UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Ingeniería

SISTEMAS RECUPERADORES DE INFORMACION BIBLIOGRAFICA

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACION

P R E S E N T A N

MARITZA ZUGASTI BOBADILLA

EDGAR IVAN SALAZAR SANDOVAL

Director: Sergio Castro Resines

1 9 8 7

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

INDICE DE CONTENIDO

INDICE DE CONTENIDO

PREFACIO

Un sistema recuperador de informacion es un sistema que se utiliza para almacenar documentacion que debera procesarse, buscarse, recuperarse y diseminarse para diferentes poblaciones de usuarios. Los sistemas recuperadores de informacion comparten algunos conceptos con otros sistemas de informacion como lo son los administradores de bases de datos y los sistemas para toma de decisiones. En este tipo de sistemas es fundamental la organizacion de los datos tanto a nivel conceptual como fisico para lograr la maxima eficiencia, ya que de esto se derivan las tareas de busqueda y diseminacion que son probablemente su principal objetivo.

Es importante mencionar que con frecuencia este tipo de sistemas se utiliza para el manejo de informacion bibliografica o textual; en cambio, otros sistemas de informacion se encargan principalmente de los datos numericos o factuales; asi como de su procesamiento para la toma de decisiones.

En la actualidad es comun encontrar que este tipo de sistemas se encuentre automatizado y es por ello que existen grupos de trabajo de computo desarrollando y perfeccionando los mecanismos y bases para su optimizacion en procesos automaticos. El interes generado por este tipo de desarrollos en el campo computacional han llevado al estudio de la teoria de la informacion, probabilidad y linguistica.

La teoria y la tecnica empleadas en estas disciplinas pueden reunirse para la creacion de modelos y profundizar con amplio conocimiento sobre algunos aspectos; de esta manera, permite auxiliar a los usuarios desde varios puntos de vista durante la busqueda y recuperacion.

El objetivo de este texto es presentar los metodos y criterios para poder evaluar y elegir entre diversas opciones; asi como proporcionar los principales conceptos practicos para el diseno de sistemas recuperadores de informacion. Ademas proporcionara las bases para que el usuario pueda entender y practicar con mayor eficiencia la elaboracion de sus estrategias de busqueda y recuperacion de informacion.

El texto comienza con la comparacion de los procesos que permiten realizar los sistemas de informacion con el proposito de lograr una perfecta distincion entre ellos. Tambien se presentan direfentes organizaciones de archivos de computadora para hacer evidente cual es la mas adecuada para los sistemas recuperadores de informacion no estructurada rigidamente, para ello en el segundo capitulo se mencionan los operadores que se utilizan con mas frecuencia durante el proceso de busqueda. Este capitulo finaliza con los metodos convencionales para plantear estrategias de busqueda, concluyendo con la descripcion y ejemplos tipicos obtenidos de la practica de algunos sistemas comerciales.

En el capitulo tres se presentan los puntos mas importantes a considerar para el diseno de las bases de datos; esto es, desde un estudio de costo-beneficio para la conveniencia de automatizar el sistema, hasta la posibilidad de codificar la informacion para reducir el espacio de almacenamiento.

En el capitulo cuatro se analizan algunas tecnicas estadisticas para la creacion de modelos y obtencion de los parametros del sistema que se esta simulando. Este capitulo esta totalmento enfocado al diseno de los programas involucrados en este tipo de sistemas.

La automatizacion de los procesos dedicados a la extraccion de terminos son uno de los puntos mas relevantes en el analisis automatico de la informacion para el reconocimiento del contenido y ubicacion dentro del contexto para la seleccion de descriptores de la informacion. Esto es lo que se trata en el capitulo cinco con amplio detalle linguistico y aspectos prioritarios en la construccion de tesauros.

El capitulo seis muestra los metodos y parametros mas adecuados para evaluar la eficacia y efectividad de los sistemas y en base a ellos modificar el sistema de manera que los parametros varien y se tenga un sistema con un alto grado de eficiencia y efectividad.

La administracion y el intercambio de informacion hacen que el contenido de las bases de datos de un sistema recuperador de informacion sea mas valiosa dada su novedad, actualidad o completez. Estos puntos son tratados en los capitulos siete y ocho desde el punto de vista de servicio, anexando algunas definiciones y recomendaciones internacionales.

En los capitulos nueve y diez se contemplan las tendencias tecnologicas enfocadas a la mejoria de los equipos especializados para los propositos de busqueda, procesamiento, almacenamiento, etcetera; asi como las tecnicas linguisticas implantadas en las computadoras para el reconocimiento e interpretacion del lenguaje y auxiliares para la automatizacion de los procesos que en la actualidad se llevan en forma manual.

Finalmente el capitulo once, es una aplicacion de los conceptos mencionados a lo largo del texto para concluir con la implantacion de un caso practico; que a su vez, es el objetivo fundamental de este trabajo.

Se ha incorporado un listado con la bibliografia mas relevante que ha servido como base para la elaboracion de algunos capitulos, asi como para enriquecer los conocimientos que teniamos y que fuimos madurando durante la elaboracion del trabajo.

Los anexos que se incorporan son parte del material que desarrollamos para el mejor entendimiento de algunos topicos cruciales en los sistemas de informacion. Estos incluyen el estudio de la semantica y la sintaxis, practica de estructuras de datos para una recuperacion rapida de documentos y estudio de estructuras de datos para el manejo de informacion no estructurada (bibliografica o textual).

CAPITULO 1

Introducción a los Sistemas Recuperadores de Información

1.1 Definicion de recuperador de informacion

Un sistema recuperador de informacion es un sistema util para almacenar elementos de informacion que necesitan procesarse, buscarse, recuperarse y diseminarse a diversas poblaciones de usuarios. Los sistemas recuperadores de informacion comparten muchos conceptos con otros sistemas de informacion, principalmente con los administradores de bases de datos y con los sistemas de toma de decisiones. Es importante y necesario organizar eficientemente tanto los registros de almacenamiento, como las estructuras de la informacion, para que sea posible implementar procedimientos de busqueda especificos, y metodos efectivos para la diseminacion de datos recuperados y la interaccion con los usuarios del sistema.

