de Archivo totalmente relacionado con los Archivos de Documentos. Para cada Archivo de Documentos existe un Archivo de *PALABRAS*. La existencia de estos archivos es totalmente transparente para el usuario. Las operaciones de ingreso, actualización, eliminación y consulta sobre los Archivos de Palabras se llevan a cabo implícitamente al ingresar, actualizar, eliminar y consultar los Archivos de Documentos. Los Archivos de Palabras, uno pare cada Tópico, contienen todas las Palabras Claves y del Título de todos los Documentos de ese Tópico. Cada registro tiene dos campos: la Palabra en si. y asociada a ella el Código del Documento que la contiene. Los Archivos de Palabras son básicos para la implementación de los algoritmos de "Busqueda Inteligente" de Documentos.

TABLA 6. DISEÑO DEL REGISTRO DE PALABRAS

	CAMPO	TIPO	DESCRIPCION
1	PALABRA	char[30]	Palabra Clave o del Titulo.
2	IDENT	char[10]	Código del Documento al que la Palabra pertenece.

Si dentro de un mismo Archivo de Palabras existen diversos registros con el mismo **valor** del campo de Palabra, esto significa que esa Palabra se encuentra en varios Documentos del Tópico que determina el Archivo. A su vez, todos los registros, dentro de un mismo Archivo de Palabras, que tengan el mismo valor del campo de Identificador, constituyen todas las **Palabras** del Documento representado por el Identificador.

Se llama ***BUSQUEDA*** al proceso mediante el cual se recuperan Documentos de un determinado Archivo, en base al requerimiento de información que tenga el usuario.

El SPMD permite realizar ***consultas*** del Archivo de Documentos de cierto Tópico, por medio del uso de los Códigos de los Documentoe. Sin embargo, es **poco** probable que el usuario pueda siempre recordar los Códigos de los Documentos **deseados.** Para suplir esta falta, el Sistema implementa además ***BUSQUEDAS POR CONTENIDO.***

Un ***PRIMER SUBMODULO DE BUSQUEDA*** (Figura 26), se basa en el ingreso por parte del usuario de una Palabra Clave o una Palabra del Título, para por medio de ella iniciar una Búsqueda que concluirá con la recuperación de Documentos relacionados directa o indirectamente a esa Palabra.

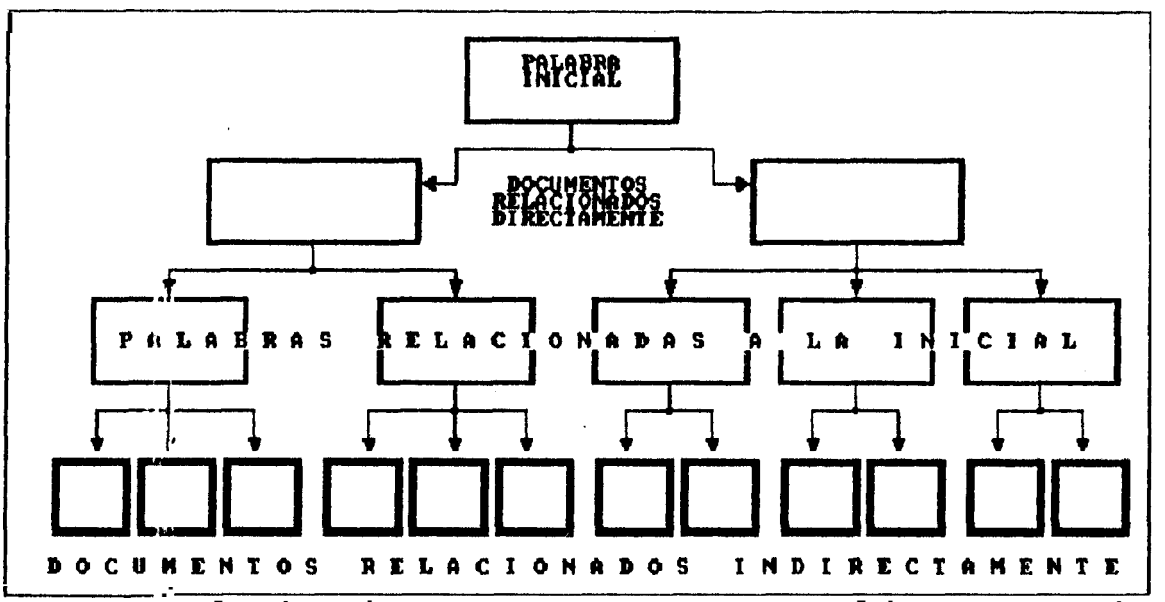


Figura 5. Algoritmo de Búsqueda en Base a una Palabra Clave *o* de Titulo.

El algoritmo de Búsqueda presenta una estructura de árbol, como lo muestra la Figura 29. Se basa en el Sistema de Obtención de Información (AIRS), programa desarrollado por Richard Belew como estudiante de Postgrado en la hi-versidac de Michigan y expuesto en Mayo de 1987 en la

Primera Conferencia Internacional sobre Inteligencia Artificial, donde Belew emitió su informe [HORA88].

La raíz del arbol es la Falabra ingresada por el usuario. El segundo nivel del árbol son los Documentos que tienen esa Falabra como parte de su Titulo o como Palabra Clave. Decimos que estos Documentos están relacionados *directamente* con la Palabra. El tercer nivel del árbol lo conforman todas las Palabras Claves y las Palabras de los Titulos de los Documentos ya recuperados, sin incluir a la Palabra de la raíz del arbol. El último nivel del árbol lo constituyen los Documentos que tienen alguna de las Palabras del tercer nivel como parte de sus Titulos o como Falabras Claves. Asi pues, el cuarto nivel contiene los Documentos *indirectamente* relacionados a la Palabra Inicial.

El proceso de Busqueda por Palabra Clave o del Título da como resultado el conjunto de Documentos del segundo y cuarto nivel del arbol, es decir los Documentos relacionados directa e indirectamente a la Falabra.

Este Submódulo de Busqueda provee también otra opción. Luego de realizada una Busqueda, graba en un archivo los Códigos resultantes de los Documentos asociados a la Pala-

bra ingresada. Así si el usuario vuelve a realizar la misma Búsqueda (en base a la misma Palabra), él podrá decidir si quiere ver la Respuesta Grabada previamente o realizar todo el proceso de búsqueda nuevamente. De esta manera se introduce en la Base de Datos un nuevo tipo de Archivo: el Archivo de *BUSQUEDAS GRABADAS*; existiendo uno por cada Archivo de Documentos, o sea: uno por cada Tópico.

El Archivo de Búsquedas Grabadas funciona como una "Guía de Direcciones". El primer campo lo constituye la Palabra que inicia el proceso de Búsqueda y asociada a ella está el nombre del Archivo (la "dirección") donde están almacenados los Códigos de los Documentos que constituyen la respuesta de la Búsqueda.

TABLA 6. DISEÑO DEL REGISTRO DE BUSQUEDAS GRABADAS

	CAMPO	TIPO	DESCRIPCION
1	PALABRA	char[30]	Palabra que dió inicio al Proceso de Búsqueda.
2	ARCHIVO	char[12]	Nombre del archivo donde están grabados los Códigos que constituyeron la Respuesta de la Búsqueda.

Cada Archivo de Búsquedas Grabadas tiene un máximo de 15 Registros. Esto significa que el Sistema almacena las respuestas a 15 Búsquedas realizadas, y luego éstas empie-

zan a ser reemplazadas por las respuestas de Búsquedas subsiguientes. Para cada Tópico habrá entonces 1 Archivo de Búsquedas Grabadas y hasta 15 Archivos de *RESPUESTAS*. Un Archivo de Respuestas contiene únicamente los Códigos de los Documentos que conforman la respuesta.

TABLA 7. DISEÑO DEL REGISTRO DE RESPUESTAS

	CAMPO	TIPO	DESCRIPCION
1	IDENT	char[10]	Códigos de los Documentos que son Respuesta a una determinada Búsqueda.

En la asignación de los nombres de los Archivos de Respuestas, interviene el Archivo de *TOPICOS*. Como habíamos dicho, el Sistema permite la existencia de varios Topicos. cada uno con un correspondiente Archivo de Documentos. de Palabras y de Busquedas Grabadas. El Archivo de Tópicos almacena como primer campo los nombres de los Tópicos existentes y tendrá tantos registros como Tópicos hayan sido definidos por el usuario.

Además del Nombre del Tópico, el registro de Tópicos tiene dos campos más. A cada Tópico se le asigna un Código constituido por una sóla letra. Esta letra es el primer caracter del nombre de todos los Archivos de Respuesta de ese Tópico. Las siguientes letras son un número secuencial

desde 0 hasta 14, pues como ya dijimos se pueden grabar un maximo de 15 Búsquedas. Si el Código de un Tópico es la letra "C", sus Archivos de Respuestas se llamarán: C0.TXT, C1.TXT, C14.TXT, etc.

El tercer campo del registro de Tópicos almacena el número de búsquedas que han sido grabadas, y coincide siempre con el numero secuencial a utilizar en la construcción del nombre del siguiente Archivo de Respuestas de ese Tópico. Siempre que se graba una Búsqueda, se aumenta un registro en el Archivo de Búsquedas Grabadas, se crea un nuevo Archivo de Respuestas y se incrementa el Número de Búsquedas Grabadas dentro del Registro de Tópico.

TABLA 8. DISEÑO DEL REGISTRO DE TOPICOS

	CAMPO	TIPO	DESCRIPCION
1	TOPICO	char[15]	Nombre del Tópico.
2	C_TOPICO	char[1]	Código único para cada Tópico: Letra a ser utilizada para la construcción de los nombres de los Archivos de Respuestas del Tópico.
3	NUMERO	char[3]	Número de Búsquedas Grabadas. Al grabar una nueva Búsqueda se incrementa este número. Sirve para poner nombre a un nuevo Archivo de Respuestas.

El *SEGUNDO SUBMODULO DE BUSQUEDA* (Figura 26) permite el ingreso opcional de **una** serie de Condicionantes o Restricciones. Se recuperan los Documentos que cumplen con todas las condiciones planteadas. Si un campo de condición permanece vacío, no existe restricción en él. El Sistema implementa los siguientes Condicionantes:

- **Una, Dos *o* Tres Palabras Claves *o* del Título, relacionadas entre sí por la operación lógica ´O´:** Cumplen con esta Condición todos los Documentos relacionados directa o indirectamente **con *por lo menos una*** de las Palabras ingresadas. Se utiliza para cada Palabra el mismo algoritmo de búsqueda explicado previamente y se construyen tantos árboles como Palabras hayan sido ingresadas. Son escogidos todos los Documentos de los segundos y cuartos niveles de los árboles.

- **Autor:** Se permite ingresar un patrón de Autor incompleto, usando como comodín un ***"*" al final de la cadena.*** Por ej: al ingresar "STE*", cumplirán esta condición todos los Documentos cuyo nombre de Autor empiece con las letras "STE". Si no se coloca un "*" al final del nombre de Autor, se asume que éste está

completo y se buscan los Documentos que cumplen exactamente con el nombre de Autor ingresado.

- **Código de Fuente:** Cumplen esta condición los Documentos cuya Fuente **corresponde** con el Código ingresado por el usuario.

- **Código de Idioma:** Cumplen esta condición los Documentos cuyo Idioma de Publicación corresponde con el Código ingresado por el usuario.

- **Código de Tipo de Documento:** Cumplen esta condición los Documentos que pertenecen al Tipo, cuyo Código he ingresado el usuario.

- **Periodo de Publicación:** Se permite ingresar un rango **en** meses y años, cumpliendo esta condición los Documentos publicados durnnte ese periodo de tiempo.

Como podemos notar, **el** Segundo Submódulo de Búsqueda incluye al Primero y permite añadir restricciones al conjunto de Documentos **que** se quieren recuperar.

De esta manera hemos presentado una visión general del "Sistema Personal de Manejo de Documentos". destacándose los conceptos de: Documento, Fuente, Tipo de Documento, Idioma, Tópico, Rusqueda por Palabra Clave o del Título y Búsqueda por Condicionantes.

3.2.1 ARCHIVOS DE LA MSE DE DATOS.-

La siguiente Tabla resume el conjunto de Archivos que conforman la Base de Datos Típica del "Sistema Personal de Manejo de Documentos".

BIBLIOTECA

TABLA 10. ARCHIVOS DE LA BASE DE DATOS

ARCHIVOS	CANTIDAD	INDICES
De Documentos	1 para cada TOPICO	- Identificador - Autor - Fuente
De Palabras (Palabras Claves o del Título)	1 para cada TOPICO	- Combinación Palabra/Identificador - Palabra - Identificador
De Tipos de Documentos	1	- Código de Tipo de Documento
De Fuentes	1	- Código de Fuente
De Idiomas	1	- Código de Idioma
De Tópicos	1	- Nombre del Tópico
De Búsquedas Grabndas	1 para cada TOPICO	
De Reepuestas	0-15 por cada TOPICO	

Como se puede apreciar al analizar el diseño de los Registros de cada Archivo de la Base de Datos, existen campos comunes entre los Archivos. Estos campos comunes permiten relacionar un Archivo con otro, como nos lo muestra la Figura 30.

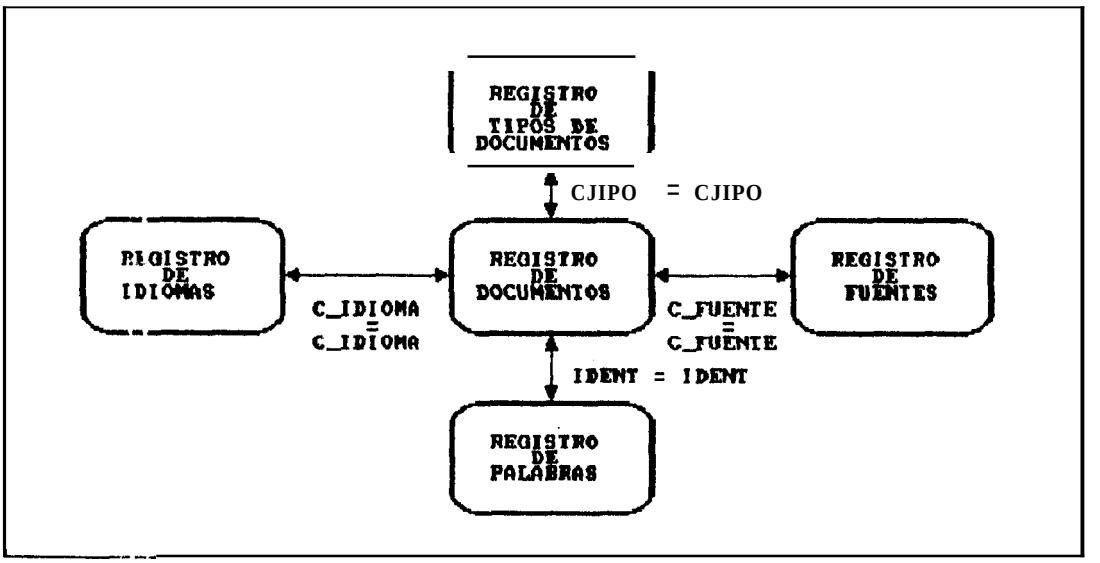


Figura 30. Relaciones entre los Archivos de la Rase de Datos.

Por último analicemos brevemente los nombres que llevan los Archivos de la Base de Datos. Para los Archivos Unicos:

- Archivo de Tipos de Documentos: TIPOS.DAT
- Archivo de Fuentes: FUENTES.DAT
- Archivo de Idiomas: IDIOMAS.DAT
- Archivo de Tópicos: TOPICOS.DAT

Los Archivos de Documentos. Palabras y Búsquedas Grabadas obtienen su nombre tomando las 4 primeras letras del nombre del Tópico del cual tratan. Para el Tópico "COMPUTACION":

- Archivo de Documentos: COMP.DAT
- Archivo de Palabras: COMP_PAL.DAT
- Archivo de Búsquedas Grabadas: COMP.BQ1

Como ya vimos, los Archivos de Respuestas obtienen su nombre tomando el Código del Tópico al que corresponden más un número secuencial. Si el Código del Tópico "COMPUTACION" es la letra "C":

- Archivos de Respuestas: C0.TXT C1.TXT
  ... C14.TXT

3.3 SISTEMA MANEJADOR DE LA BASE DE DATOS.-

Para la implementación del "Sistema Personal para Manejo de Documentos" (SPMD), utilizamos como Sistema Manejador de Base de Datos a "CDATA" (The Cheap Data Base Management System), desarrollado por Al Stevens [STEV87].

3.3.1 LENGUAJE DE DEFINICION DE DATOS DE CDATA.-

El Lenguaje de Definición de Datos de CDATA incluye tres partes, implementadas en la sintaxis del Lenguaje C y ubicadas en los archivos *CBS.C, CBS.C1, CBS.C2 y CBS.C3*:

A. Diccionario de Elementos de Datos.

B. Especificación de Archivos.

C. Especificación de Indices.

A. DICCIONARIO DE ELEMENTOS DB DATOS DE CDATA.-

Todos los Elementos de Datos nombrados en la siguiente Tabla son manejados por CDATA y están directamente relacionados a los conceptos de: Documentos, Tipos de Documentos, Fuentes, Idiomas, Tópicos y Palabras. Los Elementos de Datos que no forman parte de este Diccionario son manejados directamente por la Aplicación.

TABLA 11. DICCIONARIO DE ELEMENTOS DE DATOS

NOMBRE DEL ELEMENTO	TIPO DE DATO	LONGITUD
IDENT	alfanumérico	10
TIT	alfanumérico	60
AUTOR	alfanumérico	40
C_FUENTE	alfanumérico	8
ANO	numérico	4
MES	numérico	2
C_IDIOMA	alfanumérico	2
PAGS	alfanumérico	9
REF	alfanumérico	40
C_TIPO	alfanumérico	2
VOL	numérico	4
COLEC	alfanumérico	30

LIBRO	alfanumérico	30
CLAVES	alfanumérico	283
RESUMEN	alfanumérico	703
TIPO	alfanumérico	25
FUENTE	alfanumérico	40
EDITORIAL	alfanumérico	40
CIUDAD	alfanumérico	15
PAIS	alfanumérico	15
DIREC	alfanumérico	30
POST	alfanumérico	10
TELEFONO	alfanumérico	14
IDIOMA	alfanumérico	15
TOPICO	alfanumérico	15
C_TOPICO	alfanumérico	1
NUMERO	numérico	3
PALABRA	alfanumérico	30

Los Elementos del Diccionario de Datos deben ser descritos en un formato que los programas de aplicación puedan entender. En primer lugar, se establece una Tabla de Definiciones que otorga a cada Elemento de Dato un nemónico y le da además una equivalencia entera (Figura 31).

Para representar la longitud y el tipo de cada Elemento de Dato, se utilizan el arreglo de enteros de la Figura 32 y el arreglo de caracteres de la Figura 33. La posición dentro de cada arreglo indica a qué Elemento de Dato perte-

nece la longitud o el tipo. Ej: la primera longitud del arreglo *ellen* es para el primer Elemento de Dato de la Figura 31. Los códigos de tipos de datos utilizados por la aplicación son: 'A' (alfanumérico) y 'Z' (numérico).

```
#define IDENT 1
#define TIT 2
#define AUTOR 3
#define C_FUENTE 4
#define ANO 5
#define MES 6
#define C_IDIOMA 7
#define PAGS 8
#define REF 9
#define C_TIPO 10
#define VOL 11
#define COLEC 12
#define LIBRO 13
#define CLAVES 14
#define RESUMEN 15
#define TIPO 16
#define FCJENTE 17
#define EDITORIAL 18
#define CIUDAD 19
#define PAIS 20
#define DIREC 21
#define POST 22
#define TELEFONO 23
#define IDIOMA 24
#define TOPICO 25
#define C_TOPICO 26
#define NUMERO 27
#define PALABRA 28
```

igura 31. CBS.C1: Definición del Diccionario de Datos.

```
int ellen [] = {
        10, 60, 40, 8, 4, 2, 2, 9,  40, 2, 4, 30, 30, 28
        703, 25,
        40, 40, 15, 15, 30, 10, 14, 15, 15, 1, 3, 30
};
```

Figura 32. CBS.C3: Longitudes de los Elementos de Datos

```
char eltype [] = "AAAAZZAAAAZAAAAAAAAAAAAAAZA";
```

Figura 33. CBS.C2: Códigos de Tipos de Datos de los Elementos del Diccionario.

B. ESPECIFICACION DE ARCHIVOS.-

La Especificación de Archivos de CDATA envuelve tres pasos: primero, se dan nombres a los archivos: segundo, se determinan los Elementos de Datos que están en cada archivo: y tercero, se describe la estructura del registro de cada archivo (Figuras 34, 35 y 36) [STEV87].

```
#define ARCH        0
#define ARCH_PAL    1
#define TIPOS       2
#define FUENTES     3
#define IDIOMAS     4
#define TOPICOS     5

char *dbfiles [] = {
        "ARCH",
        "ARCH_PAL",
        "TIPOS",
        "FUENTES",
        "IDIOMAS",
        "TOPICOS",
        (char *) 0
};
```

Figura 34. CBS.C2 y CBS.C3: Nombres de Archivos.

```
int docs-file [] = {
        IDENT.
        TIT.
        AUTOR.
        C_FUENTE.
        ANO.
        MES.
        C_IDIOMA.
        PAGS.
        REF.
        C_TIPO.
        VOL.
        COLEC.
        LIBRO.
        CLAVES.
        RESUMEN.
        0
};

int tipos_file [] = {
        C_TIPO.
        TIPO.
        0
};

int idiomas_file [] ={
        C_IDIOMA.
        IDIOMA.
        0
};

int topicos_file [] ={
        TOPICO.
        C_TOPICO.
        NUMERO.
        0
};

int fuentes_file [] = {
        C_FUENTE.
        FUENTE.
        EDITORIAL.
        CIUDAD.
        PAIS.
        DIREC.
        POST.
        TELEFONO.
        0
};

int palabras_file [] = {
        PALABRA.
        IDENT.
        0
};

int *file_ele [] = {
        docs_file.
        palabras_file.
        tipos_file.
        fuentes_file.
        idiomas_file.
        topicos_file.
        (int *) 0
};
```

igura 35. CBS.C3: Contenido de los Archivos.