Los sistemas recuperadores de informacion se utilizan principalmente en el manejo de informacion bibliografica y textual. En contraste con otros sistemas, administradores de bases de datos y sistemas de administracion de informacion que procesan datos estructurados, y los sistemas de pregunta-respuesta (question-answering) que utilizan estructuras de informacion muy complicadas y procedimientos para hacer inferencias, disenados para responder preguntas en areas determinadas.

La recuperacion de informacion se encarga de la representacion, almacenamiento, organizacion y acceso de elementos de informacion. En principio no hay restricciones en el tipo de informacion que se va a manejar con el recuperador. Actualmente, la mayoria de los datos almacenados en los sistemas recuperadores de informacion son resumenes o descripciones. Este tipo de informacion debe analizarse ampliamente antes de alimentarse al recuperador, con el fin de satisfacer las necesidades de los usuarios del sistema. Los elementos que podemos encontrar en un recuperador de informacion incluyen cartas, documentos de todo tipo, periodicos, articulos, libros, resumenes medicos, reportes de investigaciones, etcetera.

Para facilitar la tarea de obtencion de informacion, las bibliotecas o centros de informacion proveen ayudas auxiliares. Cada elemento que entra en su acervo se analiza y se eligen terminos que hagan la descripcion mas adecuada para que refleje el contenido de informacion, dichos terminos reciben el nombre de descriptores. Cada elemento se clasifica de acuerdo a normas ya establecidas y se incorporan a la coleccion. Existen procedimientos para formular solicitudes disenadas con el objeto de satisfacer las necesidades de informacion y comparar estas peticiones, o queries. Finalmente se utiliza un mecanismo de recuperacion y diseminacion de informacion para escoger los elementos que contengan interes potencial para el usuario del sistema. Todos estos pasos se llevan a cabo en las bibliotecas

convencionales donde para la busqueda de informacion se utiliza como herramienta principal el sistema de tarjetas.

Se observa claramente que un sistema de este tipo requiere por su naturaleza, estar actualizado y contener volumenes gigantescos de informacion potencialmente importante y eliminar solo aquella en la que no haya duda de su obsolescencia, para que realmente satisfaga sus principios de operacion.

En la actualidad hay que establecer un compromiso entre la actualizacion y la completitud de la informacion debido a los recursos limitados con los que se cuentan. Por lo tanto, hay que incorporar al sistema de informacion todos aquellos elementos "importantes" que justifiquen su existencia. Siendo una tarea dificil evaluar la importancia de la informacion por adelantado, recurriendose en la mayoria de los casos a la experiencia o a los criterios del especialista de la informacion especifica.

La localizacion o incorporacion fisica de un documento esta relativamente resuelto con las normas de catalogacion existentes, las cuales se basan en la utilizacion de numeros normalizados y generados con procedimientos ya definidos. Estos numeros permiten hacer agrupamientos por temas, o areas especificas de conocimiento. Usualmente la clasificacion generica de un documento resulta ineficiente, ya que produce duplicidad, lo cual implica la necesidad de detallar la descripcion. Este es un trabajo lento y requiere de habilidad y conocimiento del area. La duplicidad se evita tomando ciertas decisiones locales o personales y agregando claves al numero de clasificacion, ocasionando el problema de tener que recurrir al encargado para localizar con exactitud cierto documento.

Existen otro tipo de ayudas para la clasificacion de la informacion y son los llamados Encabezamientos de Materia, estos sirven para permitir cambios en los indicadores de clasificacion que sugiere el metodo comunmente utilizado para el documento; es decir, es una clasificacion mas fina sobre el area particular. La clasificacion logica de la informacion puede hacerse por autor, titulo, tema, area, etcetera o la combinacion de ellas, sin afectar su ubicacion fisica.

1.2 CLASIFICACION DE LOS SISTEMAS DE INFORMACION

Los sistemas recuperadores de informacion (IRS, Information Retrieval System) presentan similitudes con otros sistemas como los administradores de informacion (MIS, Management Information System), administradores de bases de datos (DBMS, Data Base Management System), toma de decisiones (DSS, Decision Support System) y los de pregunta-respuesta (QAS, Question-Answering System). La figura 1.1 muestra la relacion que guardan entre si los sistemas de informacion.

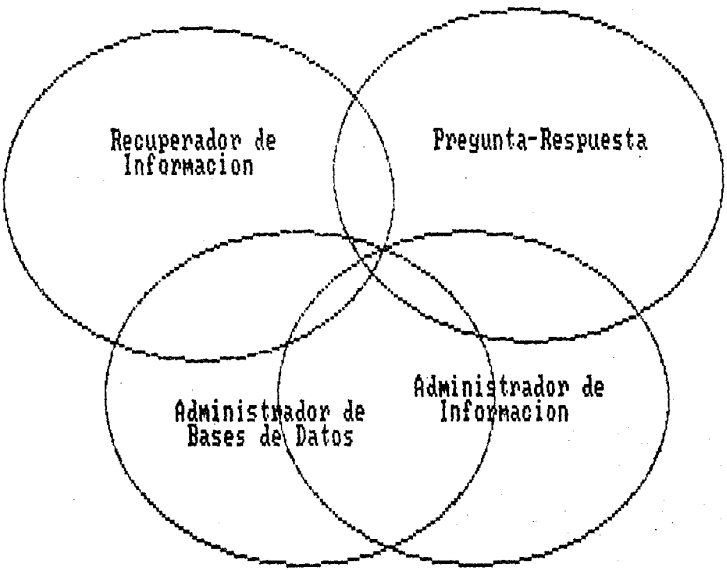


Figura 1.1 Interelacion de los sistemas de Informacion

1.2.1 Sistemas recuperadores de informacion (IRS).

La entrada al sistema es la informacion con el texto escrito en el lenguaje natural del documento o la descripcion y resumen. La salida del sistema es un conjunto de referencias como resultado de una peticion de busqueda. Estas referencias proveen al usuario del sistema, con la informacion sobre los documentos de interes potencial.