En la Figura 34, todos los archivos de Documentos estan representados por el nombre "ARCH", y todos los archivos de Palabras por "ARCH-PAL". Los Archivos de Búsquedas Grabadas y de Respuestas no son manejados por CDATÁ. sino directamente por los programas de aplicación.

```
struct docs {
    char ident [11];
    char tit [61];
    char autor [41];
    char c_fuente [9];
    char ano [5];
    char mes [3];
    char c_idioma [3];
    char pags [10];
    char ref [41];
    char c_tipo [3];
    char vol [5];
    char colec [31];
    char libro [31];
    char claves [284];
    char resumen [704];
};

struct tipos {
     char c_tipo [3];
     char tipo [26];
};

struct fuentes {
     char c_fuente [9];
     char fuente [41];
     char editorial [41];
     char ciudad [16];
     char pais [16];
     char direc [31];
     char post [11];
     char telefono [15];
};

struct idiomas {
     char c_idioma [3];
     char idioma [16];
};

struct topicos {
     char topico [16];
     char c_topico [2];
     char numero [4];
};

struct palabras {
     char palabra [31];
     char ident [11];
};
```

igura 36 **CBS.C1: Estructuras de Registro de los Archivos**

C. ESPECIFICACION DE INDICES,-

CDATA determina los indices de cada archivo y qué Elementos de Datos son las claves de esos indices. Las claves primarias (únicas) de cada archivo son:

DOCUMENTOS:	*IDENT*
TIPOS:	*C_TIPO*
FUENTES:	*C_FUENTE*
IDIOMAS:	*C_IDIOMA*
TOPICOS:	*TOPICO*
PALABRAS:	*PALABRA, IDENT*

CDATA permite que un archivo tenga varios **indices.** De ser este el caso, sólo uno de los indices se basará en una clave primaria. CDATA se asegura de que los datos de este indice no se repitan. Los restantes indices tendrán claves secundarias, y en ellos sí podrán existir valores repetidos. Una clave de un índice, primaria o secundaria, puede consistir de la concatenación de varios Elementos de Datos.

Para definir los índices de los archivos se construyen una serie de arreglos de enteros. Cada arreglo de enteros contiene los valores enteros de los campos que forman la clave de un indice. Si una clave está formada por varios Elementos concatenados, su arreglo tendrá varios enteros. Si un archivo tiene varios índices, se deberán construir pal-a ese archivo, un arreglo de enteros por cada índice.

Una vez creados los arreglos de enteros de todos los índices de la Base de Datos, se construye para cada archivo un arreglo que apunte a sus correspondientes arreglos de índices. El primer arreglo de índice que se coloque en cada arreglo de punteros, corresponde al indice primario, de ese archivo. La Base de Datos tiene finalmente un **arreglo** llamado ***index_ele***, que apunta a los arreglos de punteros de los archivos.

```
int x1_docs [] = {
     IDENT,     0
};

int x2_docs [] = {
     AUTOR,     0
};

int x3_docs [] = {
     C_FUENTE, 0
};

int *x_docs [] = {
     x1_docs,
     x2_docs,
     x3_docs,
     (int *) 0
};

int x1_tipos [] = {
     C_TIPO,    0
};

int *x_tipos [] = {
     x1_tipos,
     (int *) 0
};

int x1_fuentes [] = {
     C_FUENTE, 0
};

int *x_fuentes [] = {
     x1_fuentes,
     (int *) 0
};

int x1_idiomas [] = {
     C_IDIOMA, 0
};

int *x_idiomas [] = {
     x1_idiomas,
     (int *) 0
};

int x1_topicos [] = {
     TOPICO,    0
};

int *x_topicos [] = {
     x1_topicos,
     (int *) 0
};

int x1_palabras [] = {
     PALABRA,
     IDENT,
     0
};

int x2_palabras [] = {
     PALABRA,   0
};

int x3_palabras [] = {
     IDENT,     0
};

int *x_palabras [] = {
     x1_palabras,
     x2_palabras,
     x3_palabras,
     (int *) 0
};

int **index_ele [] = {
     x_docs,
     x_palabras,
     x_tipos,
     x_fuentes,
     x_idiomas,
     x_topicos,
     (int **) 0
};
```

Figura 37. CBS.C3: Especificación de Indices.

3.3.2 AFCHIVO CDATA.H .-

Este archivo contiene definiciones **globales**. útiles para todos los programas que conforman el "Sistema Personal de Manejo de Documentos". CDATA define **la** longitud del Nodo de los Indices con el comando ***#define NODE.*** Establece además el tipo de dirección de los archivos como un entero ***long,*** mediante el comando ***#define RPTR.*** De esta manera se permiten direcciones de 32 bits, significando que los archivos pueden tener hasta 2^{32} registros. Los valores globales ***MAXFILS, MAXELE y MAXINDEX*** definen el número máximo de archivos en la Rase de Datos, el número máximo de Elementos de Datos en un archivo y el número máximo de **Indices** por archivo. Los símbolos ***MAXKEYLEN y MAXPAL,*** establecen la longitud máxima de las claves de los indices, y de las palabras claves o del título de los documentos.

```
#define ERROR       -1
#define OK          0
#define TRUE        1
#define FALSE       0
#define NODE        512
#define RPTR        long
#define MXFILS      10
#define MXELE       100
#define MXINDEX     3
#define MXKEYLEN    50
#define MXPAL       30
```

Figura 38. CDATA.H: Algunas Definiciones.

En CDATA se definen también los códigos de **error para** las funciones del Manejador de la Rase de Datos y de los Programas de Aplicación (Figura 39).

```
#define D_NF      1  // El Registro no existe.
#define D_PRIOR   2  // No existe un registro corriente
#define D_EOF     3  // Fin de Archivo.
#define D_BOF     4  // Inicio de Archivo.
#define D_DUPL    5  // El Registro ya existe.
#define D_OM      6  // La Memoria no es suficiente
#define D_INDXC   7  // Indice corrupto
#define D_IOERR   8  // Error de Entrada/Salida
#define D_DIC     9  // No existe el Diccionario
```

Figura 39. CDATA.H: Códigos de Error.

```
typedef struct fhdr {
   RPTR first_record;
   RPTR next_record;
   int record_length;
} FHEADER;

struct soluc1 {
   char palabra
        [MXPAL+1];
   char archi[13];
};

struct soluc2 {
   char palabra1 [MXPAL+1];
   char palabra2 [MXPAL+1];
   char palabra3 [MXPAL+1];
   char autor [41];
   char c_fuente [9];
   char c_idioma [3];
   char c_tipo [3];
   char ano1 [5];
   char mes1 [3];
   char ano2 [5];
   char mes2 [3];
};
```

Figura 40. CDATA.H: Definición de Estructuras.

La Figura 40 muestra las siguientes estructuras:

- El Yeader de los Archivos de **la** Rase de Datos.

- El Registro del Archivo de Búsquedas Grabadas. Como ya dijimos, este archivo no es incluído en el Lenguaje de Definición de Datos de CDATA. Los programas trabaja directamente con él, sin utilizar las funciones suministradas por el Manejador de Base de Datos.
- El Registro utilizado para la Búsqueda por Condicionantes.

3.3.3 FORMATO DE LOS ARCHIVOS DE CDATA.-

A. ARCHIVOS DE DATOS.-

A cada archivo descrito en el Lenguaje de Definición de la Base de Datos le corresponde un archivo en el disco. El nombre de este archivo se forma tomando las primeras 8 letras de las cadenas del arreglo ***dbfiles*** y añadiendo la extension ***.DAT***. En nuestro caso tendremos: TIPOS.DAT, FUENTES.DAT, IDIOMAS.DAT, TOPICOS.DAT, etc.

Un archivo consiste de un Header seguido por una serie de registros de tamaño fijo. El Header (Figura 40) tiene la misma longitud y formato para todos los archivos, pero los registros de datos dependen de la estructura de registro de cada archivo. Ya que CDATA.H definió a RPTR como un long, la longitud del Header será de 10 caracteres.

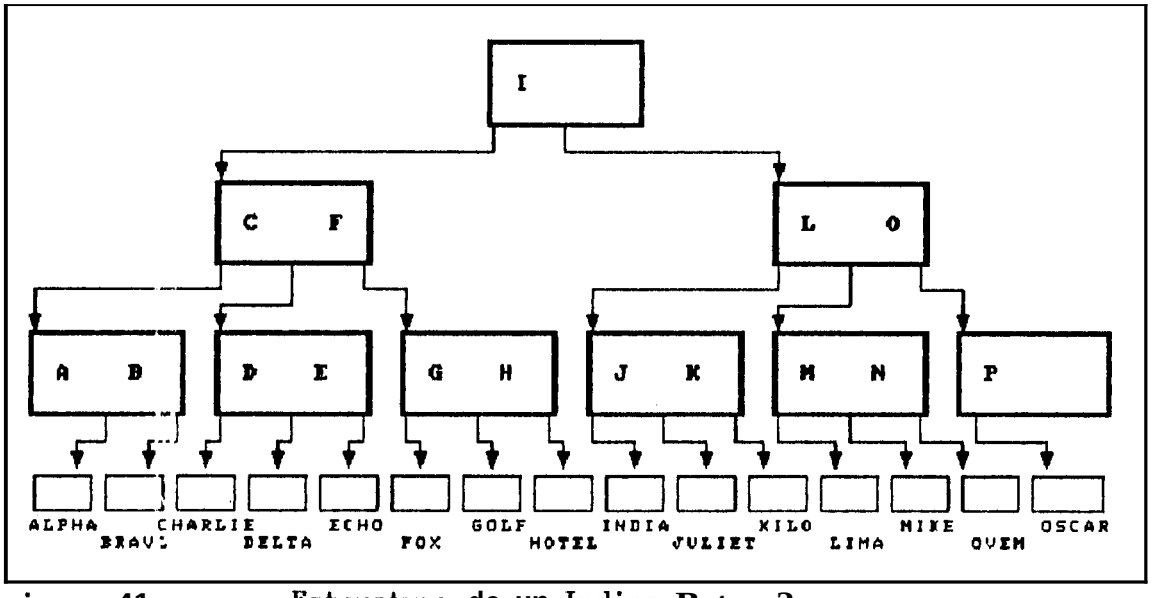
Las variables del HEADER ayudan a controlar el ***espacio de eliminaciones reusables:*** al eliminar un registro, éste es marcado con una bandera y añadido a una lista encadenada de registros reusables. Este espacio reusable elimina la necesidad de empaquetamiento o compresión de los archivos de datos para limpiarlos de los registros eliminados.

Un registro de datos de un archivo es igual en longitud a la suma de las longitudes de los Elementos de Datos que lo conforman, más un caracter por cada Elemento de Datos, debido al fin de cadena.

CDATA graba nuevos registros en el primer espacio disponible de la lista de registros eliminados o al final del archivo. Inicialmente, el archivo contiene únicamente el Header.

B. ARCHIVOS DE INDICES.-

Los indices de los archivos de CDATA se implementan mediante árboles B-tree. La Figura 41 es un ejemplo de un R-tree que utiliza letras del alfabeto como claves para localizar palabras. Las letras son análogas a los valores de las claves de un archivo, y las palabras son similares a los registros de un archivo indexado con B-tree..



igura 41. Estructura de un Indice B-tree?.

Asi pues, CDATA construye y mantiene indices R-tree, uno para cada clave del Lenguaje de Definición de Datos. Habrá índices con claves primarias e índices con claves secundarias. El nombre de los archivos de indices es el mismo que el del archivo de datos. La extensión de construye de manera que será .X01 para el primer índice de cada archivo, .X02 para el segundo y así sucesivamente. Los archivos de índices de clave primaria tendrán siempre la extensión .X01.

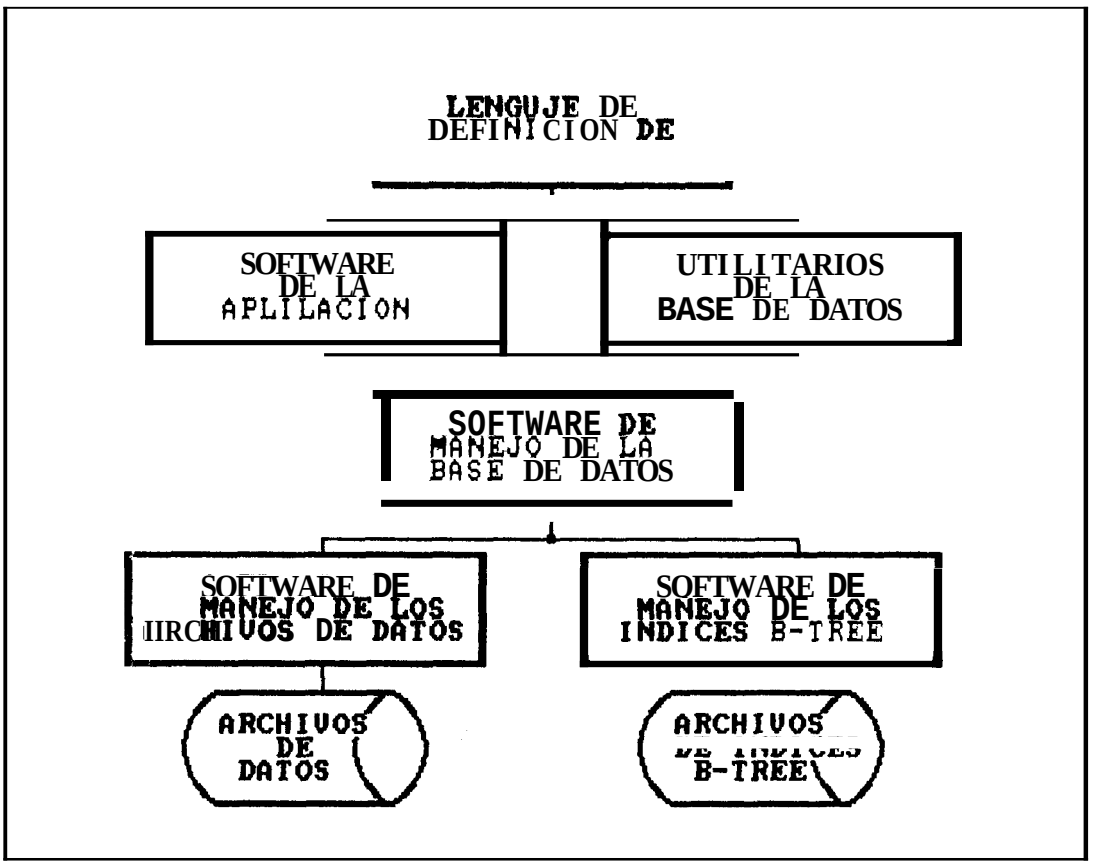
El mantenimiento de los indices B-tree es transparente para el software de aplicación. Cada archivo de índice contiene un Header, seguido de una serie de nodos. El

símbolo global ***NODE*** definido en CDATA.H da la longitud del Header y de los **nodos.** EL Header contiene el tamaño de la clave, la dirección del nodo raiz, el número de claves que un nodo puede contener, un puntero al siguiente nodo dónde se puede realizar un ingreso, y un puntero al último nodo eliminado del árbol. A su vez, cada nodo contiene una bandera que identifica al nodo como hoja o no, punteros al padre y a los hijos derecho e izquierdo, puntero al archivo y los valores de las claves.

3.3.4 ARQUITECTURA DEL SISTEMA CDATA.-

Hemos visto que CDATA maneja una Base de Datos, consistente de Archivos de Datos e Indices B-tree. La siguiente Figura presenta **los** niveles **de** software que se combinan para soportar esta Base.

En el nivel inferior de la Figura están los Archivos de Datos y de Indices. Los Archivos de Datos son manejados por un conjunto de funciones especificas que constituyen el ***Software para Manejo de Archivos de Datos.*** Los Indices son manejados por un conjunto diferente de funciones, llamado ***Software para Manejo de Indices B-tree.*** Los dos conjuntos de funciones no están relacionados y los archivos mismos están desconectados entre sí. Todas estas funciones no son



igura 42. Arquitectura del Sistema Manejador de Bases de Datos CDATA.

vistas por los programas de aplicación del "Sistema Personal de Manejo de Documentos".

Arriba del software de los Archivos de Datos y de Indices está una librería de funciones cuyo propósito es manejar la Base de Datos. Las funciones de manejo de la Base de Datos llaman directamente a las funciones que manejan los Archivos de Datos y a la funciones que manejan los Indices. De esta manera mantienen la relación lógica

que existe [illegible] e los muchos arc[illegible]os que constituyen la Base de [illegible] Este conjunto d[illegible] funciones representa el *lenguaje* [illegible] *pulación de Datos* *CDATA*.

En el software de manejo de la Base de Datos se encuentran dos conjuntos de programas: el ***Software de nuestra Aplicación*** y ciertos programas ***Utilitarios provistos por CDATA*** Todos estos programas accesan la Base de Datos mediante el lenguaje de alto nivel de Manipulación de Datos.

Por encima de todos los niveles está el ***lenguaje de Definición de Datos*** que describe la Base de Datos que CDATA está manejando.

[illegible] DESCRIPCION E IMPLEMENTACION DE MODULOS.

El "Sistema Personal de Manejo de Documentos" está compuesto p[illegible] tres programas ejecutables: ***SPMD.exe***, ***BUS1*** y ***BUS2.exe***. De estos tres programas ***SPMD.exe*** es el [illegible] y es el que llama a los d[illegible] últimos. ***BUS1.exe*** implementa el Submódulo de Búsqueda por Palabra Clave o del [illegible]rio y ***BUS2.exe*** el Submódulo de Búsqueda por Condicio[illegible] ***SPMD.exe*** se encarga de mantener la estructura

general del Sistema y de llevar a cabo todas las opciones restantes.

Cada programa ejecutable se escribió en Lenguaje C, bajo el Ambiente de Programación de Borland C++. Se diseñaron tres Proyectos (Projects), uno para cada ejecutable. Cada Proyecto está formado por una serie de Archivos Fuente con extensión ".C" o ".H" . Existen sin embargo Archivos Fuentes comunes a los tres Proyectos. La Figura 22 presenta el conjunto de Archivos Fuentes que conforman cada Proyecto.

Para el ejecutable ***BUS1.exe*** se utilizó en la Compilación el Modelo de Memoria MEDIUM y para ***BUS2.exe y SPMD.exe*** el Modelo de Memoria LARGE.

A continuación haremos un recorrido a través de los Módulos que conforman las opciones de los Menús del "Sistema Personal de Manejo de Documentos". Para cada opción explicaremos sus capacidades y las funciones de Lenguaje C que ayudaron a su implementación.

SISTEMA PERSONAL DE MANEJO DE DOCUMENTOS

Proyecto: SPMD.EXE

CDATA.H
KEYS.H
CBS.C
DATABASE.C
DATAFILE.C
BTREE.C
PANTALLA.C
FUN_ING.C
D1.C
DATOS.C
SCREEN.C
FUN_SHOW.C
DOC_ING.C
DOC_MOD.C
DOC_ELI.C
PAL_FUN.C
RESUMEN.C
TOPICO.C
IMP.C
IMP_FUN.C
IMP_SHOW.C
DS.C
CREACION.C
INDEXAR.C

Proyecto: BUS1.EXE

CDATA.H
CBS.C
DATABASE.C
DATAFILE.C
BTREE.C
PANTALLA.C
FUN_ING.C
BSQ1.C
B_CONS.C
B_PALS.C
B_IMP1.C
B_NUEVA.C

Proyecto: BUS2.EXE

CDATA.H
CBS.C
DATABASE.C
DATAFILE.C
BTREE.C
PANTALLA.C
FUN_ING.C
DS.C
BSQ2.C
BSQ2B.C
BSQ2C.C
B_CONS.C
B_PALS.C
B_IMP2.C

Figura 43. Archivos Fuentes que conforman cnda Proyecto del SPMD.

A, MODULO DE FUENTES (MENU PRINCIPAL) Y MODULOS DE TIPOS DE DOCUMENTOS, IDIOMAS y TOPICOS (OPCION 'ACTUALIZAR CODIGOS' DEL MENU DE MANTENIMIENTO).-

Estos Módulos permiten al usuario el Ingreso, Actualización, Consulta y Eliminación de los Registros de los Archivos de: Fuentes (FUENTES.DAT), Tipos de Documentos (TIPOS.DAT), Idiomas (IDIOMAS.DAT) y Tópicos (TOPICOS.DAT).

El algoritmo utilizado para la implementación de estos cuatro Modulos representa el proceso universal para Entrada y Recuperación de Datos. Los Archivos *DATOS.C y SCREEN.C* del "Sistema Personal de Manejo de Documentos", constituyen una variación del Utilitario de CDATA "QD: Data Entry and Query" [STEV87].

El usuario ingresa la Clave Principal del Registro, que dependiendo del Módulo será: el Código de la Fuente, el Código del Tipo de Documento, el Código del Idioma o el Nombre del Tópico.

Si la Clave existe en el respectivo Archivo, el Sistema recupera el Registro y lo presenta en pantalla. El usuario puede movilizarse a través de los restantes campos

del registro y modificar los valores de los Elementos de Datos. Cuando uno se encuentra satisfecho con el contenido del registro, se presiona la tecla <F2> para Grabar el registro en el Archivo. Si se quiere rechazar los cambios efectuados sobre un registro, se presiona la tecla <ESC>.

Si se ingresa una Clave que no existe, el Sistema anuncia "Registro Nuevo". Si el usuario ha cometido un error y no quiere ingresar un nuevo registro, presionando <ESC> se rechaza el ingreso. Si si se desea añadir un registro, se puede proceder a llenar los valores de los campos de Elementos de datos. Se presiona <F2> cuando los valores son los deseados.

Para eliminar un registro, se ingresa el valor de la Clave, el registro se presentará en pantalla y se presiona la tecla <F7>. El Sistema confirma si realmente se desea realizar una eliminación.

Se pueden presentar en pantalla el Registro Siguiente y el Registro Anterior al corriente, usando las teclas <PgUp> y <PgDown>. Las teclas <HOME> y <END> presentan siempre el Primer y Ultimo Registro respectivamente.

Las teclas de Flechas y las teclas <ENTER> y <TAB> permiten mover el cursor a través de la pantalla. Con la tecla <INS> se puede pasar al Modo de Inserción, típico de los editores de textos. Presionando nuevamente <INS> se regresa al Modo de Sobreescritura. La tecla <DEL> elimina el caracter sobre el cursor y la tecla <BackSpace> elimina el caracter a la izquierda del cursor.

Para abandonar estos Módulos se presiona <ESC>, cuando la pantalla está vacía.

Cabe decir que en la caso del Módulo de Tópicos, ingresar y eliminar registros tiene un especial significado. Cuando se ingresa un Tópico al Registro de Tópicos, se están creando para ese Tópico: un nuevo Archivo de Documentos, con sus Indices; un nuevo Archivo de Palabras, con sus Indices y un nuevo Archivo de Búsquedas Grabadas. Como ya vimos, en la construcción de los nombres de estos nuevos archivos intervienen las primeras 4 letras del Nombre del Tópico.

A su vez, eliminar un Registro de Tópicos significa eliminar de la Base de Datos: el Archivo de Documentos, con sus Indices; el Archivo de Palabras, con sus Indices;

el Archivo de Búsquedas Grabadas y los Archivos de Respuestas, todos pertenecientes al Tópico cuyo registro se quiere eliminar. El Sistema advierte al usuario que, de eliminar el Registro Tópicos, todos los Archivos nombrados y sus contenidos desaparecerán, pidiendole que confirme la orden.

Para el Módulo de Tópicos, el campo de "Número de Búsquedas Grabadas" es sólo de lectura. El usuario no puede alterar su valor. El Sistema no permite asignar Códigos repetidos a los Tópicos. Una vez asignado un Código a un Tópico, el Sistema pide al usuario no modificarlo.

Cuando el usuario escoge la opción de ***FUENTES*** en el Menú Principal, o las opciones de ***TIPOS DE DOCUMENTOS, IDIOMAS Y TOPICOS***, en el Menú de Mantenimiento, el Programa Principal llama a:

FUNCION DATOS: ***void datos (char *s)***

La cadena **s** representa el nombre del archivo, tal como se encuentra en el archivo **dbfiles**[I. La función **datos** llama a una **gran** variedad de funciones incluyendo las que representan el "Manejador de Pantallas" de CDATA [STEV87].

En la implementación de los Módulos nombrados se utiliza también el archivo ***KEYS.H*** y la función ***GET_CHAR()***. KEYS.H contiene una serie de comandos ***#define*** que equivalen símbolos globales a los valores retornados por el teclado IBM PC, cuando se presionan las teclas funcionales.

FUNCION GET_CHAR: int get_char ()

Esta función traduce las teclas funcionales en los valores enteros definidos por KEYS.H.

B. MODULO DE DOCUMENTOS (MENU PRINCIPAL).-

Luego de escogida la opción Documentos del Menú Principal, se presentan al usuario las opciones de: Tópico, Ingreso, Actualización/Consulta, Eliminación y Salir. El usuario no puede ingresar a ninguna de las tres opciones intermedias nombradas si no se ha establecido ántes en el Sistema un Tópico Corriente. Es que es necesario saber con qué Archivo de Documentos, de Palabras y de Búsquedas Grabadas se quiere trabajar.

B.1 OPCION **TOPICO** (MENU DE DOCUMENTOS).-

La opción Tópico del Menú de Documentos se implementa mediante las funciones del archivo ***TOPICO.C***. Se presentan en pantalla todos los Tópicos que han sido previamente de-

finidos por el usuario en el Módulo de Tópicos del Menú de Mantenimiento. Se le permite al usuario escoger uno de esos Tópicos y en ese momento los Archivos de Documentos. de Palabras y de Búsquedas Grabadas de ese Tópico se convierten en los Archivos Corrientes del Sistema.

FUNCION TOP_LISTA: int top_lista ()

Esta función abre el Archivo de Tópicos y copia los nombres de los Tópicos en un arreglo global de punteros a caracter llamado **tl**. Retorna el número de Tópicos en el arreglo.

FUNCION ESCOGER_TOP: void escoger_top ()

Este función trabaja con los contenidos de las variables globales del Sistema:

- ***int esc_top***: TRUE o FALSE, dependiendo de que se haya o no escogido en el Sistema un Tópico Corriente.
- ***char top_actual[16]***: Nombre del Tópico Corriente.
- ***char letra_actual[2]***: Código del Tópico Corriente.
- ***int num_actual***: Número actualizado de Búsquedas Grabadas para el Tópico Corriente.

Pueden darse algunas razones por las que el usuario haya escogido la opción Tópico del Menú de Documentos: se quiere escoger por primera vez un Tópico, se desea cambiar de Tópico o, se desea recordar cuál es el Tópico Corriente.

Cuando explicamos la Definición de Archivos de CDATH dijimos **que** todos los Archivos de Documentos estaban representados por el nombre ***ARCH*** y todos los Archivos de Palabrae por el nombre ***ARCH_PAL***. Es en el momento de escoger o de cambiar de Tópico que, por medio del comando ***rename*** del DOS, se renombran: el Archivo de Documentos y sus Indices. el Archivo de Palabras y sus Indices y se los convierte en: ARCH.DAT, ARCH.X01, ARCH.X02, ARCH.X03 y ARCH_PAL.DAT ARCH_PAL.X01, ARCH_PAL.X02, ARCH_PAL.X03. De esta manera forman ya parte de la Definición de Archivos de CDATA y pueden ser manejados mediante las funciones del Sistema Manejador de la Rase de Datos.

Como la variable ***top-actual*** tiene almacenado el Nombre del Tópico Actual, será posible en el momento preciso, que los Archivos de Documentos y de Palabras vuelvan a ser renombrados y obtengan sus nombres **originales** y verdaderos.

Cuando se produce un cambio de Tópico o cuando se abandona el Sistema, los Archivoa ARCH.DAT y ARCH_PAL.DAT y todos los Indices relacionados, toman sus nombres verdaderos. En el caso de un cambio de Tópico, los Archivos pertenecientes al nuevo Tópico escogido pasarán a llamarse ARCH.DAT y ARCH_PAL.DAT. En ese mismo momento se almacena en el Registro de Tópico del Tópico que fue reemplazado, el Número actualizado de Búsquedas Grabadas. La variable *num_actual* pasará a almacenar el Número de Búsquedas Grabadas del nuevo Tópico escogido.

Luego de escogido un Tópico se puede proceder a ingresar, actualizar, consultar o eliminar registros de su Archivo de Documentos. A diferencia de los Módulos de Fuentes, Idiomas, Tipos de Documentos y Tópicos, en los cuales todas estas operaciones se realizaban por medio de una sola pantalla, para los Documentos existen tres pantallas independientes.

B.2 OPCION DE INGRESO (MENU DE DOCUMENTOS)---

La Pantalla de Ingreso está implementada mediante la función *ING_DOCS* perteneciente al archivo *DOC_ING.C*.

FUNCION ING_DOCS: void ing_docs ()

Esta función no necesita recibir como argumento el Archivo de CDATA con el que debe trabajar, porque trabaja directamente con ARCH.DAT, ARCH_PAL.DAT y sus respectivos Indices. **Ing_docs** no sabe el Nombre **del** Tópico, toma directamente los Archivos Corrientes.

Primeramente se presentan al usuario los campos de: Código del Documento, Título, Autor, Código de Fuente, Año y Mes de Publicación, Código de Idioma, Número de Páginas y Referencia. Se aceptan unicamente Códigos de Documento que no existan previamente en el Archivo. El usuario puede ingresar valores en el resto de campos y recorrerlos hasta estar satisfecho de su contenido.

Para facilitar el ingreso de datos en los campos de Fuentes e Idiomas, la tecla <F10> provee Ventanas de Consulta. En ellas el usuario podrá observar los Códigos de Fuentes o Idiomas ya existentes en el Sistema y el Nombre de Fuente o Idioma que representa cada uno. De no ser deseado ninguno de los Códigos existentes, el usuario podrá llenar los campos con Códigos nuevos de Fuentes o Idiomas. Las Ventanas de Consulta se implementan mediante la ***FUNCION VENTANITA: void ventanita (int x, int y , char *s, int *iplist)*** del archivo fuente *DS.C*.

Luego de ingresar este primer bloque de datos. se debe presionar la tecla <F5> y el sistema preguntará sucesivamente si el documento ingresado es un Artículo de Revista o Periódico, **Capitulo** de Libro, Libro o Tomo de Enciclopedia. Proyecto o Tesis. Seminario o Conferencia. Cuando el usuario responda que "SI" a un determinado Tipo de Documento, su Código sera asignado al registro del Documento. Dependiendo **del** tipo establecido aparecerán en la pantalla diversos campos. Es que como habiamos explicado, existen campos en el Registro de Documentos que sólo tienen validez para ciertos Tipos.

Una vez ingresados los campos alternativos, se le pide al usuario **que** presione nuevamente la tecla <F5>. De esta manera se pasa a la Segunda Pantalla de **Ingreso**, implementada por la ***FUNCION RESUMEN: void resumen (char tclaves [284], char tresumen[704], int modo)***. y que contiene los campos para Ingreso de: Palabras Claves y Resumen. Las Palabras Claves deben estar separadas por comas (´,´). Se le pide además al usuario que no separe. mediante un guión. ninguna palabra al final de las líneas.

Luego de ingresados los campos de Palabras Claves y Resumen el **usuario** debe presionar nuevamente la. tecla <F5>.

y el sistema le preguntara si se desea Grabar el Registro en el Archivo de Documentos. Si el usuario responde afirmativamente se llevan a cabo una serie de acciones. En primer lugar se graba el registro en el archivo de Documentos del Tópico Corriente. En segundo lugar, se chequea si los valores de Código de Fuente, Idioma y Tipo de Documento del presente registro de Documento, existen en sus respectivos archivos. Si algunos de estos valores no forma parte de su archivo, se construye para ese archivo un registro vacío cuyo único campo lleno es el de Código. Eeta accion la ejecuta la ***FUNCION RELACIONES: void relaciones (int f, char *bf).***

Podemos ver de esta manera que existen hasta ahora dos formas de ingresar registros en los Archivos de Fuentes, Tipos de Documentos e Idiomas: por medio de sus respectivos Módulos, o mediante el ingreso de un Documento que contenga Códigos nuevos para estos campos.

En caso de haber sido necesario ingresar registros nuevos en los archivos de Fuentes c Idiomas o Tipos de documentos, el sistema lo notificara al usuario y le indicará que para ingresar los campos restantes de los registros, deberá recurrir A los Módulos respectivos.

Al ingresar un Documento se construyen una serie de Registros de Palabras para el Archivo ***de*** Palabras del Tópico Corriente. Cada registro contiene una Palabra Clave o una Palabra del Título del Documento. No hay que olvidar que en el Registro de Documentos. el campo de Título constituye una sóla cadena de 60 caracteres para todo el Título y el campo de Palabras Claves a su vez una única cadena de 264 caracteres, con las Palabras Claves separadas por comas tal y como las ingresó el usuario. Así pues, la ***FUNCION PAL_TRAT: void pal_trat (char *x, char *linea, char *codigo, int tipo, int accion)*** hace uso de la ***FUNCION GET_WORD CLA_TIT: char *get_word_cla_tit (char *cp, int tipo)*** para desmembrar estos campos. tomar cada palabra del Título y cada Palabra Clave separada por comas. asociarlas individualmente al Código del Documento al cual pertenecen. e ingresar estos registros en el Archivo de Palabras del Tópico corriente.

Es de nombrarse la existencia de un Diccionario en el Sistema. Siempre al inicio de la ejecución del "Sistema Personal. de Manejo de Documentos", la ***FUNCION CREAR_DIC: void crear_dic ()*** es la encargada de abrir el Archivo del Diccionario ***DIC.DOC*** y cargarlo en la memoria. En él están contenidas una serie de preposiciones. conjunciones y

artículos, tanto en Español como en Inglés. Cada vez que se ingresan o modifican los Títulos o Palabras Claves de un Documento, cada palabra es comparada con los elementos del diccionario, mediante la ***FUNCION BUSCAR DIC: int buscar_dic (char *s).*** Si la palabra pertenece al Diccionario, no debe ser ingresada al Archivo de Palabras. Recordemos que los Archivos de **Palabras** son utilizados para la ejecución del algoritmo de Búsqueda de Documentos por Palabra Clave o del Título. No tiene ningún sentido que preposiciones, conjunciones o artículos participen en la Búsqueda de Documentos.

B.3 OPCJON DE ACTUALIZACTON Y CONSULTA (MENU DE DOCUMENTOS).-

Para la modificación y consulta de documento se utiliza la función ***MOD_DOCS***, perteneciente al archivo ***DOC_MOD.C.***

FUNCION MOD_DOCS: void mod_docs ()

De igual manera que ***ING_DOCS***, ***MOD_DOCS*** realiza todas sus operaciones sobre los archivos ARCH.DAT, ARCH_PAL.DAT y sus correspondientes Indices.

La utilización de las teclas: <PgUp>, <PgDown>, <Home>, <End>, <F2> y <ESC> es **similar a** la que explicamos

para los Módulos de Fuentes. Tipos de Documentos, Idiomas y Tópicos.

El Sistema provee, al igual que para el Ingreso, dos Pantallas para la Actualización y Consulta. La Primera Pantalla presenta los 13 primeros campos del Registro de Documentos y la Segunda, los campos de Palabras Claves y Resumen. Esta Segunda Pantalla es implementada por la función *RESUMEN*, ya nombrada anteriormente.

Existen nuevamente Ventanas de Consulta (función *VENTANITA*) para los campos de Fuentes, Tipos de Documentos e Idiomas. Si el usuario modifica estos campos mediante valores no existentes en el Sistema, la función *RELACIONES* los ingresará en los archivos respectivos.

Un caso especial lo constituye el archivo de Tipos de Documentos. Como vimos en la pantalla de Ingreso de Documentos, el valor de este campo se restringe a cinco únicas posibilidades (AR, LE, CL, PT y SC). Así pues, en la pantalla de Actualización y Consulta, el programa obliga al usuario a respetar la existencia de estos cinco únicos tipos definidos por el "Sistema Personal de Manejo de

Documentos". El usuario no podrá crear un Tipo nuevo, sino sólo asignar los ya definidos por la aplicación.

En caso de que el usuario modificara el Título o las Palabras Claves de un Documento, el sistema realiza diversas acciones sobre el Archivo de Palabras. Comparando el contenido del Título y las Palabras Claves anteriores con el Título y las Palabras Claves ya modificadas, se podrá: Eliminar del Archivo de Palabras los registros con palabras ya no existentes, Mantener los registros con palabras todavía existentes, e Ingresar registros con las palabras nuevas. Para llevar a cabo todas estas acciones se utilizan nuevamente las funciones: *BUSCAR_DIC*, *GET_WORD_CLA_TIT* y *PAL_TRAT*.

B.4 OPCION DE **ELIMINACION** (MENU DE DOCUMENTOS).-

La eliminación de Documentos la implenta principalmente la función *ELI_DOCS*, perteneciente al archivo fuente *DOC_ELI.C*.

FUNCION ELI_DOCS: ***void eli_docs ()***

De igual manera que *ING_DOCS* y *MOD_DOCS*, *ELI_DOCS* realiza todas sus operaciones sobre los archivos ARCH.DAT, ARCH-PAL.DAT y sus correspondientes Indices.

En la pantalla de Eliminación el usuario ingresa el Código del Documento a eliminar. Los datos del Documento serán presentados, pero no se permite movilizacion a través de los campos. Presionando la tecla <F5> se pueden ver los campos de Palabras Claves y Resumen. La tecla <F7> realiza efectivamente la eliminación.

Al eliminarse un Documento del Sistema, la ***FUNCION PAL_ELI: void pal_eli (char *x, char *codigo)*** elimina del Archivo de Palabras del Tópico corriente todos los registros que coincidan con el Código del Documento eliminado.

A diferencia del Archivo de Palabras, los Archivos de Fuentes, Idiomas y Tipos de Documentos no ven alterado su contenido.

C. MODULO DE BUSQUEDAS (MENU PRINCIPAL).-

Luego de escogida la opción Búsquedas del Menú Principal, se presentan al usuario las opciones de: Tópico, Búsqueda por Palabra, Búsqueda por Condicionantes y Salir. El usuario no puede realizar ningún tipo de Búsqueda, si no se ha establecido ántes en el Sistema un Tópico Corriente. Es que es necesario saber con qué Archivo de Documentos, de Palabras y de Búsquedas Grabadas se quiere trabajar.

C.1 OPCION TOPICO (MENU DE BUSQUEDAS).-

La opción Tópico del Menu de Búsquedas es totalmente equivalente a la opción Tópico del Menú de Documentos, pues se implementa mediante las mismas funciones.

C.2 OPCION DE BUSQUEDA POR PALABRA CLAVE O DRL TITULO (MENU DE BUSQUEDAS).-

Como dijimos previamente, cuando el usuario escoge la Opción de Búsqueda por Palabra Clave en el Menú de Búsquedas, el ejecutable ***SPMD.exe*** cede el control a ***BUS1.exe*** y permanece esperando que el programa hijo se ejecute, para poder luego retomar el mando (función *spawnl* del Leng. C).

La función ***main (int argc,char *argv[])*** de ***BUS1.exe*** es la encargada de soportar el Módulo de Búsqueda por Palabra Clave o del Título, llamando a su vez a una serie de funciones. Recibe como argumentos de ***SPMD.exe***: el Código del Tópico Corriente y el Número de Búsquedas Grabadas del Tópico Corriente.

El Algoritmo de Búsqueda que encuentra los Documentos relacionados directa e indirectamente a una Palabra fue ya explicado en la Sección 3.2 del presente Capítulo. Se basa en el árbol de la Figura 29 y se implementa básicamente me-

diante las ***FUNCIONES BUSQ_PAL:*** ***void busq_pal (char *pal)*** ***y PONER_DOCS:*** ***void poner_docs (int f, char *pal, char *om).***

La función ***BUSQ_PAL*** recibe la Palabra Inicial ***pal*** y la da como argumento a la función ***PONER_DOCS***, junto con el Número del Archivo de CDATA con el que se está trabajando (ARCH_PAL=1) y un modo de apertura ("w+" o "a+").

La función ***PONER_DOCS***, de acuerdo al modo de apertura que recibe, abre un archivo temporal llamado "TEMP_C1.TXT". Inicialmente el modo de apertura es "w+". En este archivo temporal se almacenan los Códigos de los Documentos que contienen a la Palabra Inicial como Palabra Clave o como parte de sus Títulos. Para la construcción de este Archivo se utiliza el **Indice por Palabras** del **Archivo de Palabras** del Tópico Corriente.

La función ***BUSQ_PAL*** utiliza el archivo "TEMP_C1.TXT" **para** crear a su vez un archivo llamado "TEMP_P1.TXT". Mediante el uso del **Indice por Códigos** del **Archivo de Palabras** del Tópico Corriente. se almacenan en este nuevo archivo, el tercer nivel del árbol explicado en la figura 29, es decir las palabras pertenecientes a los documentos directamente relacionados a la palabra inicial. Seguida-

mente se procede a eliminar de él las palabras repetidas y una copia unica de **cada** palabra queda almacenada en el archivo temporal "TEMP_P2.TXT". "TEMP_P2.TXT" contiene todas las **palabras** relacionadas a la Palabra Inicial.

A continuación se toman las **palabras** de "TEMP_P2.TXT". y **para cada** una de ellas se llama nuevamente a la función ***PONER_DOCS*** con modo de apertura "a+". Los Códigos de los Documentos que contengan cada Palabra se añaden al archivo "TEMP_C1.TXT". De esta manera, **una** vez recorridas todas **las** palabras de "TEMP_P2.TXT", en "TEMP_C1.TXT", estarán los Códigos de los Documentos relacionados directa e indirectamente con la palabra inicial. Finalmente, se eliminan los cód**igos** repetidos y una copia única de cada uno se almacena en el archivo "TEMP_C2.TXT".

Al finalizar la función ***BUSQ_PAL***, se eliminan los archivos "TEMP_P1.TXT" y "TEMP_C1.TXT". El sistema mantiene existentes los archivos "TEMP_P2.TXT" y "TEMP_C2.TXT".

FUNCION CREAR_ARCHIVO: void crear_archivo ()

Esta función abre el archivo "TEMP_C2.TXT" y para cada Código busca en el Archivo de Documentos corriente el nombre del Título correspondiente. Se crea entonces un

archivo llamado "TEMP_C3.TXT", el cual almacena registros formados por: el Código del Documento y los primeros treinta caracteres del Título.