1.2.2 Sistemas administradores de bases de datos (DBMS).

Cualquier sistema de informacion se basa en una coleccion de elementos almacenados (en una base de datos) que necesitan ser recuperados. Por lo tanto, un administrador de bases de datos es simplemente un sistema disenado para manipular y mantener el control de la informacion almacenada en cualquier base de datos. Actualmente, un sistema administrador de bases de datos, se encarga del almacenamiento, mantenimiento y recuperacion de hechos residentes en el sistema en forma explicita. Es decir, que la informacion no se guarda en el lenguaje natural del texto, sino que esta almacenada en tablas. En una base de datos, la informacion de cada registro, se encuentra separada en varios campos, y cada campo contiene el valor de una caracteristica especifica o atributo que identifica al registro correspondiente. Este valor caracteristico de cada registro, permite la identificacion unica de la informacion solicitada, por lo tanto se utiliza como un identificador del registro.
La informacion contenida en un sistema administrador de bases de datos, incluye un buen manejo de datos numericos, estadisticos o la capacidad de utilizar los recursos numericos de computo.

1.2.3 Sistemas de informacion administrada (MIS)

Es un sub-sistema administrador de bases de datos, ya que esta dirigido a las necesidades de los administradores. Las funciones que realiza un administrador en una compania, dependenden de la disponibilidad de muchos tipos de datos. La informacion se presenta normalmente como un rango de valores en atributos particulares. Por lo tanto, un sistema administrador de informacion esta enmarcado por las mismas funciones que un sistema administrador de bases de datos. Sin embargo, las contribuciones de un sistema de este tipo estan sujetas a procesos especiales que normalmente no existen en un DBMS.

1.2.4 Sistemas de toma de decisiones (DSS)

Normalmente, los sistemas recuperadores de informacion no realizan funciones de administracion de informacion y vice-versa. Sin embargo, es posible ensamblar estos dos tipos de sistemas para que compartan la misma estructura, que por un lado sean recuperadores de informacion, y por otro administradores de bases de datos, que ademas permitan hacer graficas por computadora y otras posibilidades tecnicas que colectivamente provean de herramientas utiles para el proceso de toma de decisiones.

Los sistemas de toma de decisiones son poco conocidos y actualmente siguen siendo muy limitados en areas especificas.

1.2.5 Sistemas de pregunta-respuesta (QAS).

Proveen acceso a la informacion factual en un lenguaje natural. La base de datos almacenada tambien consiste de un gran numero de hechos relacionados con areas especiales de discursos, cubriendo el contexto de las conversaciones que tuvieron lugar. Las preguntas del usuario pueden recibirse en lenguaje natural y la respuesta del sistema sera tambien en forma natural. La tarea del sistema pregunta-respuesta consiste en analizar las peticiones del usuario, comparar la peticion analizada con el conocimiento almacenado y ensamblar una respuesta con los hechos aparentemente relevantes.

Actualmente los sistemas pregunta-respuesta solamente existen como dispositivos experimentales. La extraccion del significado a partir del lenguaje natural y la determinacion de las reglas generales del comportamiento inteligente se ven como las mayores barreras para crear un sistema efectivo de proposito general.

1.3 CATEGORIAS DE RECUPERACION DE INFORMACION

Los usuarios pueden requerir de un sistema de base de datos tres tipos diferentes de información que son: recuperación de hechos, inferencia estadística e inferencia deductiva.

1.3.1 Recuperación de hechos

La recuperación de hechos es el empleo primario de los bancos bibliográficos de datos o de los que contienen material textual. Este tipo de recuperación es el más popular.

Un sistema recuperador de hechos se caracteriza por el empleo extensivo de la operación de localización y por el uso de acceso indexado o directo a la información. En una consulta pueden participar una o varias relaciones. Para agilizar el tiempo de respuesta de las consultas se puede asignar la estructura de la base de datos de manera que coincida con las consultas, esto se puede hacer debido al hecho de que las respuestas están asignadas de antemano a un tipo específico de consulta.

1.3.2 Inferencia estadistica

Cuando el tamano de la respuesta a una consulta es tan grande que el usuario no pueda abarcarla, resulta necesario emplear técnicas que reduzcan los datos. Las técnicas de reducción pueden variar desde simples tabulaciones cruzadas hasta extensos procesamientos estadísticos, con el objeto de proporcionar datos significativos. Por lo tanto, el proceso de obtención de resultados es más complejo que el de consulta-respuesta.

Debido a que con este tipo de consulta es necesario hacer varios ensayos hasta que se obtenga un planteamiento satisfactorio para una consulta, resulta conveniente producir resultados intermedios que le ayuden al usuario a plantear los vínculos de relaciones y a analizar la información.

Los resultados gráficos de tendencias, los histogramas de distintos subconjuntos de variables dato de interés y la presentación de los datos en diagramas de dispersión proporcionan información acerca de las posibles correlaciones entre variables, a su vez las estadísticas permiten identificar eventos o tendencias desusados. Este tipo de procesamiento puede producir informes de excepciones que permitan determinar y corregir en caso necesario, aquellos patrones inesperados. De manera que, los datos pueden volverse información.

La inferencia estadística implica la recuperación de grandes cantidades de datos bien estructurados.

Con la tecnología común resulta difícil presentar situaciones de más de tres dimensiones; con lo que aumentan los requerimientos de procesamiento, ya que resulta necesario presentar las constelaciones de datos en varias combinaciones para desarrollar una comprensión de los vínculos entre los datos.

Para evitar el reprocesamiento de grandes volúmenes de datos es conveniente almacenar resultados intermedios en áreas de trabajo. En muchas aplicaciones es conveniente que la actualización de la base de datos no se haga durante el análisis ya que esto provocaría inconsistencias, siendo más adecuado copiar un subconjunto de la base de datos al área de trabajo temporal y una vez que el análisis se haya terminado correctamente, se regresa el subconjunto ya actualizado a la base de datos.

1.3.3 Inferencia deductiva

En la recuperación de hechos y en la inferencia estadística los resultados se relacionan directamente con la consulta, de manera que, si se plantea una pregunta del tipo "Porqué el plomo afecta el desarrollo psiconeurológico del niño ? " no puede responderse en forma directa, ya que no existe ninguna eneada en el sistema que cumpla con dicho planteamiento, aun cuando exista un vínculo entre ambos argumentos de busqueda a través de muchas relaciones intermedias, que es lo que hace la inferencia deductiva; es decir, puede recuperar información en forma indirecta. El problema consiste en construir el vínculo entre los argumentos de busqueda y la meta. Cuando el número de vínculos crece demasiado, el número de combinaciones posibles resulta inmenso y es necesario aplicar técnicas de inteligencia artificial que reduzcan las posibilidades de busqueda a un número manejable.