En la implementación del Primer Submódulo de Búsqueda se utilizan una serie de funciones adicionales a las nombradas. Una vez que el usuario ha ingresado la Palabra que va a originar la Búsqueda, el Sistema le plantea dos alternativas: Se desea realizar todo el Proceso de Búsqueda o, se desea ver la Respuesta Grabada por el Sistema en una Búsqueda Previa de esa Palabra. Estas dos posibilidades se implementan mediante las ***FUNCIONES DET_NUEVA: int det_nueva ()*** y ***PROC_NUEVA: int proc_nueva (char *pal, char *archivo, int nueva).***

Si el Usuario escoge "Realizar una Búsqueda Nueva", el Sistema reviza el Archivo de Búsquedas Grabadas del respectivo Tópico. En caso de que algún registro contenga dentro de su primer campo la Palabra ingresada por el usuario, se toma el nombre del Archivo de Respuestas correspondiente y éste será usado para almacenar la nueva Respuesta del proceso de Búsqueda. De esta manera se está reemplazando el contenido del Archivo de Respuestas. Se eliminan de él los Códigos anteriores y se colocan ahora los Códigos de

los Documentos resultantes del nuevo proceso de Búsqueda, originado por la misma Palabra. Es de resaltar el hecho de que no se está aumentando ningún registro al Archivo de Búsquedas Grabadas, ni tampoco creando un nuevo Archivo de Respuestas en el Sistema. Unicamente se reemplaza el contenido de un Archivo de Respuestas ya existente. El valor de la variable ***num_actual*** (Número de Búsquedas Grabadas para el Tópico Corriente) tampoco varia.

Si no existe una Respuesta Grabada, se siguen los siguientes cuatro pasos:

1. Se utiliza el valor de la variable ***num_actual*** en la construcción del nombre del Archivo de Respuestas.

2. Se aumenta un registro al archivo de Búsquedas Grabadas del Tópico Corriente, especificando la Palabra Inicial de Búsqueda y el Nombre del Archivo de Respuestas.

3. Se crea un nuevo Archivo de Respuestas con el nombre ya mencionado, que contiene los Códigos de los Documentos que satisfacen la Palabra de Búsqueda.

4. Se incrementa en uno el valor de la variable *num_actual.*

Si el usuario escoge "Ver Respuesta de un Búsqueda Previa". el Sistema reviza el Archivo de Búsquedas Grabadas del respectivo Tópico, para constatar si efectivamente se tiene almacenada una Respuesta para la Palabra ingresada por el Usuario. De ser así se abre ese Archivo de Respuestas y se presentan los Documentos al Usuario.

Cabe decir que, si el Archivo de Búsquedas Grabadas presenta un nombre de Archivo de Respuestas para la Palabra ingresada por el usuario, pero por alguna razón ese Archivo de Respuestas no se encuentra en el Directorio Corriente. el Sistema realiza todo el proceso de búsqueda y almacena el resultado bajo el nombre del Archivo de Respuestas desaparecido.

A su vez, si no existe una respuesta grabada en el Archivo de Búsquedas Grabadas, el Sistema realiza todo el proceso de Búsqueda, y ejecuta los cuatro pasos mencionados anteriormente para dejar grabada su Kespuesta.

La presentación en pantalla de los resultados de las Búsquedas, se lleva a cabo con las *FUNCIONES INICIO: void inicio 'char *s, int tam_reg), TRAB_PANT: void trab pant () y ESCRIBIR: int escribir (ink tipo, char *s, int tam ran).*

Si se ha llevado a cabo una Búsqueda Nueva, el Submódulo de Búsqueda por Palabra Clave o del Título presenta al usuario, además de la Palabra Original de Búsqueda, dos Ventanas Superiores: una con las Palabras Relacionadas con la Inicial (tercer nivel del árbol de la Figura 29) y otra con los Códicos de los Documentos relacionados directa e indirectamente a ella (segundo y cuarto niveles del árbol de la Figura 29) y los 30 primeros caracteres de sus respectivos Títulos. No se realiza distinción entre Documentos Directa e Indirectamente relacionados. La ventana de Palabras Relacionadas toma sus valores del Archivo Temporal "TEMP_P2.TXT" y la ventana de Códigos y Títulos del Archivo Temporal "TEMP_C3.TXT", creado por la función *CREAR_ARCHIVO.*

Si el usuario ha pedido ver la Respuesta a una Búsqueda Previa, sólo se podrá presentar a él, además de la Palabra Original de Rusqueda, una Ventana Superior con los Códigos y Títulos de los Documentos seleccionados. No se

presentan Palabras **Relacionadas**, pues no existe para este caso el archivo "TEMP_P2.TXT".

Las funciones ***TRAB_PANT y ESCRIBIR*** implementan las ventanas mencionadas y permiten al usuario utilizar diversas teclas explicadas a continuación. Una vez posicionado en una Ventana las teclas <PageUp> y <PageDown>, permiten recorrer las Palabras Relacionadas o los Documentos Seleccionados pertenecientes a esa Ventana. En caso de existir dos ventanas, la tecla <TAB> permite pasar alternativamente de una de las ventanas a la otra. Estando posicionado en cualquier ventana se puede presionar la tecla <F10>. En este momento se llama a la ***FUNCION CONSULTAR_BUSQ: void consultar_busq ()***, y aparece en la parte inferior de la Pantalla una Ventana de Consulta con los siguientes campos: un campo de ingreso del Código de un Documento y campos sólo de lectura para Autor, Título, Código de la Fuente del Documento, Referencia, Año y Mes de Publicación.

Mediante esta Ventana de Consulta el usuario puede ingresar el Código de uno de los Documentos seleccionados y así conocer los datos de Título Completo, Autor y Código de Fuente. Estando en esta Ventana la tecla <ESC> limpia los campos y permite el ingreso de otro Código de Documento.

Si la ventana está limpia y se presiona <ESC>, se abandona la Ventana de Consulta y se regresa a una de las Ventanas Superiores.

Posicionado en cualquier ventana superior, se puede también presionar la tecla <F7>. En este momento se llama a la ***FUNCION REPORTES: void reportes()***. Esta función permite imprimir los siguientes datos de los Documentos seleccionados por el proceso de Búsqueda: Código del Documento, Título, Autor y Código de Fuente. El Reporte indica además la palabra que originó la Búsqueda.

Por último, estando en cualquiera de las dos Ventanas superiores, la tecla <ESC> permite regresar a la Pantalla de Ingreso de la Palabra de Búsqueda. En este momento se eliminar los Archivos temporales "TEMP_C2.TXT", "TEMP_C3.TXT" y "TEMP_P2.TXT". Si en vez de ingresar una nueva Palabra, se presiona la tecla <ESC>, ***BUS1.exe*** finaliza su ejecución y regresa el mando a ***SPMD.exe***.

Recordemos que ***BUS1.exe*** recibió de ***SPMD.exe*** el valor de la variable ***num_actual*** previa a la ejecución de búsquedas. Para poder actualizar esta variable luego de retomado el mando, ***SMPM.exe*** se da cuenta de la cantidad de Registros

en el Archivo de Búsquedas Grabadas y así puede asignar un nuevo valor a ***num_actual***.

C.3 OPCION DE BUSQUEDA POR CONDICIONANTES (MENU DE BUSQUEDAS).-

Cuando el usuario escoge la Opción de Búsqueda por Condicionantes del Menú de Búsquedas, el ejecutable ***SPMD.exe*** cede el control a ***BUS2.exe*** y permanece esperando que el programa hijo se ejecute, para poder luego retomar el mando (función ***spawnl*** del Lenguaje C).

La función ***main()*** de ***BUS2.exe*** no recibe ningún Argumento de ***SPMD.exe*** y es la encargada de llamar a una serie de funciones que implementan el Segundo Submódulo de Búsqueda.

Para facilitar el ingreso de datos en las Condiciones de: Código de Fuente, Código de Idioma, y Código de Tipo de Documento, la tecla <F10> provee Ventanas de Consulta para cada uno de estos campos. En ellas el usuario podrá observar los Códigos de Fuentes, Idiomas y Tipos de Documentos ya existentes en el Sistema. Estas Ventanas de Consulta son las implementadas por la función ***VENTANITA***.

La función principal es *BUSQ2: int busq1(struct soluc2 *t).* *BUSQ2* va revizando los campos de la Pantalla de Condicionantes, para asi poder decidir qué estratègia de Búsqueda se va a utilizar.

Existen diversos casos:

1. Se ha ingresado alguna Palabra Clave o del Titulo.
2. No se ha ingresado ninguna Palabra Clave o del Titulo, pero sí un valor en el campo de Autor.
3. No se han ingresado valores en los campos de Palabras, ni Autor, pero sí en el campo de Fuente.
4. No se han ingresado valores en los campos de Palabras, Autor, ni Fuente, pero sí en algún campo restante.

El ubicar una Búsqueda en uno de estos cuatro casos, permite a la funcion *BUSQ2* determinar qué Indice del Archivo de Documentos se utilizará para resolver la Búsqueda. El objetivo general de esta función es almacenar los registros que cumplen todas las Condiciones en el archivo temporal "TEMP_C2.TXT".

Para el **Primer Caso**, se reviza primeramente si alguna de tres posibles Palabras de Búsqueda existe efectivamente en el Archivo de Palabras del Tópico Corriente (función

verify_red()* de *CDATA). Si ningún Documento contiene ninguna de las **Palabras**, se puede ya comunicar al usuario que no existe respuesta a las Condiciones de Búsqueda.

Si por lo menos una de las Palabras Claves o del Título pertenecen al Archivo de Palabras, el Sistema se asegura de que, de existir un valor en la Condición de Autor, éste sea valido. Es que no vale la pena realizar el Algoritmo de Búsqueda por Palabra, si ningún Documento va a poder cumplir con el campo no válido de Autor. Si el patrón de Autor es inválido, se puede ya comunicar al usuario que no hay respuesta a las Condiciones de Búsqueda.

De no presentarse ningún problema con el Campo de Autor, se llama a la función ***BUSC_PAL*** para cada una de las Palabras ingresadas por el usuario y que existen en el Sistema. De esta manera se da inicio al Algoritmo de Búsqueda por Palabra, ya conocido. Ya que se crea un árbol de Búsqueda para cada Palabra (Figura 29), el Sistema utiliza diversos archivos temporales para unir las respuestas procedentes de cada árbol. Finalmente los Códigos no repetidos de los Documentos que satisfacen por lo menos una de las palabras ingresadas por el usuario se encuentran el el Archivo "TEMP_C2.TXT". Las ***FUNCIONES UNION_TEMP Y***

UNION2: ***void union-temp (char *s)*** ***y*** ***void union2 ()*** son las encargadas de la unión de los árboles v de la creacion de la primera versión del archivo "TEMP_C2.TXT".

Hasta el momento tenemos un subconjunto de Documentos Seleccionados que cumplen con la Condición de Palabra. Para asegurar el cumplimiento de las demás condiciones, se llama a la ***FUNCION CHEQ_PAL:*** ***int cheq_pal (struct soluc2 *t).*** Esta función abre el archivo temporal "TEMP_C2.TXT" y para cada Código de Documento llama a la ***FUNCION ACEPTAR:*** ***int aceptar (struct soluc2 *t, struct docs *r).*** La función ***ACEPTAR*** compara cada campo del Documento correspondiente al Código, con las condiciones ingresadas por el usuario: Rango de Publicación en mes y año; Códigos de Fuente, Idioma y Tipo; y Autor. La función ***CHEQ_PAL*** almacena en el archivo temporal "TEMP_C3.TXT" los Códigos de Documentos que, además de cumplir con la Condición de Palabra Clave o del Título, cumplan con las demás condiciones del usuario. Finalmente se renombra a "TEMP_C3.TXT" como "TEMP_C2.TXT".

En el **Segundo Caso. el** campo de Autor ha sido llenado y es neiesario revisar la validez de su contenido. Recordemos que se permite ingresar un patrón de Autor incompleto, usando como "comodin" uno o más ***"*" al final de la***

cadena. Por ej: al ingresar "STE*", cumplirán esta condición todos los Documentos cuyo nombre de Autor empiece con las letras "STE". Si no se coloca un "*" al final del nombre de Autor, se asume que éste está completo y se buscan los Documentos que cumplen exactamente con el nombre de Autor ingresado.

La *FUNCION BUSQ_ASTER: int busq_aster (char *s, char word[41])*, retorna:

- **FALSE** si no existe un asterisco en el final de la cadena,
- **TRUE** si existe un asterisco como fin de cadena, y
- **ERROR** si después de encontrar un asterisco en la cadena, los caracteres siguientes no son asteriscos.

De acuerdo al valor retornado por *BUSQ_ASTER* la funcion *BUSQ2* toma diferentes acciones. En caso recibir el valor de ERROR, se le notifica al usuario que el asterisco de la Condición de Autor debe estar al final de la cadena y que no hay respuesta a las Condiciones de Búsqueda.

Si se ha retornado **FALSE**, significa que el usuario ha ingresado un valor exacto de Autor (no hay comodín). El Sistema se avegura de que el Autor pertenezca a por lo me-

nos un Registro del Archivo de Documentos del Tópico Corriente (función ***verify_rcd***). De ser asi llama a **la** ***FUNCION BUSQ_CAMPO: int busq_campo (int f, struct soluc2 *t, int tipo)***, con **valor** de **tipo** igual **a** 0. Esta función será explicada más adelante. En caso contrario notifica al usuario que no hay respuesta a las condiciones ingresadas.

Si la función ***BUSQ_ASTER*** retorna **TRUE**, significa que se está utilizando un "comodin" en el campo Autor. El Sistema llama a la ***FUNCION BUSQ_PAT: int busq_pat (int f, struct soluc2 *t, char *patron).*** Esta busca en el Archivo de Documentos, por medio del Indice de Autor, los Documentos cuyo Autor cumpla con el patrón ingresado.

El algoritmo es el siguiente:

1. Se crea un Registro de Documento vacio.

2. El campo del Identificador se llena con 10 cruces minúsculas y el campo del Autor con el patrón ingresado por el usuario, eliminado el asterisco. Este registro se ingresa en el Archivo de Documentos de Tópico corriente.

3. Mediante la funcion ***find_rcd()*** de CDATA y el Indice por Autor se localiza al registro recién ingresado. La función ***next_rcd()*** nos permite seguir secuencialmente el Indice por Autor y comparar los valores de Autor con el Patrón. Los registros que contengan al patrón se envían como argumento a la funcion ***ACEPTAR.***

4. Mientras el Autor siga cumpliendo con el patrón, la función ***ACEPTAR*** chequea los restantes campos del registro de Documentos en función de los campos ingresados por el usuario en la Pantalla de Condicionantes.

5. Los registros que cumplen todas las condiciones se almacenan en un archivo temporal "TEMP_C2.TXT".

6. Cuando, siguiendo el Indice por Autor, se descubre que ya se deja de cumplir el patrón, se elimina el registro cuyo Identificador es "xxxxxxxxxx" y se da por terminada la Búsqueda.

En el **Tercer Caso,** si no están llenos los campos de Palabras, ni tampoco el de **Autor**, se pasa a revizar la condición de Código de Fuente. Si este campo tiene un va-

lor existente en el Sistema (funcion ***verify_rcd()***) se llama a la funcion ***BUSQ_CAMPO*** con su argumento **tipo** igual a 1.

En el **Cuarto Caso.** si tampoco está lleno el campo de Código de Fuente. se analizan las condiciones restantes llamando a la función ***BUSQ_CAMPO***, con el argumento de **tipo** igual a 2.

Podemos apreciar asi que la función ***BUSQ_CAMPO*** ayuda en las estrategias **del** Segundo (**tipo**=0). Tercer (**tipo**=1) y Cuarto Caso (**tipo**=2). Estos tres casos tienen en común que el usuario no ha llenado las condiciones de Palabras. Dependiendo del valor de **tipo** se utiliza en la estrategia de Búsqueda un determinado Indice del Archivo de Documentos.

Si **tipo**=0, se ha ingresado un valor exacto en la Condición de Autor. Se utiliza entonces el Indice por Autor. La función ***find_rcd()*** ubica al primer registro del Archivo de Documentos que cumple con el valor. de Autor ingresado por el usuario. Siguiendo secuencialmente el Indice (función ***next_rcd()***) se va llamando a la función ***ACEPTAR*** una vez por cada Documento que cumple con el Autor. Los documentos aceptados por la función ***ACEPTAR*** se graban en el archivo "TEMP_C2.TXT".

Si tipo=1, existe la Condición de Código de Fuente y se utiliza en la estrategia de Búsqueda el Indice por Código de Fuente del Archivo de Documentos. La función *find_rcd()* ubica el primer registro que cumple con la Fuente. A medida que se recorre secuencialmente el Indice, se va llamando a la función *ACEPTAR* una vez por cada Documento que cumple con la Condición de Fuente. Se graban en el archivo "TEMP_C2.TXT" los Códigos de los Documentos, que además de satisfacer la Fuente, cumplen los demás Condicionantes.

Si tipo=2, se toma el Indice de Código de Documento y se recorre todo el archivo de Documentos desde el principio. Cada documento es pasado a la función *ACEPTAR* y aquellos que cumplen con todas las Condiciones ingresadas por el usuario son grabados en el archivo temporal "TEMP_C2.TXT".

Una vez construído el archivo "TEMP_C2.TXT" REI deben presentar al usuario los resultados de la Búsqueda. Sin importar la estrategia de Búsqueda seguida, la función *CREAR_ARCHIVO* abre a "TEMP_C2.TXT" y crea a "TEMP_C3.TXT". En este último archivo se incluyen los Códigos de los Documentos Seleccionados y los primeros treinta caracteres de sus respectivos Títulos.

El contenido de "TEMP_C3.TXT" se presenta en una única Ventana de Respuesta, utilizando la función ***ESCRIBIR.*** Esta permite utilizar las teclas <PageUp> y <PageDown> para recorrer los Documentos Seleccionados presentados en la Ventana. De igual manera que para la Búsqueda por Palabra, la tecla <F10> presenta una Ventana de Consulta en la parte inferior de la Pantalla, donde el usuario puede ingresar el Código de uno de los Documentos Seleccionados y así conocer los datos de Título, Autor y Código de Fuente y Referencia.

Estando en esta Ventana de Consulta, la tecla <ESC> limpia los campos y permite el ingreso de otro Código de Documento. Si la ventana está limpia y se presiona <ESC>, se abandona la Ventana de Consulta y se regresa a la Ventana de Documentos Seleccionados.

Posicionado en cualquier ventana superior, se puede también presionar la tecla <F7>. En este momento se llama a la ***REPORTES,*** la cual imprime los siguientes datos de los Documentos seleccionados por el proceso de Búsqueda: Código del Documento, Título, Autor y Código de Fuente, además de todos los Condicionantes que originaron la Búsqueda.

Finalmente, si estando en la Ventana Superior se presiona la tecla <ESC>, el usuario regresa a la Pantalla de Ingreso de Condicionantes. En este momento se eliminan los Archivos temporales "TEMP_C2.TXT", y "TEMP_C3.TXT". Si en vez de ingresar nuevos Condicionantes, el usuario presiona la tecla <ESC>, ***BUS2.exe*** finaliza su ejecución y regresa el mando a ***SPMD.exe***.

D. MODULO DE GENERACION DE REPORTES (MENU PRINCIPAL).-

El Módulo de Generación de Reportes se implementa mediante las funciones de los archivos fuente ***IMP.C***, ***IMP_FUN.C*** y ***IMP_SHOW.C***, y permite imprimir el contenido de los Archivos de: Documentos, Tipos de Documentos, Fuentes, Idiomas y Tópicos. En el caso del Archivo de Documentos, el usuario debe especificar el Tópico al que pertenecen los Documentos, y el campo por el cual deben estar ordenados los Registros del Reporte: Código del Documento, Autor o Código de Fuente. El Sistema pregunta al usuario si se usará Papel Continuo o Individual, y en el último caso si el tamaño del papel es Oficio o Carta. La impresión se realiza siempre con letra condensada y a 80 columnas.

Del Archivo de Documentos se imprimen únicamente: el Código del Documento. el Título. Autor y el Código de la Fuente.

Del Archivo de Fuentes se imprimen: el Código de la Fuente. su Nombre, Ciudad. Teléfono y Código Postal.

De los Archivos de Tipos de Documentos. Idiomas y Tópicos se imprimen los campos de Código y Nombre.

La impresión se realiza de manera semejante al comando **PRINT** del DOS. El Sistema chequea que la impresora esté encendida y en línea. Si el usuario apaga la impresora. el programa cancela la impresión y espera una nueva orden.

E. MODULO DE MANTENIMIENTO (MENU PRINCIPAL).-

Al escogerse la Opción Mantenimiento del Menú Principal se pasa a un nuevo Menú con las Opciones de: Actualizar Códigos. Reindexar Archivos. Vaciar o Inicializar Archivos, Crear la Base de Datos y Salir.

La opción 'Actualizar Códigos' presenta a su vez los Módulos de Tópicos, Idiomas y Tipos de Documentos. que fueron previamente explicados. junto con el Módulo de

Fuentes del Menu **Principal.** A continuación explicaremos las opciones restantes.

E.1 REINDEXACION E INICIALIZACION DE ARCHIVOS.-

El archivo fuente ***INDEXAR.C*** contiene las funciones que implementan los Módulos de Reindexación e Inicialización de Archivos de **la Base** de Datos. La función principal es ***FUNCION INDICES: void indices (tipo).*** Dependiendo del valor de **tipo**, ***INDICES*** llama a la ***FUNCION PROCESO: void proceso (int dentro, char *s, int a, int tipo)*** para Reindexar o a **la** ***FUNCION VACIAR:*** **void** ***vaciar (int dentro, char *s, int a, int tipo)*** para Inicializar.

En estos dos Módulos, si el usuario desea trabajar con un Archivo de Documentos, se le pide que especifique el Tópico al que el Archivo se refiere. Al Reindexar o Inicializar un Archivo de Docuemntos se esta reindexando e inicinlizando automáticamente su respectivo Archivo de Palabras.