Una base de datos empleada para la inferencia deductiva requiere de una categorización semántica muy fina, con el objeto de reducir el número de vínculos que se vayan a explorar, ya que sin

tales reglas resultaría muy costoso el hacer busquedas, en donde el programa de procesamiento deductivo de consulta tiene que considerar el significado del vínculo a fin de seguir la ruta adecuada. Cuando la base de datos no está acotada, el algoritmo de busqueda recolectará eneadas que contengan llaves que coincidan con la consulta, y después con base en los valores encontrados en esas eneadas se seleccionarán otras eneadas, de forma tal que el proceso de busqueda requiera de retroalimentación por parte del usuario para ir limitando la busqueda hasta que se obtenga el resultado desado. Mientras que, si la base de datos está acotada, la consulta encontrará sucesores siguiendo apuntadores, debido a que con un conjunto acotado se puede tener un buen seguimiento de la informacion.

El proceso de consulta tiene que evaluar la probabilidad de éxito al seguir una ruta específica. La base de datos puede construirse haciendo reglas que representen funciones o cuadros, entidades, o una combinación de ambos. También puede ser conveniente proporcionar una medida de peso o fuerza entre los cuadros, como lo sería el describir la frecuencia de ocurrencia de un evento como: algunas veces, casi siempre, etc.

Para el manejo de inferencia deductiva es importante contar con el apoyo de sistemas de memoria virtual, de manera que el área de trabajo pueda mantenerse sin requerir demasiado de la paginación. Sin embargo, todavía no es clara la forma en que los sistemas de bases de datos puedan apoyar a las necesidades planteadas por los sistemas de inferencia deductiva que utilizan millones de eneadas.

1.4 ESTRUCTURAS DE ARCHIVOS SIMPLES

Un archivo es un conjunto de registros comunmente descritos por la misma definicion de archivo.

A pesar de que el archivo es un concepto logico mas que fisico algunas estructuras de archivo resultarian ineficientes al implementarse en un medio de almacenamiento particular.

Se conoce como estructura de archivo a la forma en que los registros estan organizados para ser accesados desde el archivo; asi como a ciertos tipos de relaciones que existen entre los registros del archivo.

En la Tabla 1-A se muestra la jerarquia de la informacion

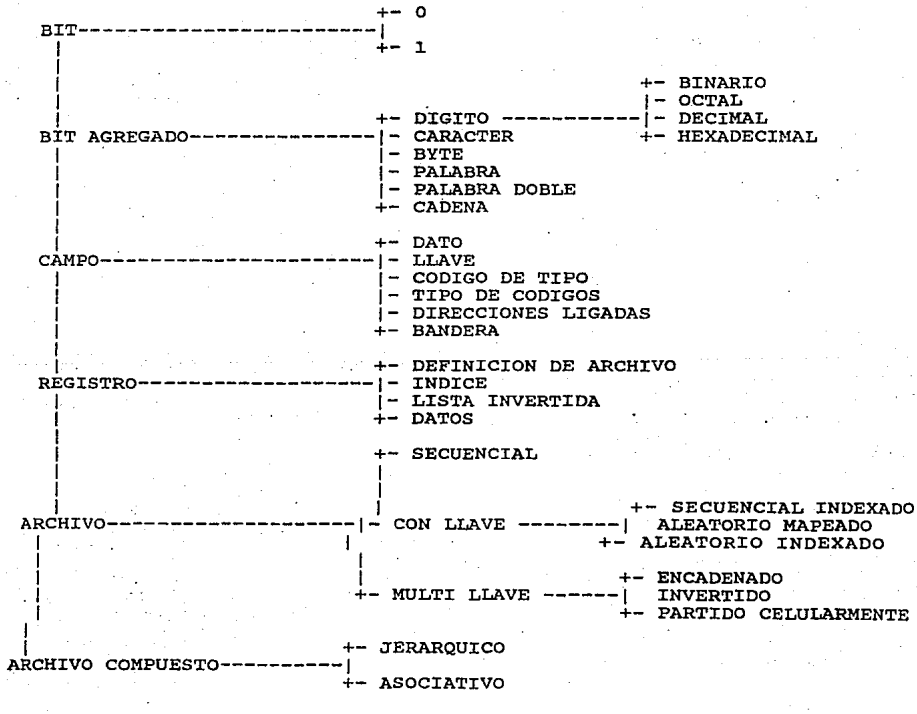


TABLA 1.A Jerarquia de la informacion

En lo sucesivo utilizaremos un sub-lenguaje para la manipulacion de datos, el cual contendra los cinco siguientes comandos basicos de entrada/salida:

OPEN, GET, SEARCH, PUT y CLOSE.

Existen siete estructuras de archivo basicas que permiten el acceso a la informacion de diferente manera, las cuales se explicaran a continuacion:

1.4.1 Estructura de archivo secuencial

Este tipo de organizacion es el unico que impone un estricto requerimiento fisico, que consiste en que los registros de un archivo deben estar fisicamente contiguos. La figura 1.2 muestra la organizacion de archivo secuencial.

El acceso a un archivo secuencial comienza con el primer registro contenido en el primer bloque. Con el primer GET realizado, despues del OPEN correspondiente, cada GET que se efectue, encontrara el siguiente registro del archivo almacenado.

Si estuvieran almacenados N registros por bloque, bastaria que el dispositivo leyera una vez para recuperar N registros, esto es gracias al almacenamiento intermedio (buffer) que se utiliza para el intercambio de informacion, es decir, que para que los N registros se encuentren en ese bloque, bastara una lectura del dispositivo y las consiguientes operaciones GET se haran sobre la memoria.

Si el archivo se encuentra ordenado conforme a alguna llave y se necesita agregar informacion, entonces la unica manera de mantener consecutivo fisicamente al archivo, es copiando desde el principio hasta donde se deba insertar el nuevo registro, agregar el nuevo y finalmente copiar el resto del archivo original. Si el archivo no esta ordenado, simplemente la insercion de un nuevo registro se efectua al final del archivo.

Un archivo secuencial se utiliza principalmente en el procesamiento serial de los registros; es decir, cuando se quiere procesar un archivo empezando con el primer registro y despues procesar cada registro en secuencia.

El uso de los archivos secuenciales hoy en dia se asocia grandemente con el uso de la cinta magnetica.