Ya **que** inicializar un Archivo por equivocación puede ser un error irreparable, el Sistema advierte varias veces al usuario que el contenido del Archivo va a ser eliminado. Además, ántes de inicializar el Archivo, éste es respaldado

con la extensión ".OLD". En el caso de un Archivo de Documentos, se respalda también el correspondiente Archivo de Palabras. Los Indices no son respaldados, de manera que si el usuario desea retomar un Archivo ".OLD", deberá renombrarlo como ".DAT" desde el prompt del Sistema Operativo y luego proceder a reindexarlo con la opción 'Reindexar Archivos' del Menú de Mantenimiento, para crear sus correspondientes Indices.

El Sistema no permite la Inicialización del Archivo de Tópicos, pues esto implicaría borrar todos los registros representativos a los Tópicos del Sistema y junto con ellos todos los Archivos de Documentos, Palabras y Búsquedas Grabadas. Por lo tanto el Sistema le pide al usuario que recurra al Módulo de Tópicos para, de quererlo, eliminar uno a uno los Tópicos del Sistema con sus respectivos Archivos asociados.

E.2 MODULO DE CREACION DE LA BASE DE DATOS (MENU DE MANTENIMIENTO).-

El Módulo de Creación de Base de Datos se implementa mediante la ***FUNCION CREACION:*** **void** ***creacion ()*** ubicada en el archivo fuente ***CREACION.C*** .

Esta función hace uso de las funciones de CDATA ***file_create()*** ***y build_index()*** para construir por primera vez Archivos e Indices en base a los nombres definidos en los arreglos ***dbfiles[] e index_ele[].*** Así pues, se crearán inicialmente los Archivos: ARCH.DAT, ARCH_PAL.DAT, TIPOS.DAT, FUENTES.DAT, IDIOMAS.DAT, TOPICOS.DAT y todos los Indices respectivos.

Como en el momento de la creación de la Base de Datos no existe ningún Tópico definido por el usuario, tampoco deben existir Archivos de Documentos y Palabras. Por esta razón, la función ***CREACION*** elimina de la Base de Datos los archivos ARCH.DAT, ARCH_PAL.DAT y todos sus Indices.

Dijimos anteriormente que cada vez que el usuario ingresa un Tópico al Sistema, mediante el Módulo de Tópicos, se creaban para ese Tópico: un Archivo de Documentos, un Archivo de Palabras, un Archivo de Búsquedas Grabadas y todos los Indices respectivos. Hablando con más exactitud se puede decir que en la función ***DATOS***, cada vez que el usuario ingresa un Tópico, el Sistema utiliza las funciones ***file_create() y bindex2()***, para crear **Archivos** de Documentos y de Palabras Vacíos y **con** los siguientes nombres: VACI.DAT, VACI_PAL.DAT, etc. Estos Archivos Vacíos ayudan

en la creación de los Archivos de Documentos y de Palabras definitivos.

LLama la atencion el hecho de que en la definición de CDATA, (poi- ej: arreglos ***dbfiles[]*** e ***index ele[]***) no **existe** nunca un nombre semejante a "VACI" o "VACI PAL". La Aplicación "engaña" momentaneamente A CDATA, obligándola A crear Archivos e Indices no definidos por ella. Como ***file-create*** tiene como argumentos: el nombre con extensión **del** Archivo y la longitud de un registro, la función ***DATOS*** le envía coms, **nomhre** "VACI.DAT" y como longitud la longitud real **de** "ARCH" ***(file-create ("VACI.DAT",rlen(ARCH))***. De **igual** manera **para** crear a "VACI_PAL.DAT". ***file_create*** recibe como nombre "VACI_PAL.DAT" y como longitud la de "ARCH_PAL" (***file_create ("VACI_PAL.DAT", rlen (ARCH_PAL))***). Para crear los respectivos Indices hubo que diseñar una segunda versión de la función de CDATA ***build_index***, llamada esta vez ***FUNCION BINDEX2: void bindex2 (char *path, int f, char *fname)***. A diferencia de ***build_index***, que utiliza las definiciones de CDATA, ***bindex2*** recibe el nombre sin extensión de los Indices, dentro de su argumento **fname**. Asi pues, para crear los Indices de VACI.DAT y VACI_PAL.DAT, se utilizaron los comandos: ***bindex("", ARCH, "VACI")*** y ***bindex('",ARCH_PAL, "VACI_PAL")***.

El Sistema **crea** estos Archivos Vacíos ("VACI.DAT", "VACI_PAL.DAT", y sus Indices). y luego los renombra: sus 4 primeras letras ("VACI") son reemplazadas por las 4 primeras letras del Nombre del Tópico. En ese momento se han construído ya: un Archivo de Documentos y un Archivo de Palabras para un Tópico Determinado. Para crear el Archivo de Búsquedas Grabadas se utilizan simplemente los comandos **fopen** *y* **fclose.** del Lenguaje C.

Debe quedar claro que en 1a Base de Datos del "Sistema Personal de Manejo de Documentos" existen estos Archivos e **Indices** con nombres "VACI.DAT", "VACI_PAL.DAT". etc, sólo en el momento ciel ingreso de un nuevo Tópico. En seguida estos son renombrados, cumpliendo así su función y desapareciendo de la Base **de** Datos.

Reg:esando a **la** función *CREACION*, cabe decir que ántes **de** crear la Base de Datos, ésta chequea la existencia en el **director**io corriente de archivos "*.PAT" y "4.SO*". Si este tipo de Archivas existen, el Sistema alerta al usuario la posible existencia de una Base de Datos. El usuario podrá decidir si quiere mantener esta Base antigua o eliminarla y **crear** una Base nueva.

3.5 MANUAL DEL USUARIO.-

A. GENERALIDADES.

A.1 USO DE LOS MENUS.-

El "Sistema Personal de Manejo de Documentos" [illegible] formado por una serie de Menús Verticales y Horizonta[illegible] Para [illegible] correcto [illegible] tome en cuenta los siguientes punto[illegible]

Usted puede movilizarse a través de las Opciones de los Menús, utilizando las teclas de Flechas Arriba y Abajo o las Flechas de Derecha e Izquierda.

- Para escoger una determinada Opción, [illegible] sobre ella y [illegible] la tecla -ENTER-.
  Existen Menús cuya primera letra del nombre de cada Opción se presenta en un color resaltante. Para [illegible] tecla de Menú si Ud. presiona justamente esa tecla se ejecuta la opción respectiva.

Para abandonar un Menú escoja la Opción Salir.

A.2 USO DE TECLAS.

Cada Módulo del Sistema Personal de Manejo de Documentos [illegible] de diversas teclas para llevar a cabo

ciertas funciones. Usted puede encontrar la información sobre las teclas disponibles en cada Módulo, en la parte superior o inferior de la pantalla de ese Módulo. Existen sin embargo teclas de uso común a lo largo de todos los Módulos del Sistema:

- Presione la tecla <ESC> cuando quiera limpiar los contenidos de los **campos** de una Pantalla.

- Si los campos de una Pantalla están limpios, presione la tecla <ESC> para abandonar el Módulo.

B. INSTALACION.-

Al instalar por primera vez el "Sistema Personal de Manejo de Documentos" (SPMD) tome en cuenta los siguientes pasos:

1. Los archivos necesarios para el funcionamiento del sistema son: SPMD.EXE, BUS1.EXE. BUS2.EXE. DIC.DOC. Asegúrese de tenerlos todos en **el** Directorio Corriente de su Disco Duro.

2. Para ejecutar el SPMD. ingrese en el Prompt del Sistema Operativo el comando:

C:\ spmd

y presione la Tecla <ENTER>.

3. Escoja la Opción ´Mantenimiento´ en el Menú Principal.

4. Escoja la Opción ´Crear Base de Datos´ en el Menú de Mantenimiento. Una vez dentro de este Módulo, el Sistema le preguntará si desea Crear e Inicializar todos los Archivos de la Base de Datos. Escoja la Opción OK. A continuación se presentará a Usted una Ventana con la longitud en Bytes del Registro de cada Archivo creado. Cuando aparezca el Mensaje "Presione una Tecla", el Proceso de Creación de la Base de Datos habrá terminado. Presione cualquier tecla y abandonará este Módulo.

5. Escoja la Opción ´Actualizar Códigos´ en el Menú de Mantenimiento, y seguidamente la opción ´Materias/ Tópicos´. Ingrese los Tópicos a los que se refieren los Documentos que Usted piensa ingresar en la Rase de Datos. No sobrepase el límite máximo de 19 Tópicos y asigne a cada Tópico un Código diferente. Ejs:

NOMBRE	CODIGO
COMPUTACION	C

ELECTRONICA	E
MEDICINA	M
MATEMATICAS	A

6. Escoja la Opción 'Tipos de Documentos'. ubicada dentro de la Opción 'Actualizar Códigos' del Menu de Mantenimiento. Ingrese los 5 Tipos de Documentos clefinidos por el "Sistema **Personal** de Manejo de Documentos":

CODIGO	DESCRIPCION
AR	ARTICULO DE REVISTA O PERIODICO
CL	CAPITULO DE LIBRO
LE	LIBRO O TOMO DE ENCICLOPEDIA
PT	PROYECTO O TESIS
SC	SEMINARIO O CONFERENCIA

8. Escoja la Opción 'Idiomas'. ubicada dentro de la Opción 'Actualizar Códigos' del Menú de Mantenimiento. Ingrese algunos Idiomas en los cuales Usted sabe que estarán escritos los Documentos que serán parte del Sistema. Ejs:

CODIGO	DESCRIPCION
ES	ESPANOL
IN	INGLES
AL	ALEMAN

C. MODULOS DEL SISTEMA PERSONAL DE MANEJO DE DOCUMENTOS.-

C.1 MODULO DE FUENTES (MENU PRINCIPAL), Y MODULOS DE TIPOS DE DOCUMENTOS, IDIOMAS Y TOPICOS (OPCION 'ACTUALIZAH CODIGOS' DEL MENU DE MANTENIMIENTO).-

Estos Mbdulos permiten al usuario el Ingreso, Actualización, Consulta y Eliminación de Fuentes/Editoriales, Tipos de Documentos, Idiomas y Tópicos. A continuación se presenter, las características comunes de estos Módulos:

- Para llamar a una Fuente, Tipo de Documento, Idioma o Tópico en particular, ingrese el dato correspondiente al primer campo de .la pantalla y presione la tecla <ENTER>. El primer campo es siempre llamado "Clave Principal". Dependiendo del Módulo, éste sera: el Código de la Fuente, el Código del Tipo de Documento, el Código del Idioma o el Nombre del Tópico.

- Si la Clave Principal existe en el respectivo Archivo, el Sistema recupera el Registro correspondiente y lo presenta en pantalla. Usted puede movilizarse a través de los restantes campos de la pantalla y modificar sus valores. Cuando se encuentre satisfecho con los

datos de una Fuente. Idioma. Tipo de Documento o Tópico, presione la tecla <F2> para que estos sean grabadoe. Si quiere rechazar los cambios efectuados sobre un Registro, presione la **tecla** <ESC>.

- Si se ingresa una Clave Frincipal que no existe, el Sistema anuncia "Registro Nuevo". Si Usted ha cometido un error y no quiere ingresar un Nuevo Registro, presione la tecla <ESC> para rechazar el ingreso. Si sí desea ingresar un Registro Nuevo, proceda a llenar los valores de los campos de la pantalla. Presione <F2> cuando los valores sean los deseados.

- Para eliminar un Registro, **ingrese** el valor de su Clave Principal. Cuando el Registro se presente en la pantalla, presione la tecla <F7>. El Sistema confirma si realmente **Usted** desea realizar una eliminación.

- Se pueden presentar en pantalla el Registro Siguiente y el Registro Anterior al corriente, usando las teclas <PgUp> y <PgDown>.

- Las teclas <HOME> y <END> presentan siempre el Primer y Ultimo **Registro** respectivamente.

- Las teclas de **Flechas** y las teclas <ENTER> y <TAB> permiten mover el **cursor** a través de la pantalla.

F2	*Grabar un Registro Nuevo/Modificado en la Base de Datos.*
F7	*Eliminar un Registro de la Base de Datos.*
PgUp	*Ver el Registro Previo.*
PgDown	*Ver el Registro Siguiente.*
Home	*Ver el Primer Registro.*
End	*Ver el Ultimo Registro.*
Esc	*(Si el Registro está visible)* *Rechazar ingreso/cambio.* *Limpiar la Pantalla.* *(Si la Pantalla está vacía)* *Salir del Módulo.*
Ins	*Pasar la Modo de Inserción.*
Tab	*Mover el cursor al siguiente campo.*
Enter	*Mover el cursor al siguiente campo.*
Flecha Abajo	*Mover el cursor al siguiente campo.*
Flecha Arriba	*Mover el cursor al campo anterior.*
Flecha Derec.	*Mover el cursor un caracter a la Derecha.*
Flecha Izq.	*Mover el cursor un caracter a la Izquierda.*
BackSpace <-	*Eliminar el caracter a la izq. del cursor.*
Del	*Eliminar el caracter sobre el cursor.*

Figura 43. Resumen de las funciones de teclas. Módulos de: Fuentes, Tipos de Documentos, Idiomas y Tópicos

- Con la tecla <INS> se puede **pasar** al Modo de Inserción, típico de los editores de textos. Fresionando nuevamente <INS> se regresa al Modo de Sobreescritura.

- La tecla <DEL> elimina el caracter sobre el cursor y la tecla <BackSpace> elimina el caracter a la izquierda del cursor.

- Para abandonar estos Módulos se presiona <ESC>. cuando la pantalla esta vacia.

La Figura **43** presenta un resumen de las funciones de las teclas para los Módulos de Fuentes, Tipos de Documentos, Idiomas y Tópicos.

En el caso particular del **MÓDULO DE TOPICOS.** tome en cuento los siguientes puntos:

- No se deben ingresar más de 19 Tópicos en el Sistema.
- Cada Tópico debe tener un Código diferente. No se permiten Códigos repetidos.
- Una vez asignado un Código a un Tópico. NO lo modifique.
- Eliminar un Tópico del Sistema significa eliminar de la Base de Datos todos los Documentos que han sido ingresados sobre ese Tópico. Cuando Usted presione la tecla <F7> para eliminar un Tópico, el Sistema le

advertirá esta situación y le pedirá que confirme la orden.

- El campo de "Numero de Busquedas Grabadas" es solo de lectura. Usted no puede alterar su valor. Este campo le indica cuántas Búsquedas por Falabra, realizadas por Usted, han sido grabadas por el Sistema (Ver Módulo de Busqueda de Busyueda por Palabra).

C.2. MODULO DE DOCUMENTOS (MENU PRINCIPAL).-

Escoja esta opción **para:**

a. Ingresar los datos de un nuevo Documento.

b. Modificar los datos de un Documento ya existente.

c. Consultar los datos de los Documentos existentes.

d. Eliminar un Documento.

Al presionar la Opción ´Documentos. del Menu Principal, se presenta a Usted un Menú Horizontal con las opciones de: Tópico, Ingreso, Actualización/Consulta, Eliminación y Salir. El Sistema no le permite ingresar a los Módulos de Ingreso, Actualización/Consulta y Eliminación de Documentos si no se ha determinado ántes el nombre del Tópico al que pertenecen los Documentos con los que Usted quiere trabajar.

C.2.1 OPCION TOPICO (MENU DE DOCUMENTOS).-

Al escoger esta Opción aparece en la pantalla una Lista Vertical con los nombres de todos los Tópicos existentes en el Sistema. Estos son los Tópicos que han sido ingresados mediante el Módulo de Tópicos del Menú de Mantenimiento.

- Utilice las Flechas Arriba y Abajo para recorrer la Lista de Topicos.

- Una pequeña Ventana ubicada al lado derecho de la Li&>~da Tópicos le comunica si existe ya un Tópico escogido, y de ser así el nombre de ese Tópico.

- Para escoger por primera vez un Tópico o para cambiar el Tópico actual por otro, posiciónese sobre el nombre del Tópico deseado y presione la tecla <ENTER>. En ese momento la Ventanita de la derecha actualizará el nombre del Tópico actual.

- Una vez escogido el Tópico con el que Ud. desea trabajar, presione una tecla, para abandonar la Lista de Tópicos y regresar al Menu Horizontal de Documentos.

Lucqo de escogido un Tópico, Usted puede proceder a Ingresar, Actualizar, Consultar o Eliminar los Documentos referentes a él. A diferencia de los Módulos de Fuentes, Idiomas, Tipos de Documentos y Tópicos, en los cuales todas estas operaciones se realizan por medio de una sóla pantalla, para los Documentos existen tres pantallas independientes: una para Ingreso, otra para Actualización y Consulta, y la tercera para Eliminación. Las utilización de las teclas presentadas en la Figura 43 sigue siendo válida, aunque esta vez su uso se distribuye entre las tres pantallas.

C.2.2 OPCION DE INGRESO (MENU DE DOCUMENTOS).-

El Módulo de Ingreso presenta un constante diálogo entre el Sistema y Usted. Una serie de Ventanas le asisten constantemente y le permiten la correcta ejecución del Proceso de Ingreso de Documentos. Este Proceso Comprende los siguientes pasos:

- Se presenta a Usted una pantalla con los campos de: Código del Documento, Título, Autor, Código de la Fuente, Año y Mes de Publicación, Idioma, Número de Páginas y Referencia. Asigne un Código al Documento

que desea ingresar y escríbalo en el primer Campo de la pantalla.

- Si el Código del Documento no existia previamente en el Sistema aparece en la parte inferior de la pantalla el rnensaje "Registro Nuevo". Proceda a ingresar valores en el resto de campos y recórralos hasta estar satisfecho de su contenido.

- Para facilitar el ingreso de datos en los campos de Código de Fuente y Código de Idioma, el Sistema provee una Ventana de Consulta para cada uno de estos campos. Posiciónese en uno de los campos nombrados y presione la tecla <F10>. Aparece una Ventana donde Usted puede observar los Códigos de Fuentes o de Idiomas ya existentes en el Sistema y sus nombres respectivos. Las teclas <PageUp> y <PageDown> le permiten recorrer el contenido de la Ventana y la tecla <ESC> abandonar la Ventana. De no ser deseado ninguno de los Códigos existentes, llene los campos del Documento con Códigos nuevos de Fuentes o Idiomas.

- Para el Campo de Autor guarde coherencia, ingrese siempre el Apellido del Autor Antes que su Nombre.

- El campo de Mes de Publicación debe llenarse con la Nomenclatura: 1 (01), 2 (02), ... 12.

- Luego de ingresar el Primer Bloque de datos, presione la tecla <F5>. El sistema pregunta sucesivamente si el Documento ingresado es un "Articulo de Revista o Periódico", "Capítulo de Libro". "Libro o Tomo de Enciclopedia". "Proyecto o Tesis" o "Seminario o Conferencia". Responda "S" (SI) o "N" (NO) a cada pregunta. Al responder afirmativamente a un determinado Tipo de Documento, el Código de ese Tipo es asignado al Documento. Dependiendo del Tipo establecido aparecen en la pantalla diversos campos ha ser llenados.

- Una vez ingresados valores en los campos alternativos. presione nuevamente la tecla <F5>. De esta manera se pasa a la Segunda Pantalla de Ingreso, que contiene los campos para Ingreso de: Palabras Claves y Resume'). Se llaman "Palabras Claves" a ***palabras o frases que representan el contenido del Documento.*** Ejs: LENGUAJE DE PROGRAMACIÓN, TURBO C, MODELO OSI, RED LOCAL. etc.

- Todas las Palabras Claves deben estar separadas por comas (,). NO separe palabras. mediante guiones (-), al final de las líneas. Una vez ingresadas las Palabras Claves. presione la tecla <F5> para pasar a ingresar un Resumen del Documento.

- Luego de ingresado el campo de Resumen presione nuevamente la tecla <F5>. En este momento Usted ya no podrá regresar a la Primera Pantalla de Ingreso, ni tampoco alterar ninguno de los valores ingresados en la Segunda Pantalla. Si desea hacer algún cambio al Documento actual, grábelo ahora y posteriormente. mediante la Opcion Actualización/Consulta del Menú Horizontal de Documentos, haga los cambios deseados.

- El Sistema le preguntará si se desea grabar el Documento.

- En caso de que Usted haya ingresado valores nuevos de Fuentes, Idiomas o Tipos de Documentos. el Sistema se lo notifica y le indica que para ingresar los datos restantes de las Fuentes. Idiomas o Tipos de Documentos debe recurrir a los Módulos respectivos.

C.2.3 OPCION DE ACTUALIZACION Y CONSULTA (MENU DE DOCUMENTOS).-

- La utilización de las teclas: <PgUp>, <PgDown>, <Home>, <End>, <F2> y <ESC> es similar a la que explicamos para los Módulos de Fuentes, Tipos de Documentos, Idiomas y Tópicos.

- El Sistema provee, al igual que para el Ingreso, dos Pantallas para Actualización y Consulta. La Primera Pantalla presenta los 13 primeros campos del Registro de Documentos y la Segunda, los campos de Palabras Claves y Resumen. Esta Segunda Pantalla es llamada mediante la tecla <F5>.

- Una vez en la Segunda Pantalla, la tecla <F5> le permite pasar del campo de Palabras Claves al campo de Resumen. Para regresar a la Primera Pantalla vuelva a presionar la tecla <F5>.

- Existen Ventanas de Consulta para los campos de Fuentes, Tipos de Documentos e Idiomas. Estas Ventanas son similares a las explicadas para el Módulo de Ingreso de Documentos.

- Cuando actualice el campo del Código del Tipo de Documento no olvide que su valor. está restringido a cinco únicas posibilidades (AR. LE, CL, PT y SC) predefinidas por el SPMD.

C.2.4 OPCION DE ELIMINRCION (MENU DE DOCUMENTOS).-

- Ingrese el Código del Documento a eliminar. L o datos del Documento son presentados en la pantalla. pero no se permite movilización a través de ellos.

- Presione la tecla <F5> para pasar a la Pantalla de Palabras Claves y Resumen. Presionar nuevamente la tecla <F5> lo regresa a la pantalla original.

- La tecla <F7> realiza la eliminacion. pidiendo su confirmaci6n.

C.3. MODULO DE BUSQUEDAS (MENU PRINCIPAL)--

La Opción de Busquedas del Menú Principal le permite a Usted encontrar una serie de Documentos que satisfagan su necesidad de información.

Al escoger esta Opción se presenta en pantalla un Menú Horizontal con las siguientes Opciones: Tópico, Búsqueda por Palabra y Busqueda por Condicionantes y Salir. Usted no podrá realizar ningun tipo de Búsqueda, si no se ha establecido antes en el Sistema un Tópico Corriente.

C.3.1 OPCION TOPICO (MENU DE BUSQUEDAS).-

Esta opción es totalmente equivalente a la Opción Tbpico del Menu Horizontal de Documentos. Utilicela para escoger por primera vez un Tópico o cambiar el Tópico actual. Todas las Búsquedas se realizan entre los Documentos pertenecientes al Tópico actual.

C.3.2 OPCION DE BUSQUEDA POR PALABRA CLAVE O DEL TITULO (MENU DE BUSQUEDAS).-

El Módulo de Buisqueda por Palabras presenta una pantalla con un unico campo. En él puede Usted ingresar una palabra o frase en base a la cual se realizara la Búsqueda. La palabra que Usted ingrese puede formar parte del Titulo de algún Documento que desee recuperar. También puede tratarse de una "Palabra Clave" que represente la información que Usted necesita. Ejs: LENGUAJE DE PROGRAMACIÓN, BASES DE DATOS. LAN. PASCAL. etc. El Sistema decide qué

Documentos estan relacionados directa e indirectamente con la palabra que Usted ingresa y los recupera.

El Proceso para efectuar una Busqueda es el siguiente:

- Ingrese una palabra o frase en el único **campo** presentado en la pantalla. Si presiona la tecla <ESC> luego de haber ingresado algun caracter, el campo se limpia. Si el campo está limpio y se presiona <ESC> se abandona el Módulo. Cuando Usted esté satisfecho con la palabra o frase ingresada, presione la tecla <F5>, y se dará inicio el proceso de Búsqueda.

- Luego de presionada la tecla <F5>, aparece en la pantalla una nueva Ventana, presentando dos Opciones al usuario. La primera Opción es: Realizar todo el Proceso de Búsqueda y **la** segunda: Ver la Respuesta Grabada por el Sistema en una Búsqueda Previa de esa Palabra. El sentido de estas dos Opciones se basa en el hecho de que el **SPMD** acostumbra mantener grabadas un conjunto máximo de quince Respuestas a Búsquedas realizadas.

- Si el Sistema tiene grabada la Respuesta de la Búsqueda basada en **la** palabra que Usted acaba de ingresar,

esccger la Opción 2 le permite obtener los resultados mas rápidamente, pues no se tiene que rsalizar todo el Proceso interno de Busqueda.

- Si al contrario, Usted es conciente que se han ingresado nuevos Documentos al Sistema y que de existir una Respuesta Grabada de la Búsqueda que quiere realizar, ésta debe estar muy desactualizada, es mejor escoger la Opción 1. En este caso, el Sistema realiza todo el Proceso de Búsqueda y graba la nueva Respuesta.

- Si alguna vez Usted escoge la Opción 2 esperando ver la Respuesta Grabada de una Búsqueda y resulta que ésta no se encuentra grabada, no se preocupe pues el Sistema realizara todo el Proceso de Búsqueda Y grabará la Respuesta.

- Luego de un tiempo de escogida una de las dos Opciones men-ionadas, se presentan en la pantalla los resultados de la Búsqueda.

- En el caso de que el Sistema haya realizado todo el Proceso de Busqueda se presentan a Usted, además de la Palabra Original de Busqueda, dos Ventanas Superiores.

La Ventana de la Izquierda contiene las palabras que el Sistema, a lo largo de la Búsqueda, relacionó a la palabra ingresada por Usted. La Ventana de la Derecha presenta: los Códigos de los Documentos relacionadoe directa e indirectamente con la palabra objeto de la Búsqueda, y los 30 primeros caracteres de sus respectivos Titulos. No se realiza distinción entre Documertos Directa e Indirectamente relacionados.

- Si el usuario ha pedido ver la Respuesta a una Búsqueda Anterior. sólo se presenta a el. además de la Palabra Original de Búsqueda, una Ventana Superior con los Códigos y Titulos de los Documentos seleccionados.

- Una vez posicionado en una Ventana. las teclas <Page Up> y <PageDown> permiten recorrer las Palabras Relacionadas o los Documentos Seleccionados pertenecientes a esa Ventana. En caso de existir dos ventanas, la tecla <TAB> permite pasar alternativamente de una de las ventanas a la otra.

- Estando posicionado en cualquier Ventana Superior se puede presionar la tecla <F10>. En eete momento aparece en la parte Inferior de la Pantalla una Ventana

de Consulta con los siguientes campos: un campo de ingreso del Código de un Documento y campos sólo de lectura para Autor. Titulo. Código de la Fuente. Referencia. Año y Mes de Publicación del Documento.

- Mediante esta Ventana de Consulta Usted puede ingresar el Código de uno de los Documentos seleccionados y así conocer los respectivos datos de Titulo Completo, Autor, Código de Fuente, Referencia. Año y Mes de Publicación del Documento. Estando en esta Ventana la tecla <ESC> limpia los campos y permite el ingreso de otro Código de Documento. Si la ventana está limpia y se presiona <ESC>, se abandona la Ventana de Consulta y se regresa a una de las Ventanas Superiores.

- Posicionado en cualquier Ventana Superior, Usted puede presionar la tecla <F7> para imprimir los resultados de la Búsqueda. El Sistema crea un Reporte con los siguientes datos de los Documentos Seleccionados: Código del Documento, Título, Autor y Código de Fuente, además de la Palabra que originó la Búsqueda.

- El Sistema le **pregunta** si desea utilizar Papel Continuo o Individual, y en el segundo caso, si éste es

de Tamaño Oficio o Carta. Existe siempre la opción de retroceder mediante la tecla <ESC>.

- La impresión se realiza siempre por medio del puerto LPT1, con letra condensada y a 80 columnas.

- Una última Ventana recuerda a Ud. encender su impresora y presionar el botón "ON LINE". Tiene la oportunidad de cancelar la impresión o seguir adelante.

- En caso de que el Sistema encuentre problemas en la impresión, ya sea porque la impresora no esté encendida o no se encuentre en línea, él se lo notifica. Usted tiene la oportunidad de solucionar el problema y realizar la impresión, o cancelarla.

- Una vez terminada la impresión, se retorna a una de las Ventanas Superiores.

- Si durante la impresión de un Reporte, Usted apaga la impresora, el Sistema cancela la impresión y espera una nueva orden.

- Estando en cualquiera de las dos Ventanas Superiores, la tecla <ESC> permite regresar a la Pantalla de Ingreso de la Palabra de Búsqueda.

c.3.3 OPCION DE BUSQUEDA POR CONDICIONANTES (MENU DE BUSQUEDAS).-

El Módulo de Busqueda por Condicionantes proporciona a Ud. la posibilidad de plantear una amplia gama de Condiciones y Restricciones a los Documentos que se quieren recuperar. El Sistema encontrará y presentará a Usted los Documentos que cumplan con todas las Condiciones planteadas.

La Pantalla de Busqueda por Condicionantes está compuesta de diversos campos. Cada campo es una Condición que deben cumplir los Documentos para ser recuperados. A continuación se presenta una descripción del Proceso de Búsqueda por Condicionantes:

- Los tres primeros campos pertenecen a la 'Condición de Palabra' y son semejantes al Módulo de Búsqueda por Palabra donde, dada una palabra o frase se obtenían los Documentos relacionados directa o indirectamente a ella. Esta vez Usted puede ingresar hasta tres palabras, ampliando asi el campo de Selección. . Estas tres

palabras están relacionadas entre sí por la operación lógica 'O'. Cumplirán la Condición Triple de Palabra todos los Documentos que estén relacionados directa o indirectamente ***con*** por lo menos UNA DE LAS TRES posibles palabras a ingresar.

- La 'Condición de Autor' permite a Usted recuperar Documentos escritos por una determinada persona. Guarde coherencia con la manera en que ingresó el campo de Autor de los Documentos. Acostúmbrece a escribir primero el Apellido y luego el Nombre.

- Si Usted no recuerda exactamente el Apellido o el Nombre de un Autor, la 'Condición de Autor' permite utilizar como "comodín" uno o más ***"*" al final de la cadena.*** Ej: cumplen la Condición de Autor "STE*" todos los Documentos cuya cadena de Autor empieza con las letras "STE". Si no se coloca un "*" al final de la cadena de Autor, se asume que ésta está completa y se buscan los Documentos que cumplen exactamente con el Autor ingresado.

- El Sistema no permite utilizar asteriscos ("comodín") en una posición inicial o intermedia de la 'Condición de Autor', sino sólamente al final de la cadena.

- Para facilitar el ingreso de las Condiciones de: Código de Fuente, Código de Idioma, y Código de Tipo de Documento, la tecla <F10> provee Ventanas de Consulta para cada uno de estos campos. Estas Ventanas son semejantes a las explicadas para los Módulos de Ingreso y Actualización de Documentos.

- Si Usted desea recuperar Documentos pertenecientes a una determinada Editorial, Revista o Institución, llene la Condición de Código de Fuente.

- Si Usted desea recuperar Documentos escritos en un determinado Idioma, llene la Condición de Código de Idioma.

- Si Usted desea recuperar Documentos de un determinado Tipo, entre los cinco definidos por el Sistema, llene la Condición de Código de Tipo de Documento.

Usted puede recuperar Documentos que hayan sido publicados en un determinado lapso de tiempo, especificado en meses y años. Existen Condiciones para: Mes Inicial, Mes Final, Año Inicial. y Año Final de Publicación. LLenando uno u otro de estos campos se pueden establecer periódos de tiempo durante los cuales deben haber sido publicados los Documentos a ser seleccionados. Ej: **Para** especificar un periodo compuesto por un único **Mes** (Agosto) de un determinado Año (1990) llene los campos de Mes Inicial y Mes Final con el valor '08' y los campos de Año Inicial y Año Final con el valor '1990'.

- Los **campos** que Usted deje vacios de la Pantalla de Condicionantes significan que no existe ninguna Restricción **para el valor** de ese campo.

- Si Usted desea limpiar los valores de todas las Condiciones ingresadas, presione la tecla <ESC>. Si todos los campos estan limpios y se presiona **la** tecla <ESC>. Usted abandona el Módulo. Si está satisfecho de las condiciones ingresadas y quiere dar inicio al proceso de Búsqueda, presione la tecla <F5>.

- Si no existen Documentos que **satisfagan** todas las Condiciones ingresadas, el Sistema se lo notifica.

- De existir una respuesta a la Búsqueda, el Sistema presenta una única Ventana Superior con los Códigos de los Documentos recuperados y los primeros 30 caracteres de sus **respectivos** Títulos.

- Posicionado en esta Ventana, utilice las teclas <Page Up> y <PageDown> para recorrer los Documentos Seleccionados.

- De igual manera que **para** la Búsqueda por Palabra, la tecla <F10> presenta a Usted una Ventana Inferior de Consulta donde puede ingresar el Código de uno de los Documentos Seleccionados y así conocer los datos de Título Completo, Autor, Código de Fuente, Referencia, Mes y Año de Publicación.

- Estando en la Ventana de Consulta, la tecla <ESC> limpia los campos y permite el ingreso de otro Código de Documento. Si la Ventana está limpia y se presiona <ESC>, se regresa a la Ventana de Documentos Seleccionados.

- Posicionado en la Ventana Superior. Usted puede presionar la tecla <F7> para imprimir los resultados de la Búsqueda realizada. El Sistema crea un Reporte con los siguientes datos de los Documentos Seleccionados: Código del Documento, Título, Autor y Código de Fuente, además de presentar todos los Condicionantes que originaron la Búsqueda.

- El Sistema le pregunta si desea utilizar Papel Continuo o Individual, y en el segundo caso, si éste es de Tamaño Oficio o Carta. Existe siempre la opción de retroceder mediante la tecla <ESC>.

- La impresión se realiza siempre por medio del puerto LPT1, con letra condensada y a 80 columnas.

- Una ultima Ventana recuerda a Ud. encender su impresora y presionar el botón "ON LINE". Tiene la oportunidad de cancelar la impresión o seguir adelante.

- En caso de que el Sistema encuentre problemas en la impresión, ya sea porque la impresora no esté encencida o no se encuentre en línea, él se lo notifica.

Usted tiene la oportunidad de solucionar el problema y realizar la imyresion, o cancelarla.

- Una vez terminada la impresión, se retorna a la Ventana Superior.

- Si durante la impresión de un Reporte, Usted apaga la impresora, el Sistema cancela la impresión y espera una nueva orden.

- Si estando en la Ventana Superior Ud. presiona la tecla <ESC>, regresa a la Pantalla de Condicionantes.

C.4. MODULO DE GENERACION DE REPORTES (MENU PRINCIPAL).-

El Módulo de Generación de Reportes permite a Usted imprimir el contenido de los Archivos de: Documentos, Tipos de Documentos, Fuentes, Idiomas y Tópicos.

A continuación las características de evte Módulo:

- De cada Documento se imprimen únicamente los siguientes datos: Código del Documento, Título, Autor y Código de la Fuente.

- De cada Fuente se imprimen únicamente los siguientes datos: Código de la Fuente. Nombre, Ciudad, Teléfono y Código Postal.

- Para los Tipos de Documentos, Idiomas y Tópicos se imprimen los datos de: Código y Nombre.

- Para imprimir un Archivo de Documentos, el Sistema le pide especificar el nombre del Tópico al que éstos pertenecen.

- Para los Reportes de Documentos, Usted puede decidir el campo por el cual éstos se ordenan (ascendentemente) : Código del Documento, Autor o Código de Fuente.

- Los Reportes de Fuentes, Tipos de Documentos, Idiomas y Tópicos son siempre ordenados (ascendentemente) por el campo de Código.

- El Sistema le pregunta si desea utilizar Papel Continuo o Individual, y en el segundo caso, si éste es de Tamaño Oficio o Carta. Existe siempre la opción de retroceder mediante la tecla <ESC>.

- El Sistema presenta a Usted una Ventana resumiendo las características de la impresión: número del puerto. tipo de papel, tamaño del papel, tipo de letra y número de columnas.

- La impresión se realiza siempre por medio del puerto LPT1, con letra condensada y a 80 columnas.

- Una 61tima Ventana recuerda a Ud. encender su impresora y presionar el botón "ON LINE". Tiene la oportunidad de cancelar la impresión o seguir adelante..

- En caso de que el Sisterna encuentre problemas en la impresión, ya sea porque la impresora no esté encencida o no se encuentre en línea, él se lo notifica. Usted tiene la oportunidad de solucionar el problema y realizar la impresión, o cancelarla.

- Si durante la impresión de un Reporte, Usted apaga la impresora, el Sistema cancela la impresión y espera una nueva orden.

C.5. MODULO DE MANTENIMIENTO (MENU PRINCIPAL).-

Al escogerse la Opción Mantenimiento del Menu Principal se presenta a Usted un Menú Vertical con las Opciones de: Actualizar Códigos, Reindexar Archivos, Vaciar Archivos, Crear la Rase de Datos, y Salir.

La opción 'Actualizar Códigos' presenta a su vez los Módulos de Tópicos, Idiomas y Tipos de Documentos, que fueron previamente explicados, junto con el Módulo de Fuentes del Menú Principal. A continuación explicaremos las opciones restantes.

C.5.1 MODULO DE REINDEXACION DE ARCHIVOS (MENU DE MANTENIMIENTO).-

El Módulo de Reindexacion de Archivos permite crear Indices nuevos para los Archivos de: Documentos, Tipos de Documentos, Fuentea, Idiomas y Tópicos.

Reindexe un Archivo en cualquiera de estos casos:

1. El Archivo tiene Indices, pero estos están "corruptos" o "dañados". Keindexar el Archivo crea Indices nuevos en base a los registros existentes en el respectivo Archivo.

2. El Archivo NO tiene Indices. estos han desaparecido de la Base de Datos. Reindexar el Archivo **crea** Indices nuevos en **base** a los registros existentes en el respectivo Archivo.

Tome en cuenta los siguientes puntos:

- Para Reindexar un Archivo de Documentos. el Sistema le pide especificar el nombre del Tópico al que éstos pertenecen.

- Al Reindexar un Archivo de Documentos, automáticamente el Sistema reindexa una serie de Archivos relacionados a él y de cuya existencia Usted no se debe preocupar.

C.5.2 MODULO **DE INICIALIZACION DE ARCHIVOS (MENU DE MANTENIMIENTO).-**

El Módulo de Inicialización de Archivos permite vaciar el contenido de los Archivos de: Documentos, Tipos de Documentos, Fuentes, Idiomas y Tópicos.

Inicialize un Archivo en cualquiera de estos casos:

1. Usted desea eliminar TODOS los registros contenidos en un Archivo. Inicializar el Archivo lo **vacía** totalmente y **crea** Indices nuevos y vacíos para él. .

2. El Archivo de Fuentes, Tiyos de Documentos, Idiomas o Tópicos ha desaparecido de la Rave de Datos. Inicializar uno de estos Archivos desaparecidos crea uno . **nuevo** *y* **vacio,** además de sus Indices también nuevos y vacíos.

- El Archivo de Documentos, de uno de los Tópicos definidos, ha **desaparecido** de la Rase de Datos. Inicializar uno de estos Archivos desaparecidos crea uno **nuevo y vacio.** además de sus Indices también nuevos y vacíos.

Tome en cuenta los siguientes puntos:

- **Para** Inicializar un Archivo de Documentos, el Sistema le pide especificar el nombre del Tópico al que éstos pertenecen.

- Al Inicializar un Archivo de Documentos, automáticamente el Sistema inicializa una serie de Archivos relacionados a **el** y de cuya existencia Usted no se debe preocupar.

- Ya que inicializar un Archivo por equivocación puede ser un error irreparable, el Sistema advierte varias

veces a Usted que va a proceder a eliminar el contenido del Archivo y **pide** su confirmación.

- Antes de inicializar un Archivo, éste es respaldado con la extensión ".OLD".

- Para retomar un Archivo ".OLD". Usted debe ejecutar ciertos comandos desde **el** Prompt del Sistema Operativo Siga los siguientes pasos:

1. Para los Archivos de Fuentes, Tipos de Documentos, Idiomas y Tópicos ejecute respectivamente uno de los siguientes comandos:

   C:\ ren FUENTES.OLD FUENTES.DAT

   C:\ ren TIPOS.OLD TIPOS.DAT

   C:\ ren IDIOMAS.OLD IDIOMAS.DAT

   C:\ ren TOPICOS.OLD TOPICOS.DAT

2. Al realizar la inicialización de un Archivo de Documentos, el Sistema **crea** dos Archivos de respaldo con extensión ".OLD". Ej: **para** un Tópico con nombre "COMPUTACION", existirAn los **Archivos** "COMP.OLD" **y** "COMP_PAL.OLD". El segundo Archivo

nombrado es manejado por el "Sistema Personal de Manejo Documentos" y Usted no debe preocuparse mayormente por él. Construya un comando ***rename*** del DOS, tomando las cuatro primeras letras del nombre del Tópico, cuyo Archivo de Documentos respaldado desea retomar. Para nuestro ejemplo, ejecute el siguiente comando:

C:\ ren COMP*.OLD COMP*.DAT

3. **Para** cualquiera de los dos casos superiores, luego de ejecutado el comando ***rename*** desde el Prompt del Sistema Operativo, ingrese al "Sistema Personal de Manejo de Documentos". Proceda inmediatamente a reindexar el Archivo recuperado, mediante la opción 'Reindexar Archivos' del Menú de Mantenimiento.

- El Sistema no permite la Inicialización del Archivo de Tópicos, pues esto implicaría borrar TODOS los Documentos, de TODOS los Tópicos del sistema. Sistema le pide a Usted que recurra al Módulo de Tópicos para, de quererlo, eliminar uno a uno los Tópicos del Sistema y sus respectivos Documentos asociados.

C.5.3 MODULO DE CREACION DE LA BASE DE DATOS (MENU DE MANTENIMIENTO).-

Este Módulo es utilizado para construir por primera vez una Base de Datos para el Sistema. La Base de Datos queda conformada originalmente por los Archivos: TIPOS.DAT, FUENTES.DAT, IDIOMAS.DAT, TOPICOS.DAT y sus Indices respectivos. Los Indices de la Base de Datos tienen siempre por extensión: ".X01", ".X02" o ".X03".

Como al momento de la Creación de la Base de Datos no existe ningún Tópico definido por el usuario, no existe tampoco ningún Archivo de Documentos. Al ingresar un Tópico al Sistema, mediante el Módulo de Tópicos, es cuando se crean para ese Tópico: un Archivo de Documentos y sus Indices respectivos. Los Archivos de Documentos toman por nombre las primeras 4 letras del Tópico al cual pertenecen y la extensión ".DAT". Ejs: para el Tópico "COMPUTACION", el Archivo de Documentos es "COMP.DAT": para el Tópico "MEDICINA", el Archivo de Documentos es "MEDI.DAT", etc.