Los archivos secuenciales se utilizan con dispositivos de almacenamiento de acceso directo en aplicaciones que requieren gran velocidad en el acceso de registros sucesivos.

1.4.2 Estructura de archivo secuencial indexada

Como se aprecia en la figura 1.3, cuando el archivo se accesa via la llave, esta se somete al indice, cuyo proposito es traducir la llave en una direccion de bloque.

En principio, la llave no necesita ser unica, aunque en la practica normalmente asi es.

Para insertar un registro nuevo al archivo se utiliza la propiedad aleatoria del dispositivo de almacenamiento para minimizar el tiempo de actualizacion y al mismo tiempo para mantener la organizacion secuencial del archivo. El bloque en donde se va a insertar el registro nuevo se encuentra al decodificar el indice de la primera llave que lexicograficamente sea mayor que la llave del registro nuevo. Se accesa el bloque indicado, y si existe un espacio reservado suficiente dentro del bloque, el registro nuevo se inserta en su propia posicion dentro del bloque. En caso de no existir espacio suficiente para acomodar el registro, el bloque se extiende logicamente en una region de sobrepaso (overflow).

En esta organizacion de archivos la insercion de registros nuevos usualmente no se hace al final del bloque, por el contrario, se insertan en la posicion apropiada dentro del bloque, y los registros subsecuentes son colocados al final del bloque.

Estas operaciones no son ineficientes debido a que se ejecutan en el medio de almacenamiento.
Cuando el archivo se almacena secuencialmente, la busqueda del registro que contiene la llave, se hace explorando el bloque hasta que encuentra dicho registro.
El uso principal de esta organizacion de archivo es para accesar aleatoriamente registros con llaves no seriales.
Los valores de la llave por si mismos no tienen que ser seriales debido a que el mecanismo de acceso a los registros no utilizan la llave por si misma como un localizador. La llave puede ser alfanumerica y de longitud variable.

Antes de continuar es necesario incluir el concepto de dos caracteristicas de llaves y archivos que son:

1) Llave unica contra llave generica.- Una llave unica, es aquella para la cual solo existe un registro en el archivo que contenga ese valor. Como en el caso del numero de empleado de una institucion. Mientras que una llave generica puede representar mas de un registro, como seria el caso de la edad de una persona.

2) Registro con llave unica contra registro con multiplicidad de llaves.- Registro con llave unica es aquel que contiene una y solo una llave, mientras que el registro con llaves multiples es aquel que contiene mas de una llave.

1.4.3 Estructura de archivo aleatoria indexada

Esta estructura difiere de la organizacion secuencial indexada en dos aspectos. Primero todos los registros anadidos recientemente se colocan al final del archivo fisico o dondequiera que el procesador de mantenimiento de espacio indique un espacio disponible. Segundo, cada llave del archivo debe aparecer en el indice, debido a que cada llave en el indice apunta directamente al bloque en el cual esta contenido el registro con esta llave. Por esta razón se le llama indice completo.

El archivo no necesita mantenerse fisicamente en secuencia. Este se accesa traduciendo una llave via el indice en una direccion de bloque. A este metodo se le conoce como HASH. Dentro del bloque se encontrara el registro que contenga dicha llave. La llave debe ser unica, debido a que solo puede haber una posicion para el registro dentro del archivo y cada registro debera contener al menos una llave.

En la figura 1.4 se muestra la organizacion de archivo aleatoria indexada.

Un archivo aleatorio indexado solo puede leerse secuencialmente explorando el indice y despues leyendo aleatoriamente los registros del archivo, tal y como lo indique el indice.

Generalmente se desea que en los sistemas con multiplicidad de llaves se separen los indices de la llave tanto fisicamente construyendo indices multiples, indices diferentes o prefijando cada llave en unas series identificadas por una letra distintiva. Por ejemplo que el numero de RFC empiece con la letra "R" y el número de empleado empiece con la letra "E", de manera que cada tipo (en este caso "R" y "E") se agrupen en diferentes partes del índice. Esto permite que los registros del archivo sean accesados secuencialmente por cada uno de los tipos de llave, aun cuando no esten almacenados en secuencia fisica.

La organizacion de archivo aleatoria indexada se utiliza principalmente en el acceso aleatorio de registros con llaves unicas, con llaves multiples o en situaciones donde el porcentaje de actualizacion es tan alto que se pueden desarrollar encadenamientos excesivos para mantener el archivo en secuencia.

La mayor ventaja de la organizacion de archivo secuencial indexada sobre la organizacion aleatoria indexada es la habilidad del primero para proveer acceso secuencial eficiente comenzando desde cualquier punto del archivo. Asimismo, el indice es menor, debido a que es parcial.

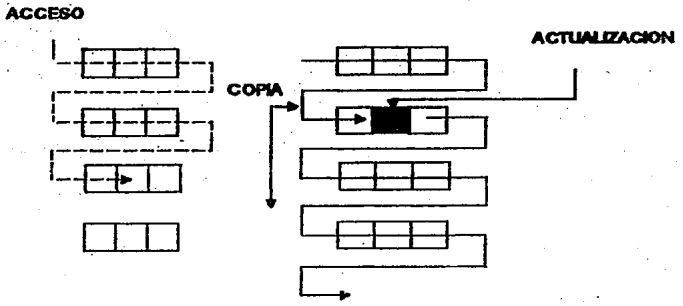


Figura 1.2 Organizacion de archivo secuencial

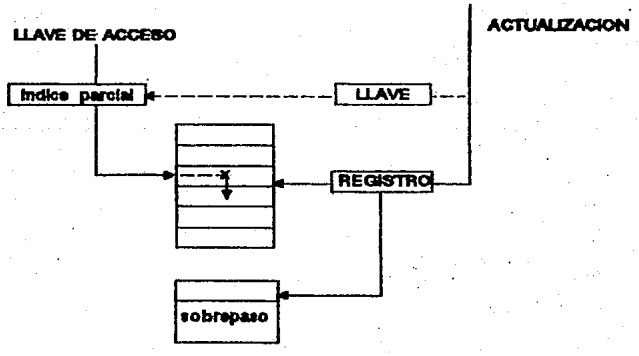


Figura 1.3 Organizacion de archivo secuencial indexada

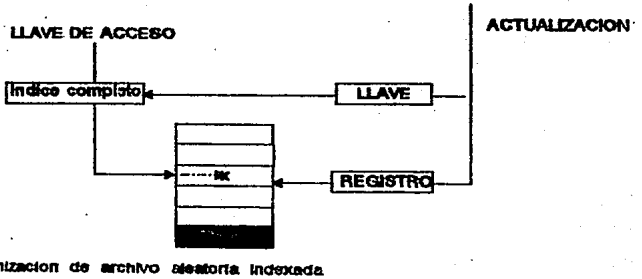


Figura 1.4 Organizacion de archivo aleatoria indexada

1.4.4 Estructura de archivo aleatoria mapeada

Esta estructura de archivo fue desarrollada con el objeto de eliminar el requerimiento adicional de almacenamiento y a su vez pretende aligerar el problema de acceso.