Si Usted ha ingresado al Módulo de Creación de Base de Datos y ha ordenado Crear e Inicializar TODOS los Archivos que la conforman, el Sistema reviza la existencia en su directorio corriente de Archivos con extensión "*.DAT" y

"*.XO*". Si existen Archivos de este tipo, el Sistema alerta la posible existencia de una **Rase** de Datos con Archivos de Datos y de Indices. Usted tiene que decidir si quiere mantener estos Archivos existentes o. eliminarlos y crear una Base de Datos nueva.

D. RESPALDO DE LA BASE DE DATOS.-

Cuando Usted desee respaldar su Base de Datos copie en un diskette TODOS los Archivos del Directorio Corriente del "Sistema Personal de Manejo de Documentos" que tengan las siguientes extensiones:

.DAT , *.X* , *.TXT y *.BQ1

Los Archivos "*.DAT" **son** Archivos de Datos. Un subconjunto de ellos son los Archivos cuyo nombre cumple con el patrón "*_PAL.DAT", los cuales son utilizados por el Sistema en los **Procesos de** Busqueda.

Los Archivos "*.X*" son Archivos de Indices.

Finalmente. los **Archivos** "4.TXT" y "*.BQ1". son también utilizados por el Sistema para la ejecución de los Procesos de Búsqueda.

E. CODIGOS DE ERROR.-

A continuación se detallan los 2 más comunes Mensajes de Error que el Sistema puede presentar cuando encuentra problemas en su ejecución, además de la manera de solucionar esos problemas.

MENSAJE: "Falta Archivo DIC.DOC (DICCIONARIO)"

El Sistema maneja un Diccionario compuesto por un conjunto de preposiciones, conjunciones y artículos, tanto en Español como en Inglés y lo utiliza en la ejecución de los Módulos de Documentos y Búsquedas.

Si el Archivo DIC.DOC ha desaparecido de su Base de Datos, pida ayuda a su Técnico.

MENSAJE: "Indice Corrupto"

Solución: Reindexe el Archivo mediante el Módulo de Reindexación de Archivos en el Menú de Mantenimiento.

CAPITULO IV

FUNDAMENTOS DE LENGUAJE C++

4.1 LA PROGRAMACIÓN ORIENTADA A OBJETO (POO).-

La Programación Orientada a Objeto (POO) es una nueva manera de abordar la tarea de la programación [SCHI90].

Desde la invención del computador, los esquemas usados para enfrentar la programacion han ido cambiado paulatinamente, siempre en busca de satisfacer la creciente complejidad de los programas. Cuando las computadoras fueron recién desarrolladas, la programación se realizaba switcheando Instrucciones Binarias de Máquina en un panel frontal. Luego los programas crecieron en tamaño y complejidad y se invent0 el Lenguaje Ensamblador. Por medio de él, el programador pudo manejar los programas usando representaciones simbólicas de la instrucciones de máquina. Eventualmente re introdujeron los Lenguajes de Alto Nivel que ofrecieron al programador una serie de herramientas nuevas.

En 1960 nació la Programación Estructurada, que tiene como abanderados principales a los Lenguajes C y PASCAL. Gracias al lenguaje estructurado fue posible escribir programas moderadamente complejos con cierta facilidad.

Sin embargo, hoy en dia. el tamaño y la complejidad de muchos proyectos estan en el punto en el que los métodos estructurados no trabajan más. Para solucionar este problema se invent6 la Programación Orientada a Objeto.

La POO toma las mejores ideas de la Programación Estructurada y las combina con conceptos nuevos y poderosos que presentan una nueva visión de la programación. El nuevo enfoque permite descomponer fácilmente un problema en subgrupos de partes relacionadas. Estos subgrupos se convierten en unidades independientes llamadas Objetos. Todos los lenguajes de POO tienen tres características comunes: Objetos, Polimorfismo y Herencia.

4.1.1 OBJETOS.-

La mas importante característica de un Lenguaje Orientado A Objeto es la definición de ***Objeto.*** Un Objeto es una entidad lógica que contiene datos. y código que manipula esos datos [SCHI90]. Dentro de un Objeto, algo del código o datos puede ser privado, inaccesible por cualquier cosa fuera del Objeto. De esta manera, un Objeto provee un nivel eignificativo de protección contra modificaciones accidentales o usos incorrectos. Este enlace entre código y datos es llamado comunmente ***Encapsulación.***

Un Objeto constituye una variable, cuyo tipo es definido por el usuario. Tal vez sea extraño pensar en un Objeto, que enlaza código y datos, como una variable. Sin embargo, éste es precisamente el caso en la POO. Cuando se define un Objeto se está implícitamente creando un nuevo tipo de dato.

4.1.2 POLIMORFISMO.-

El propósito del Polimorfismo es permitir que un nombre sea usado para especificar una clase general de acción. Dependiendo del tipo de dato con el que se está tratando, una específica instancia del caso general, es ejecutada. Por ejemplo, podemos tener un programa que define tres tipos diferentes de pilas: una para enteros, otra para valores float, y otra para longs. Si se crean tres conjuntos de funciones llamadas push() y pop(), el compilador seleccionará la rutina correcta dependiendo del tipo de dato del argumento con el que se esté llamando la función

4.1.3 HERENCIA.-

Herencia es el proceso por el cual un Objeto puede adquirir las propiedades de otro Objeto. Esto es importante, pues permite soportar el concepto de clasificación. La

mayoría del conocimiento es manejable gracias a clasificaciones jerárquicas. Ej: una Manzana Roja es parte de la clase Manzana, la cual a su vez es parte de la clase Fruta, la cual se encuentra en la clase Comida. Sin el uso de clasificaciones, cada Objeto tendría que definir todas sus características explicitamente. Usando clasificaciones, un Objeto sólo necesita definir aquellas cualidades que lo hacen único en su clase. Es el mecanismo de Herencia el que hace posible que un Objeto sea una instancia específica de un caso más general.

4.2 INTRODUCCION AL LENGUAJE C++.-

A pesar de su poderosa flexibilidad, cuando un proyecto excede cierto tamaño, el Lenguaje C alcanza su límite. En 1980, trabajando en los Laboratorios Bell en Murray Hill, New Yersey, Bjarne Stroustrup añadió una serie de extensiones al Lenguaje C y lo llamó "C con Clases". El nombre fue cambiado a "C++" en 1983.

Cuando invent6 el C++, Stroustrup sabia que era importante mantener el espiritu original del C: incluir su eficiercia, flexibilidad y la filosofía de que es el programador y no el lenguaje, quien lleva el mando. Al mismo tiempo, era necesario añadir soporte a la Programación

Orientada a Objeto. La meta fue alcanzada y hoy en día C++ provee a los programadores la libertad y el control del C y el poder de los Objetos. Las capacidades de Orientación a Objeto, según Stroustrup, permiten a los programas ser estructurados con claridad, ser extensibles y fáciles de mantener, sin perder la eficiencia.

4.2.1 ENTRADA Y SALIDA DE DATOS.-

C es considerado un subconjunto del C++. Por lo tanto, la mayoría de programas en C son también programas en C++. Sin embargo, escribir un programa en C++ que se vea como todo programa en C, es no aprovechar al máximo las capacidades del C++.

Algunas diferencias básicas entre C y C++ se aprecian en el siguiente ejemplo.

```
#include <iostream.h>

main (void)
{
      int i;
      char str[80];

      cout << "Bienvenidos al C++.\n";    // comentario

      cout << "Ingrese un Número:  ",
      cin  >> i;
      cout << "El Número es " << i << "\n";

      cout << "Ingrese una Cadena: ";
```

```
        cin  >> str;
        cout << "La Cadena es " << str << "\n";

        return 0;
I
```

El archivo ***iostream.h*** es definido en C++ y utilizado para el soporte de las operaciones de Entrada y Salida.

La yrimera linea del programa presenta en pantalla la frase: "Bienvenidos al C++". Se incluye al final de la cadena un caracter de retorno. En C++, el << tiene un rol expandido. Sigue siendo el operador de desplazamiento a la izquierda, pero usado como en el ejemplo, es también el Operador de Salida. La palabra ***cout*** es un identificador asociado a la pantalla. Se **usa** ***cout*** ***y*** << **para** presentar en pantalla cualquier tipo básico de datos y añadir a él cadenas de caracteres.

En C++ se pueden definir comentarios en dos maneras. Siguen teniendo validez los comentarios de C (/* .... */), pero existen también los ***comentarios de una línea (//).*** Cuando se inicia un comentario usando //. todo lo que sigue hasta terminar **la linea.** es ignorado por el compilador.

El programa de ej. pide al usuario que ingrese un número. El número es leído desde el teclado con el comando:

cin >> i;

El operador >> retiene su significado de desplazamiento a la derecha, pero usado como en el ejemplo, asigna a *i* el valor leído del teclado. El identificador ***cin*** está asociado al teclado. En general, usando ***cin y*** >> se puede asignar a una variable cualquiera de los tipos básicos de datos o cadenas.

Incluyendo tambien el archivo *stdio.h*, se podrían utilizar las funciones ***scanf() y printf()***, sin embargo ***cin >> y cout <<*** son mas propias del estilo de C++.

4.2.2 CLASES.-

Para crear un Objeto en C++, se debe definir primeramente su forma general, mediante la palabra reservada *CLASS*. Una Clase es sintácticamente similar a una estructura. La siguiente Clase define un tipo llamado **queue**, utilizado **para** crear una cola.

```
class queue {
      int q[100];
      int sloc, rloc;
public:
```

```
        void init(void);
        void qput(int i);
        int qget(void);
};
```

Una Clase puede contener partes privadas o públicas. Por omisión, todos los items definidos en una Clase son privados. En el ejemplo, los Datos **q**, **sloc** **y** **rloc** son privados, significando que no pueden ser accesados por ninguna función **que** no **sea** miembro de la Clase. Esta es una forma de Encapsulación, pues se puede controlar **el** acceso a ciertos items de datos manteniéndolos privados. También se pueden definir Funciones privadas, las cuales sólo pueden ser llamadas por otros miembros de la Clase.

Para hacer públicas partes de una Clase, hay que declararlas después de la palabra reservada ***public***. Todos los datos y funciones definidas como públicos son accesibles a todas las otras funciones del programa. Esencialmente, un programa accesa un Objeto a través de sus funciones públicas. Sin embargo, a pesar de que se pueden implementar los datos públicos, es mejor limitar o eliminar su uso. Todas los datos de una Clase deberían ser privados y controlar su acceso a través de funciones públicas.

Las funciones **init()**, **qput()** y **qget()** son llamadas ***funciones miembros***. porque son parte de la Clase **queue.** Sólo las funciones miembros tienen acceso a las partes privadas de la Clase en la que están declaradas.

Una vez definida una Clase, se puede crear un Objeto de ese tipo usando el nombre de la Clase. En esencia, el nombre de la Clase ***se*** convierte en un nuevo especificador de tipo de dato. Para **crear** un Objeto llamado cola1 de tipo **queue:**

```
queue cola1;
```

Es posible también asignar variables. **cuando** al definir una Clase se ponen los nombres de las variables después del corchete de **cierre,** exáctamente de la misma forma que para lac estructuras.

```
class Nombre-de-Clase {
     datos y funciones privadas
public:
     datos y funciones públicas
} lista de nombres de los Objetos;
```

Dentro de la declaración de **queue**, se deben usar los prototipos de las funciones miembros. En C++, cuando se necesita informar al compilador de una función, se debe utilizar **la** forma completa del prototipo.

Cuando llega el momento de codificar una función que es miembro **de** una **Clase**, hay que decirle al compilador a qué Clase pertenece la función.

```
void queue::qput(int i)
{
     if (sloc==100) {
          cout<<"la cola está llena";
          return;
     }
     sloc++;
     q[sloc]=i
}
```

El *::* es llamado *Operador de Alcance de Resolución.* Esencialmente, él le dice al compilador que esta versión de **qput()** pertenece a la Clase **queue.** En C++, diferentes Clases pueden utilizar los mismos nombres de funciones. El compilador sabrá a qué Clase pertenece cada función, basándose en el Operador de Alcance de Resolución y el nombre de la Clase.

Para llamar a una función miembro desde un lugar del programa que no es parte de la Clase, se usan el nombre del Objeto y el operador . (punto). Asi pues, **para** inicializar una cola de nombre **a:**

```
queue a,b;
a.init();
```

Es importante entender **que** a y **b** son Objetos diferentes. Inicializar a no inicializa **b**. La única relación entre **a** y **b** es que son Objetos de un mismo tipo.

Sólo cuando una función miembro es llamada por código que no pertenece a la Clase, se debe utilizar el nombre de la Clase y el operador "punto". En otro caso, una función miembro puede llamar a otra función miembro directamente, sin usar el operador "punto".

```
#include <iostream.h>

class queue {
     int q[100];
     int sloc, rloc;
public:
     void init(void);
     void qput(int i);
     int  qget(void);
3;

void queue::init(void)
{
  rloc=sloc=0;
3

void queue::qput(int i)
{
     if (sloc==100) {
          cout<<"La cola está 11ena";
          return:
     3
     sloc++;
     q[sloc]=i;
}

void queue::qget(void)
{
```

```
        if (rloc==sloc) {
            cout<<"La cola está vacía";
            return 0;
        }
        rloc++;
        return q[rloc];
}

main(void)
{
        queue a,b;

        a.init();
        b.init();
        a.qput(10);
        b.qput(19);
        a.qput(20);
        b.qput(1);
        cout << a.qget() << " ";
        cout << a.qget() << " ";
        cout << b.qget() << " ";
        cout << b.qget() << "\n";
        return 0;
}
```

4.2.3 SOBRECARGA DE FUNCIONES Y OPERADORES.-

A. SOBRECARGA DR FUNCIONES.-

Este es un mecanismo para conseguir polimorfismo. En C++, dos o más funciones pueden compartir el mismo nombre, siempre y cuando su declaración de parámetros sea diferente. Se dice que estas funciones estan "sobrecargadas".

```
#include <iostream.h>
int    sqr_it(int i);
double sqr_it(double d);
long   sqr_it(long l);

main(void)
```

```
f
     cout << sqr_it(10)   << "\n";
     cout << sqr_it(11.0) << "\n";
     cout << sqr_it(9L)   << "in":
     return 0;
J

int sqr_it(int i)
{
     cout <<   "Función  sqr_it() que  usa un argumento
               entero\n";
     return i*i;
}

double sqr_it(double d)
{
     cout <<   "Función  sqr_it() que  una un argumento
               double\n";
     return d*d;
}

double sqr_it(long l)
{
     cout <<   "Función  sqr_it() que  usa un argumento
               long\n";
     return l*l;
}
```

Este programa crea tres funciones llamadas **sqr_it()**, similares pero diferentes entre sí. Cada una retorna el cuadrado de su argumento. La importancia de las funciones sobrecargadas es que permiten conjuntos relacionados de funciones ser accesadas con un nombre común. En este sentido, la sobrecarga de funciones permite crear un nombre genérico para cada operación y deja en manos del compilador resolver qué función usar en cada caso, basándose en el tipo del **argumento.**

La sobrecarga de funciones ayuda a manejar la complejidad. Ej: Turbo C++ contiene en su librería las funciones **itoa(), ltoa()** *y* **utoa()**, que convierten diferentes tipos de números a sus cadenas equivalentes. A pesar de que las **tres** funciones llevan a cabo acciones idénticas, C utiliza tres nombres diferentes para representarlas. Esto hace la situación mas compleja de lo que realmente es. En C++ es posible usar un único nombre, como **numtoa()**, para nombrar las tres funciones y representar la acción general **que** se está llevando a **cabo**. Así, aplicando polimorfismo, **tree** cosas para recordar se reducen a sólo una.

B. SOBRECARGA DE OPERADORES.-

Esta constituye otra forma de polimorfismo. Como ya vimos, en C++ es posible usar los operadores << y >> para **llevar** a cabo operaciones de E/S. Esto se debe a que, en el archivo de cabecera ***iostream.h***, estos operadores están sobrecargados.

Cuando un operador está sobrecargado, él toma significados adicionales relativos a una cierta Clase, manteniendo además su signif**icado** original. Cada Clase define lo que un operador significa en relación a ella. EJ: Se puede sobrecargar **el** operador '+' en relación a los Objetos de

tipo queue, de manera que añada el contenido de una cola al contenido de otra. Sin embargo, el '+' sigue reteniendo su significado relativo a otros tipos de datos.

4.2.4 HERENCIA.-

En C++, la Herencia es soportada permitiendo que una Clase incorpore a otra Clase en su declaración.

Para los siguientes ejemplos, trataremos con la Clase vehiculo, que define vehículos que viajan por la carretera y almacena su numero de ruedas y el número de pasajeros que viajar en él.

```
class vehiculo {
     int ruedas;
     int pasajeros;
public:
     void set_ruedas (int num);
     int  get_ruedas (void);
     void set_pas (int num);
     int  get_pas (void);
};
```

Esta definición amplia de vehículo puede ser utilizada para definir otros Objetos mas especificos. A continuación se declara la Clase camion usando la Clase vehiculo.

```
class camión : public vehiculo {
     int carga;
public:
```

```
        void set_carga (int size);
        int  get_carga (void):
        void mostrar (void);
};
```

Se dice que **vehiculo** "ha otorgado una Herencia". La forma general para heredar es:

```
class Nuevo-nombre-Clase : access Clase-que-hereda {
        cuerpo de la nueva Clase
}
```

El acceso ***access*** es opcional, pero de estar presente debe ser: público, privado o protegido. Si las Clases que otorgan la Herencia presentan acceso ***public***, esto significa que todos los elementos públicos del ancestro son también públicos en la Clase que recibe la Herencia. Para nuestro ejemplo, miembros de la Clase **camion** tienen acceso a las funciones miembros de **vehiculo**, tal y como si estas últimas funciones hubieran sido declaradas dentro de **camion**. Sin embargo, las funciones miembros de **camion** no tienen acceso a la parte privada de **vehiculo**.

```
#include <iostream.h>

class vehiculo {
        int ruedas;
        int pasajeros;
public:
        void set_ruedas (int num);
        int  get_ruedas (void);
        void set_pas (int num);
        int  get-gas (void];
};

class camion : public vehiculo {
        int carga;
public:
```

```
        void set_carga (int size);
        int  get_carga (void);
        void mostrar (void);
};

enum tipo {carro, van, carreta};

class automovil : public vehiculo {
        enum tipo tipo_carro;
public:
        void set-tipo (enum tipo t);
        enum tipo get_tipo (void);
        void mostrar (void);
};

void vehiculo:: set_ruedas(int num)

        ruedas=num;
}

int vehiculo:: get_ruedas(void)
{
        return ruedas;
}

void vehiculo:: set_pas(int num)
{
        ruedas=num;
3

int vehiculo:: get_pas(void)
{
        return pasajeros;
}

void camion::set_carga(int num)
{
        carga=num;
}

void camion::get_carga(void)
{
        cout << "ruedas: " << get_ruedas() << "\n"   ;
        cout << "pasajeros: " << get_pas() << "\n";
        cout << "capacidad de carga: " << carga << "\n";
}

void automovil::set_tipo(enum tipo t)
```

```
{
     tipo_carro = t;
}

enum tipo automovil::get_tipo(void)
{
     return tipo_carro;
}

void automovil:: mostrar(void)
f
     cout << "ruedas: " << get_ruedas() << "\n";
     cout << "pasajeros: "<< get_pas() << "\n";
     cout << "tipo: ";
     switch(get_tipo()) {
          case van: cout << "van\n";
          break;
          case carro: cvut << "carro\n";
          break;
          case carreta: cout << "carreta\n";
     }
3

main(void)
{
     camion t1, t2;
     automovil c;

     t1.set_ruedas(18);
     t1.set_pas(2);
     t1.set_carga(3200);
     t2.set_ruedas(6);
     t2.set_pas(3);
     t2.set_carga(1200);
     t1.mostrar();
     t2.mostrar();
     c.set_ruedas(4);
     c.set_pas(6);
     c.set_tipo(van);
     c.mostrar();
     return 0;
}
```

La mayor ventaja de la Herencia es que permite crear una clasificación básica e incorporarla en Clases más especificas.

Tanto camion como automovil incluyen funciones miembros llamadas mostrar(), las cuales presentan en pantalla información sobre cada Objeto. Este es otro aspecto del polimorfismo. Cada mostrar() es enlazada a su propia Clase y el compilador puede fácilmente saber cuál usar en cada circunstancia.

4.2.5 CONSTRUCTORES Y DESTRUCTORES.-

Es muy comun que un Objeto requiera inicialización ántes de ser usado. Por ejemplo: ántes de que una cola de tipo queue pueda ser utilizada, las variables rloc y sloc deben haber obtener valores de 0 mendiante la función init(). Ya que el requerimiento de inicialización es muy comun, C++ permite a los Objetos inicializarse a sí en el momento en que son creados. Esta inicialización automática se lleva a cabo usando una funcion *Constructora.*

Una función Constructora es una función especial. miembro de una Clase y cuyo nombre es el igual al nombre de la Clase. A continuación se presenta la Clase queue,

modificada para utilizar una función Constructora de inicialización:

```
class queue {
      int q[100];
      int sloc, rloc;
public:
      queue(void);
      void qput(int i);
      int qget(void);
};
```

El Constructor **queue()** no tiene un tipo específico de retorno. En C++, **las** funciones Constructoras no pueden retornar valores.

```
queue::queue(void);
{
      sloc=rloc=0;
      cout << "cola inicializada\n";
}
```

En este caso el mensaje "cola inicializada" es presentado para ilustrar el uso del Constructor. En la práctica, **la** mayoría de las funciones **Constructoras** no presentan ninguna Entrada ni Salida.

La función Constructora de un Objeto es llamada cuando el Objeto es creado, es decir cuando la declaración del Objeto es ejecutada. En el caso de Objetos Locales, el

Constructor es llamado cada vez que la declaración del Objeto es encontrada.