Los registros de datos del archivo se almacenan aleatoriamente, pero el indice es reemplazado por un mapa, el cual es una transformacion algoritmica de la llave del registro a la direccion de bloque.

El mapa ejecuta la misma transformacion llave-a-direccion que el indice, pero existe una diferencia funcional entre la estructura aleatoria mapeada y las estructuras indexadas, la cual consiste en que no es posible leer el archivo mapeado en el orden secuencial de su llave, debido a que no hay una ennumeracion indexada de las llaves y que los registros se almacenan en forma aleatoria. Otra diferencia es que resulta imposible construir un mapeo que asegure la transformacion de dos valores llave diferentes a una misma direccion de bloque, por lo que un registro solo puede tener una llave tnica.

Eventualmente puede ocurrir que se mapee a un bloque que ya haya sido llenado, en cuyo caso se debe hacer una extension logica de este bloque, que seria de la misma forma que para la extension logica del bloque secuencial indexado. A esta forma de extension logica de un bloque se le llama una cadena.
La llave generica tambien puede ser manejada mediante esta estructura de archivo, debido a que simplemente es otro caso de mapeo de muchos a uno.

La estrategia de acceso de un registro principia con la transformacion mapeada de la llave en una direccion de bloque. El archivo se actualiza mapeando la llave del registro nuevo en una direccion de bloque. Si el bloque esta desocupado o esta ocupado, pero todavia hay espacio dentro de el, entonces ahi se almacena el registro. Cuando no hay espacio dentro del bloque debe hacerse una extension logica para crear un bloque encadenado.

La organizacion de archivo aleatoria mapeada se utiliza principalmente en el acceso aleatorio de registros de longitud fija con llaves seriales unicas o para el acceso aleatorio de registros de longitud fija o variable con llaves genericas no seriales.
Una desventaja del mapeo es que el archivo no puede ser accesado secuencialmente via los valores de la llave mediante una lectura serial de los bloques del archivo de datos, debido a que estan almacenados aleatoriamente. Ademas no pueden ser accesados secuencialmente de manera conveniente a traves del mapeo, ya que todas las llaves posibles tendrían que ser generadas, mapeadas y examinadas para que se encontraran en los registros accesados.

La tecnica no puede ser utilizada con registros de llaves multiples, a menos que se utilice un nivel indirecto de direccionamiento entre el mapeo y el archivo de datos que

traduzca las transformaciones de la direccion llave de muchas-a-muchas en una transformacion de muchas-a-una; pero debido a que el nivel de direccionamiento indirecto interpuesto es equivalente a un indice, generalmente resulta mas efectivo utilizar un metodo indexado cuando se trata de registros con llaves multiples.

La figura 1.5 muestra un esquema de la organizacion aleatoria mapeada para archivos.

El diagrama muestra que la estructura de archivo aleatoria indexada puede construirse concatenando las organizaciones de los archivos secuencial indexada y aleatoria mapeada. El archivo secuencial indexado traduce la llave del registro en un numero serial que corresponda a la llave. Este es alimentado a una organizacion mapeada para traducirlo a un numero de bloque en el archivo de datos. Por lo tanto esta combinacion actua como una estructura de archivo aleatoria indexada, con la excepcion de que las direcciones del archivo de datos no pueden ser asignadas libremente.

La utilidad principal de las estructuras de archivo secuencial indexada, aleatoria mapeada y aleatoria indexada es almacenar y accesar registros con una sola llave unica, aunque algunas de ellas pueden adaptarse para manejar registros con multiplicidad de llaves y para llaves genericas. La estructura secuencial indexada maneja llaves genericas, pero no registros con multiplicidad de llaves. La estructura aleatoria indexada o aleatoria mapeada maneja registros con multiplicidad de llave pero no llaves genericas.

Para manejar registros con llave generica y con multiplicidad de llaves simultáneamente, se han desarrollado tres tipos de estructuras de archivo que son las siguientes:

1) Lista encadenada
2) Lista invertida
3) partido celularmente

1.4.5 Estructura de archivo de lista encadenada

La metodologia empleada es una extensión del concepto de la estructura de archivo aleatoria indexada, como se puede apreciar en la figura 1.6.

Para simplificar el diagrama solo se muestran los registros lógicos. Dentro del registro cada llave genèrica está asociada con un campo de dirección liga, cuyo propósito es apuntar al siguiente registro en el archivo que contenga el mismo valor llave.

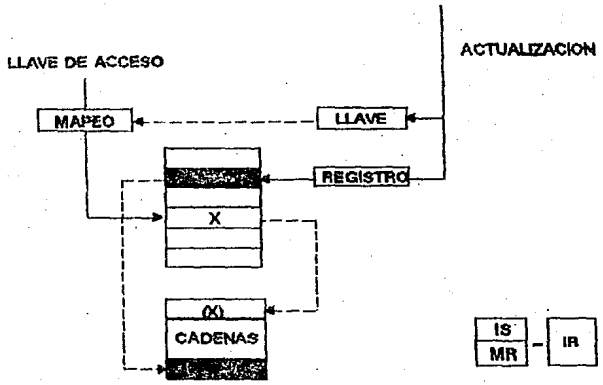


Figura 1.5 Organizacion de archivo aleatoria mapeada

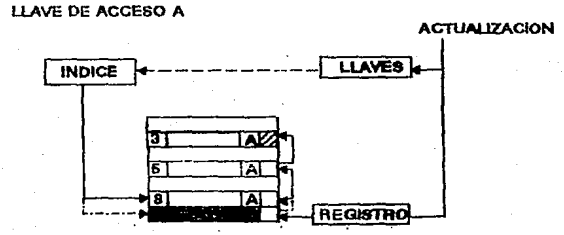


Figura 1.6 Organizacion de archivo de lista encadenada

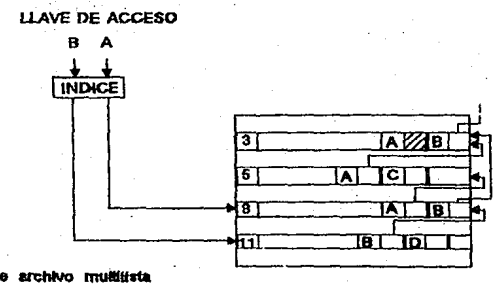


Figura 1.7 Organizacion de archivo multilista

Cuando los registros tienen llaves múltiples y las llaves son genéricas, entonces se construye una cadena de este tipo para cada llave en cada uno de los registros de llave múltiple, tal y como se muestra en la figura 1.6.

A una estructura de lista encadenada que maneja tanto registros con llaves genéricas, como registros con llaves múltiples se le llama organización de archivo multilista. Este tipo de organización se muestra en la figura 1.7.

La actualización de un archivo de lista encadenada está en función de como se accesa el archivo para añadir un registro. El procedimiento normal es añadir un registro nuevo al inicio de la cadena y mantener una sola liga unidireccional. Cuando el registro se añade al principio de la cadena como se muestra en la figura 1.6, la dirección del registro nuevo se convierte en la dirección liga que emana del índice, (mostrada por una extensión de línea punteada que sale del índice), y la dirección de la cabeza anterior de la lista se convierte en la dirección liga del registro nuevo; por lo tanto, el registro nuevo reemplaza al anterior como cabeza de la lista.

Cuando se recorre la cadena con el propósito de accesar o recuperar, se observa que el último registro que se añade a la cadena será el primero en recuperarse; por lo tanto este procedimiento sigue el método de recuperación Last-In-First-Out (LIFO).

Si se quisiera mantener simultaneamente los controles LIFO y FIFO (First-In-First-Out), entonces se emplearía la inserción al principio o al final del archivo, teniéndose que mantener en el índice ambas direcciones, la de principio y fin de lista. Además los registros deben tener direcciones de liga bidireccionales para que la lista pueda recorrerse en ambas direcciones.

Cuando se introduce un registro nuevo al archivo multilista, resulta necesario actualizar el índice con la llave genérica del registro nuevo. Si la llave es única se requerirá la inclusión de una llave nueva al índice, si la llave es genérica, puede ser nueva, en cuyo caso debe añadirse al índice, si existe en el índice, solo se actualiza la indicación de longitud de lista y se añade el registro a la lista apropiada.

1.4.6 Estructura de archivo de lista invertida

La organización de archivo de lista invertida se ilustra en la figura 1.8.

En este método se utiliza un archivo intermedio entre el índice y el archivo de datos. Este archivo intermedio contiene registros que son simplemente listas de apuntadores a los registros del archivo de datos. Cada llave en el índice apunta a una de estas listas llamadas lista invertida, y esta a su vez apunta a todos los registros del archivo de datos que contengan dicha llave

genèrica. En la figura 1.8 la llave A se traduce a través del índice para apuntar a un registro particular de la lista invertida, cuyo registro ennumera y apunta a tres registros que son el 3, 5 y 8. El apuntador contenido en la lista invertida puede ser una dirección liga o una llave de registro único. La dirección liga puede ser una dirección de bloque, o una dirección de registro lógico. Este último tipo de dirección normalmente se utiliza cuando se emplea lógica multillave, debido a que precisamente coordina los registros que se van a accesar.

La decodificación de bloque o registro para lógica multillave se ejecuta accesando todas las listas invertidas que corresponden a varias llaves en la expresión lógica y después ejecutando la lógica indicada sobre estas listas. Por ejemplo si la lógica requiere accesar todos los registros que contengan las llaves A y B, entonces la lista invertida con las direcciones de A se intersectarán con las de la lista B. Para acelerar este proceso, la lista invertida mantiene en secuencia las direcciones.

La única ventaja del direccionamiento por bloque radica en el uso de listas invertidas para sistemas donde no se emplea la lógica multillave. Actualmente el direccionamiento por bloque es un tipo especial de particiòn celular.

El uso de una llave de registro único es una alternativa para aquel de una direccion de registro.

La ventaja de utilizar una llave primaria como apuntador es que la relocalización de un registro con un cambio consecuente en su dirección no necesitará una actualización de las listas invertidas en las que aparece el registro. Solamente se necesita actualizar el índice de dirección del registro a la llave primaria.

Una aproximación alternativa para el almacenamiento de direcciones o apuntadores llave es la generación de series de mapas de bit, una por llave en el sistema. Cada mapa mantiene tantos bits como registros en el archivo. Por lo tanto, debe establecerse un tamano màximo de archivo o los programas que generan y mantienen estos mapas deben ser capaces de expandir o condensar dinàmicamente el tamano del mapa. Si el registro N contiene la llave X, entonces el mapa para la llave X tendrá en el enèsimo bit un uno. La ventaja de la aproximación del mapa de bit es la velocidad de actualizaciòn.

Una comparación de su velocidad de procesamiento con aquella de listas es una función del tamano de lista y archivo, donde los archivos pequenos están a favor del mapa de bits y las listas pequenas a favor de la aproximación de lista.

La comparación del requerimiento de almacenamientos es una función del radio de la longitud de la lista promedio al tamano de archivo (en registros), y la sobrecarga se lee:

B= Número de bits en la palabra de computadora

P= Radio de la longitud de la lista promedio al tamano de archivo.

F= Tamano de archivo

Suponiendo que la direcci6n de registro requiere de una palabra de computadora. Puesto que, el n6mero de palabras requerido para la lista invertida de tamano promedio es PF y el n6mero de palabras requerido para un mapa de bits es igual a F/B, el almacenamiento de mapa de bit sera menor que el almacenamiento de lista cuando:

$$F/B < pF$$

o

$$1/B < p$$

El almacenamiento de mapa de bit sera menor que el almacenamiento de lista cuando el radio de la longitud de la lista promedio al tamano de archivo es mayor que el recíproco del n6mero de bits por palabra de computadora.