El complemento del Constructor es el ***Destructor***. En muchas circunstancias, un Objeto necesita llevar a cabo ciertas acciones en el momento de ser destruído. Recordemos que los Objetos Locales son creados cuando se entra a su bloque y destruídos cuando se abandona el bloque. A su vez, los Objetos Globales son destruídos cuando el programa termina. Existen muchas razones por las que se necesita una función Destructora. Ej: un Objeto puede necesitar desalocar memoria que previamente alocó.

El Destructor tienen el mismo nombre que el Construc-**tor** pero está precedido de un " ~ ". A pesar de que la Clase **queue** no necesita un Destructor, a continuación se crea uno como ilustración.

```
#include <iostream.h>

class queue {
     int q[100];
     int sloc, rloc;
public:
     queue(void);
     ~queue(void);
     void init(void);
     void qput(int i);
     int qget(void);
};
```

```
queue::queue(void)
{
  rloc=sloc=0;
  cout << "cola inicializada\n";
}

queue::~queue(void)
{
  cout << "cola destruida\n";
}

void queue::qput(int i)
{
      if (sloc==100) {
            cout<<"La cola eota llena";
            return;
      }
      sloc++;
      q[sloc]=i;
}

void queue::qget(void)
{
      if (rloc==sloc) {
            cout<<"La cola esta vacia";
            return 0;
      }
      rloc++;
      return q[rloc];
}

main(void)
{
      queue a,b;

      a.qput(10);
      b.qput(19);
      a.qput(20);
      b.qput(1);
      cout << a.qget() << " ";
      cnut << a.qget() << " ";
      cout << b.qget() << " ";
      cout << b.qget() << "\n";
      return 0;
}
```

La ejecución del programa presenta lo siguiente en pantalla:

```
cola inicializada
cola inicializada
10 20 19 1
cola destruida
cola destruida
```

4.2.6 FUNCIONES friend,-

Si una función no miembro de una Clase es declarada "amiga" de esa Clase, puede obtener acceso a las partes privadas de ella. A continuación, la función **frd()** es declarada "amiga" de la Clase **cl**. La palabra reservada **friend** precede a **la** declaración de la función.

```
class cl {
     .
     public:
          friend void frd(void);
                    .
                    .
};
```

Una razón por **la** cual las funciones **friend** son permitidas en C++, es para mantener la eficiencia en situaciones en las cuales dos Clases deben compartir una misma función.

Consideremos un programa que define dos Clases llamadas **linea y cuadro.** La Clase **linea** contiene todos los

datos y códigos necesarios para dibujar una linea horizontal punteada de cualquier longitud, empezando en una coordenada (x,y) y utilizando un determinado color. A su vez, la Clase **cuadro** contiene todo el código y datos necesarios para dibujar una cuadro de acuerdo a las coordenadas de esquina superior izquierda y esquina inferior derecha y con un color determinado. Las dos Clases usan la función **mismo_color()** para determinar si tanto la línea como el cuadro han sido dibujados con el mismo color.

```
class linea;

class cuadro {
     int color:
     int supx, supy;
     int infx, infy;
public:
     friend int mismo_color (linea I. cuadro b);
     void mismo_color (int c);
     void def_cuadro (int x1, int y1, int x2, int y2);
     void mostrar_cuadro (void);
3;

class linea {
     int color;
     int inix, iniy;
     int largo;
public:
     friend int mismo_color (linea l, cuadro b);
     void mismo_color (int c).:
     void def_linea (int X, int y, int l);
     void mostrar_linea ();
};
```

La función **mismo_color()**, la cual no es miembro de ninguna de las dos Clases, pero sí amiga de las dos, retor-

na 1 si tanto el Objeto línea como el Objeto cuadro están dibujados en el mismo color y 0 en el caso contrario.

```
int mismo_color (linea l, cuadro b)
{
     if (l.color==b.color) return 1;
     return 0;
}
```

La función **mismo_color()** necesita acceso a las partes privadas, tanto de **linea** como de **cuadro**, para poder ejecutar su tarea eficientemente. Se podrían haber creado funciones públicas que retornen los colores de la línea y del cuadro, y así cualquier función era capaz de comparar los dos colores. Sin **embargo**, esta solución es ineficiente, pues requiere más llamados a funciones.

En el listado a continuación, es de notar la declaración vacía de **linea** al inicio de la declaración de las Clases. Ya que la función **mismo_color()**, "amiga" de **cuadro**, referencia a **linea** antes de que ésta sea declarada, **era** necesario referenciar **a linea** en un principio. Si esto no se hace, el compilador no sabrá qué es linea, al encontrarla en la declaración de **cuadro**. En C++, para referenciar previamente una Clase símplemente se pone la palabra reservada **class**, seguida del nombre de la Clase. Usual-

mente, el único caso que necesita referencias previas. es cuando se utilizan funciones "amigas".

```
#include <iostream.h>
#include <conio.h>

class linea;

class cuadro {
     int color;
     int supx. supy;
     int infx, infy:
public:
     friend int mismo_color(linea 1. cuadro b);
     void set_color(int c);
     void def_cuadro(int xl, int yl, int x2, int y2);
     void mostrar_cuadro(void) :
};

class linea {
     .int color:
     int inix, iniy;
     .int largo;
public:
     friend int mismo_color(linea 1, cuadro b);
     void mismo_color(int c);
     void def_linea(int x, int y, int l);
     void mostrar_linea();
};

int mismo_color (Jinea l, cuadro b)
{
     if (l.color==b.color) return 1:
     return 0;
}

void cuadro::set_color(int c)
{
     color=c;
3

void linea::set_color(int c)
{
     color=c;
}
```

```
void cuadro::define_cuadro(int x1, int y1,
                           int x2, ink y2)
{
      supx=x1;
      supy=y1;
      infx=x2;
      infy=y2;
}

void cuadro::mostrar_cuadro(void)
{
      int i;

      textcolor(color);
      gotoxy(supx, supy);
      for (i=supx; i<=infx, i++) cprintf("-");
      gotoxy(supx, infy-1);
      for (i=supx; i<=infx; i++) cprintf("-");
      gotoxy(supx, upy);
      for (i=supy; i<=infy; i++) {
            cprintf("¦");
            gotoxy(supx,i);
      }
      gotoxy(infx, supy);
      €or (i=supy; i<=infy; i++) {
            cprintf("¦");
            gotoxy(infx,i);
      }


void linea::def_linea(int x, int y, int I)
{
      inix=x;
      iniy=y;
      largo=l;
}

void linea::mostrar_linea(void)
{
      int i;

      textcolor(color);
      gotoxy(inix, iniy);
      for (i=0; i<largo; i++) cprintf("-");
}

main (void)
{
```

```
        cuadro b;
        linea 1;

        b.def_cuadro(10,10,15,15);
        b.set_color(3);
        b.mostrar_cuadro();
        1.def_linea(2,2,10);
        1.set_color(2);
        1.mostrar_linea();

        if (!mismo_color(1,b)) cout<< "Los   colores   son
                                       diferentes";
        cout<< "\nPresione una Tecla";
        getch();

        1.def_linea(2,2,10);
        1.set_color(3);
        1.mostrar_linea();

        if (mismo_color(1,b)) cout<<  "Los   colores   son
                                       iguales";
        return 0;
}
```

4.3 IMPLEMENTACION DE INTERFASES GRAFICAS CON C++.-

En años recientes la expresion *Interfases Gráficas al Usuario* (IGU) se ha popularizado. Las IGU presentan al usuario una especie de ambiente artificial, una abstracción del ambiente físico. Ejs: Las Interfases de Microsoft Windows y Apple Macintosh. En este tipo de sistemas el computador presenta la información básica, manejando objetos semejantes a los de un escritorio verdadero: papeles, archivos, cajones y basureros. Usando un Mouse, el usuario selecciona un dibujo y de esa forma se corre un programa o se inicializa un diálogo con el sistema. Hoy en día, las

IGU soportan gráficos a color de alta resolución, capacidad de sombreado, menús variados y otras facilidades, que proveen una presentación más realista del mundo físico.

Como parte del presente trabajo de Investigación, se ha implementado un Prototipo que presenta, mediante Interfases Gráficas en C++, los Menús del "Sistema Personal de Manejo de Documentos" descrito en el Capítulo III. Se utilizó como Compilador: Borland Turbo C++ Versión 2.0. Para el desarrollo del Prototipo se requiere un computador PC o AT compatible con IBM, con DOS Versión 3.1 o mayor, equipado con un Mouse compatible con Microsoft y una tarjeta EGA o VGA.

A continuación se ofrece un análisis general de las Clases definidas por Ted Faison [FAIS91], y que constituyen la base para la implementación de nuestro programa demostrativo de las Interfases Gráficas al Usuario, en C++.

4.3.1 CONCEPTOS BASICOS SOBRE IGUS.-

El primer paso en la construcción de una IGU, es enseñar al usuario a ver la pantalla como un "Panel de Control". Actuar en este panel es equivalente a llamar a un comando del programa controlado por la IGU.

En la construcción de interfases intuitivas, es importante mantener consistencia. Al estar aislados de los detalles de implementación, los usuarios desarrollan un modelo mental de cómo trabaja el Sistema. Cuanto más rápido se formen este modelo mental, les será más fácil utilizar el Sistema.

Una Superficie de Control presenta Objetos como botones, potenciómetros móviles, lecturas digitales, etc. Todos estos dispositivos de control son llamados ***Iconos***. Un Icono puede ser visible o no.

En la mayoría de aplicaciones, una única Superficie de Control no puede presentar en forma adecuada toda la informacion disponible. Así pues, se organiza la Superficie de Control como una secuencia de páginas de un libro y se relegan las funciones a diferentes páginas, de acuerdo a como ellas serán usadas. De esta manera, una Superficie de Control debe tener un dispositivo Selector de Páginas, presentado en la Pantalla con el resto de controles.

En general podemos decir que en la construcción de una interfase intervienen diferentes objetos: Gráficos, Controles visibles o no, Manejadores de Mouse, Programas de

Aplicación, Lazos de Eventos, etc. En el nivel más bajo de la abstracción se encuentran los cuadros, botones, etc. presentados en la pantalla. Luego se encuentra la organización de estos Iconoe en Páginas. A continuación, es necesario un paquete que maneje las páginas. En el nivel más alto se inician las acciones de acuerdo a los requerimientos del usuario.

4.3.2 COMPONENTES DE UNA IGU.-

Los componentes basicos del programa implementado se basan en las Clases desarrolladas por Ted Faison para la implementación de Interfases Gráficas con C++ [FAIS91].

A. MANEJADOR DE EVENTOS.-

Se llama *EVENTO* a cualquier acción efectuada por el usuario. El **Manejador de Eventos** cumple el propósito de toda IGU, de soportar la entrada de datos por parte del usuario, ya sea desde el teclado o utilizando un Mouse.

Para tratar con Eventos en una forma consistente, todos ellos son detectados en un ***Lazo de Eventos***. La IGU espera por un Evento y llama a un programa de Aplicación cuando uno es detectado. Ningun Objeto o aplicación puede correr lazos locales para esperar por eventos. Son los

eventos los que invocan Objetos. Cuando el procesamiento de un evento es terminado. el Manejador de Eventos retoma el control.

La IGU tiene una función llamada *top-level* la cual toma el control después de que el programa es iniciado. Existe un lazo que chequea por entradas por parte del usuario: acciones del Mouse o golpes de teclas. Al detectar un click del mouse, la función *top-level* notifica al Objeto pertinente del Sistema.

Los eventos son procesados de acuerdo a su tipo y la mayoría son dependientes de la máquina. Algunos de los eventos que pueden ser detectados son:

- Entrada normal desde el teclado
- Entrada <Ctrl><Break>
- Error de. división por cero
- Cambios de status desde el teclado: Shift.. Alt. Caps Lock
- Errores de impresión: no en línea, falta de papel
- Mouse: clicks o movimientos con botón presionado
- Interrupciones de Software
- Interrupciones del Reloj

B. MANEJADOR DE GRÁFICOS.-

El **Manejador de Gráficos** implementa la **Clase Graphics**. la cual contiene los Objetos Gráficos básicos de una IGU. Existen funciones para presentar en pantalla: círculos, cuadrados, lineas, texto y sombras. Dispone de 16 colores. Las funciones disponibles son muy flexibles y sirven para crear, en base a ellas, gráficos complejos. Soporta adaptadores EGA y VGA, y utiliza una resolución de 640 x 350.

La Clase **Graphics** no soporta: operaciones **con** paletas de colores, la utilización de diferentes tipos de letras y el llenar areas con un patrón determinado. Para añadir estas capacidades, existen diversas alternativas:

1. Se ias añade a la **Clase Graphics**.
2. Se deriva **una** Clase nueva y se la añade a la **Clase Graphics**.
3. Se escribe por completo una Clase nueva.

Cuando se **llama** a una función miembro de **la Clase Graphics**. se comunican A ella todos los atributos necesarios para realizar una determinada acción. Podemos reconocer que la librería de funciones BGI trabaja de manera diferente, pues su interes es la rapidez: se establecen atributos mediante ciertas funciones y luego otra funciones

- Interrupciones del Puerto Serial
- Interrupciones del Rus IEEE 488

El Manejador de Eventos implementa la **Clase Events**, la **Clase Mouse** y la **Clase Delay.**

Clase Events: Representa por si sóla al Manejador de Eventos. Tiene por clases hijas a **Mouse** y **Delay.** Aisla de la aplicación las dependencias de **hardware.** Maneja el Mouse, el chip temporizador y el teclado.

Clase Mouse: Provee una interface con un Driver de Mouse, compatible con Microsoft. Aisla completamente el Driver de la Aplicación y utiliza un sistema de coordenadas que coincide con el de la Pantalla (el origen del sistema de coordenadas está en la esquina inferior izquierda).

Clase Delay: Implementa un objeto de autocalibración que se ajusta a la velocidad de la máquina en la que funciona. Es especialmente util para temporizar rangos de milisegundo. cuando la función estándard sleep no puede ser usada.

los utilizan implícitamente. El problema de esta implementación es que siempre hay que recordar qué atributos se encuentran definidos en un momento determinado. En la **Clase Graphics** cada vez que se llama a una función, se especifican los atributos de color y estilo de línea. Esto hace al sistema mas lento, pero favorece la simplicidad.

La Pantalla sigue el sistema derecho: el origen del sistema de coordenadas está en la esquina inferior izquierda. Todas las coordenadas de la **Clase Graphics**, al igual que en el caso de la **Clase Mouse**, son convertidas a este formato.

C. MANEJADOR DE PANTALLA.-

El **Manejador de Pantalla** comprende tres Clases. En primer lugar, implementa la **Clase Icon**, básica para crear un objeto presentable **en** pantalla. A su vez, la **Clase DisplayList**, maneja en una lista, todos los objetos presentados en la pantalla en un momento específico. Por último, la **Clase Screen** agrupa toda la actividad de la pantalla y tiene como Clases Hijas a **DisplayList, Icon** y **PageSelector**.

Se dice que la **Clase Icon** es abstracta, pues su único propósito es actuar como la Clase raíz de todos los objetos

presentables en pantalla. Los Objetos **Icon**, por sí sólos, no tienen consistencia física.

La Pantalla tiene una configuración típica para presentar las Páginas de Iconos [FAIS91]. Al lado derecho está el "Selector de Páginas", que permite swichear de una Página de Aplicación a otra. En la zona superior se encuentra un área para el "Título de la Página". En el espacio restante de la Pantalla, llamado "Area de Trabajo", la **Clase Screen** maneja los Iconos, despachándoles eventos relacionados al mouse y al teclado (**Clase Events**), y controla la escritura y el borrado de los objetos en la pantalla, utilizando la Lista de Presentaciones (**Clase Display List**).

D. ICONOS BASICOS.-

Clase Box: Define cuadros rectangulares, presentables en pantalla. Se puede seleccionar la forma, el borde y el color interno del cuadro. Por simplicidad, sólo se utilizan colores sólidos para llenar las áreas. Los Cuadros son los objetos que soportan la mayoría de items presentables.

Clase Legend: Toma un string y lo presenta dentro de un area rectangular de la pantalla. Existen instrucciones de formateo, que permiten alinear a la derecha, a la izquierda o centrar.

Existen otros Iconos básicos, como cursores y botones, que si bien no fueron utilizados en nuestro pequeño programa, podrían servir para continuar su implementación.

E. ICONOS COMPUESTOS.-

Clase Selector: Utiliza los Cuadros y Leyendas, previamente definidos, e implementa Menús Verticales y Horizontales, donde el usuario puede seleccionar una opción deseada.

A su vez, mediante Botones y Cursores se pueden construir Potenciómetros y Bancos de Botones, pero estas implementaciones no fueron utilizadas por nuestro programa.

F. PAGINAS.-

Una Página básica está vacía y presenta su Titulo en la zona superior. Páginas reales tienen Controles e Iconos relacionados con el programa de Aplicación. La **Clase Page** no soporta por sí sóla los varios objetos a localizarse en

una Página de Aplicación, pues no sabe que existen. Cada Página específica es una Clase derivada, responsable de añadir la serie de capacidades que la van a representar.

A su vez, la **Clase PageSelector** permite la selección de una Página de Aplicación, la cual es presentada al usuario en el Area de Trabajo de la Pantalla.

CONCLUSIONES Y RECOMENDACIONES

El presente trabajo de Investigación ofrece como producto práctico y concreto, el "Sistema Personal de Manejo de Documentos" (SPMD). Este combina técnicas de indexación manual y automatizada para lograr reflejar la información contenida en los Documentos. Es así como por un lado, pide al usuario que determine una serie de Palabras Claves, y por el otro, extrae del Título del Documento las palabras representativas de él. El conjunto de palabras asociadas a un Documento, son utilizadas por los algoritmos de búsqueda en la recuperación de la información deseada por el usuario.

Nuestro Trabajo presenta también una serie de Productos existentes en el mercado, para Manejo de Documentos, y pone énfasis en la ventaja de que implementen la indexación total del Texto ("Full-Text Indexing"). Sin embargo, es claro que lograr entender el lenguaje natural de los libros, de manera que la recuperación de información pueda basarse en un análisis total del texto, constituye una implementación demasiado extensiva.

Frente a estas dos alternativas: Palabras Claves vs Indexación Total del Texto, aparece hoy en dia una tercera alternativa: la Indexación de Conceptos, la cual constituye la nueva meta de la Recuperación de Información.

Esta nueva meta contempla el desarro de Sistemas Basados en Conocimiento, que permitan a los usuarios plantear su requerimiento de información en forma directa. Así, eliminando la necesidad de conocer la estructura de la Base de Datos y las estrategias de búsqueda, y poniendo más énfasis en el conocimiento del Tópico, el especialista y el estudiante pueden utilizar más fácilmente los Sistemas de Recuperación de Información.

La implementación de este nuevo paradigma de la Recuperación de Información es una tarea difícil ya que, además de la necesidad de indexar Conceptos, las interfases con el usuario tendrán que desarrollarse para alcanzar la meta de un acceso directo y efectivo por parte del usuario.

En lo referente a la implementación de nuestro "Sistema Personal de Manejo de Documentos", es posible incorporar a él una serie de facilidades que aumentarían su funcionalidad. Podemos hacer las siguientes recomendaciones:

Generar un índice en base al campo de Resumen, ampliando así la información sobre el contenido de un Documento.

El punto anterior podría ser mejorado aún más, reemplazando el campo de 'Resumen' por un campo de 'Contenido'. De esta manera se podría implementar la indexación "Full Text", gracias a la conversión a formato electrónico de todo el contenido del Texto.

Permitir el ingreso automático del texto de una Publicación, a través del empleo de Scanners, evitando así el tedioso trabajo de ingreso manual.

Utilizar Herramientas de Hypertext para representar, en una forma más gráfica para el usuario, las relaciones existentes entre los Documentos.

Elaborar una versión del "Sistema Personal de Manejo de Documentos" que funcione bajo ambiente de Redes. El principal cambio sobre la versión actual sería supervisar el acceso a los Archivos de la Base de Datos, por parte de los múltiples usuarios.

Incorporar al Sistema la Búsqueda por "Proximidad o Similitud de Palabras": palabras con igual raíz, pero diferente sufijo o terminación, estarían relacionadas entre sí. Ej: Network, Networks, Networking. Esta característica aumentaría la inteligencia del Sistema.

Desarrollar el Sistema mediante Interfases Gráficas y permitir el uso del Mouse. Para esta meta existe ya un primer paso dado, puesto que los Menús del "Sistema Personal de Manejo de Documentos" fueron ya implementados mediante procedimientos en lenguaje C++.

REFERENCIAS BIBLIOGRAFICAS

[DUNC91] Duncan, Thom. DOCUMENT MANAGEMENT SOFTWARE. Revista Lan Times, EEUU, 1991. pp. 61-80.

[FAIS91] Faison, Ted. GRAPHICAL USER INTERFACES WITH TURBO C++. SAMS, EEUU, 1991. 286 pp.

[HORA88] Horak, Karl. TURBO PROLOG & DBASE III PLUS. Revista Al Expert, EEUU, 1988. pp. 26-33.

[PCKW89] Parsaye Kamran, Chignell Mark, Khoshafian Setrag y Wong Harry. INTELLIGENT DATABASES: OBJECT ORIENTED, DEDUCTIVE HYPERMEDIA TECHNOLOGIES. Wiley, EEUU, 1989. 478 pp.

[SCHI90] Schildt, Herbert. TURBO C/C++: THE COMPLETE REFERENCE. McGraw Hill, EEUU, 1990. 1016 pp.

[STEV87] Stevens, Al. C DATA BASE DEVELOPMENT. MIS: PRESS, EEUU, 1987. 312 pp.

[STEV90] Stevens, Al. TEACH YOURSELF... C++. MIS: PRESS, EEUU, 1990. 274 pp.