Como se indica en la figura 1.8, los registros nuevos se anaden al final del archivo de datos, y la identificaci6n del registro sea esta un apuntador de direcci6n o una llave primaria, se inserta en secuencia en la lista invertida.

En el caso de la multilista, todas las llaves nuevas deben anadirse al índice, pero normalmente no es necesario mantener una cuenta de la longiutd de la lista, debido a que esto se requiere para optimizar la estrategia de recuperaci6n, utilizando solo la técnica multilista.

Cuando se designa una llave 6nica como identificador del registro para utilizarla como un apuntador de lista invertida o para controlar la secuencia y el acceso a un archivo secuencial indexado, se le llama "llave primaria" del archivo. En archivos de llaves múltiples a las demás llaves se les llama llaves secundarias. El archivo de datos en la organizaci6n de archivo encadenado o multilista puede indexarse aleatoriamente, con ambas llaves primaria y secundaria apareciendo en el mismo índice, o puede ser secuencial indexado, en cuyo caso solo aparece la llave primaria en el índice del archivo. Un archivo secuencial indexado adicional traduce todas las llaves secundarias en primarias.

La mayor ventaja de una organizaci6n de lista invertida es que el acceso al archivo puede hacerse eficientemente mediante la investigaci6n de las multillaves, debido a que las operaciones lógicas como la intersecci6n y la uni6n pueden ejecutarse fácilmente sobre estas listas para producir una lista resultante de apuntadores de acceso.

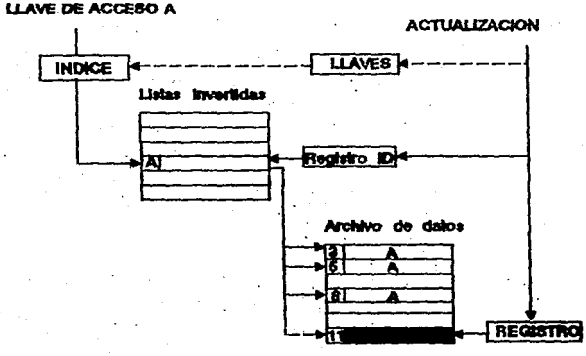


Figura 1.8 Organizacion de archivo de lista invertida

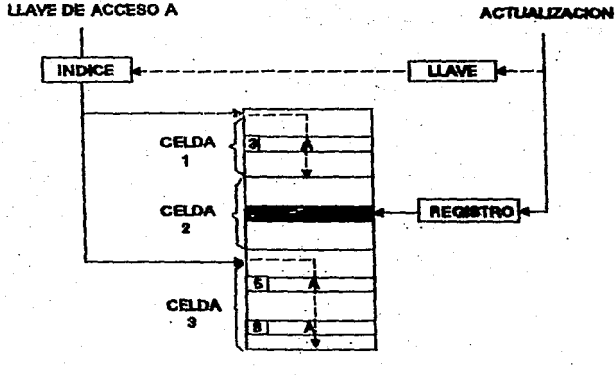


Figura 1.9 Organizacion de archivo partido celularmente

1.4.7 Estructura de archivo partido celularmente

Este método es equivalente al método de lista invertida con apuntadores de dirección de bloque, excepto que los apuntadores hacen referencia a grandes particiones contiguas de almacenamiento DASD llamado celdas.

Las celdas pueden ser de tamaño arbitrario, o pueden estar relacionadas con alguna partición física del dispositivo.
La celda puede estar organizada internamente por cualquiera de los otros mecanismos de estructura de archivo; pudiendo ser secuencial, indexado, o mapeado. El ejemplo de la figura 1.9 muestra la celda organizada secuencialmente. El índice decodifica la llave A a las celdas 1 y 3. La celda 1 es leída secuencialmente, y se recupera un bloque que contiene al registro 3, conteniendo la llave A; la celda 3 tiene 2 registros, el 5 y 8 que contienen la llave A.

El propósito principal de la partición celular es reducir el tamaño y mantenimiento del índice. Este reside en que una extensión mayor en el procesamiento dentro y fuera de las celdas puede verse como una jerarquía de organizaciones. Cuando la suborganización celular es secuencial, el tamaño de la celda determina un estricto trato de espacio-tiempo entre el tamaño y procesamiento del índice contra el tiempo de procesamiento en el archivo de datos por sí mismo.

Cuando se actualiza la partición celular con un registro nuevo, el primer intento es colocar el registro dentro de una célula que ya contiene todas las llaves del registro, de manera que el índice no tiene que ser actualizado adicionalmente. Faltando esto, el registro se coloca en una celda disponible que contiene la mayoría de las llaves del registro. Con este principio es posible determinar un esquema de distribución para almacenamiento de registros que minimizaría el tamaño del índice distribuyendo registros a celdas basadas en el mayor grado de ocurrencia mutua de llaves en ese registro con llaves de otros registros están en la misma celda. Por ejemplo, si todos los registros con la llave A puedieran reunirse en una llave, habría un solo índice de referencia a la celda con la llave A, y un acceso a la celda produciría todos los registros con la llave A. Además existen otras llaves dentro de esos registros que tendrían atracciones similares a otras celdas, y fuera de esto el proceso completo no es diferente de un programa dinámico de optimización.

La tabla 1.B presenta una matriz que correlaciona los atributos de archivo con la organización de archivo. Las siete estructuras u organizaciones se clasifican en la tabla como cinco, donde las tres estructuras de lista se han agrupado en una categoría llamada LISTA. Se han indicado tres grandes categorías de atributos de archivo, que son:

1) Sin llave
2) Con llave
3) Longitud del registro

ATRIBUTOS DE ARCHIVO									ORGANIZACION DE ARCHIVO				
NO	LLAVEADO						Long. del registro		S	SI	AI	AM	LISTA
LLAVEADO	UNICO	GENERICO	SERIAL	NO SERIAL	SIMPLE	MULTIPLE	FIJA	VARIABLE					
X								X	X				
	X			X	X			X		X	X	X	
	X			X		X		X			X		
	X		X		X		X					X	
		X		X	X			X		X		X	
		X		X		X		X					X

TABLA 1.8 Atributos de archivo comparados con las organizaciones de archivo

La categoría con llave se subdivide en tres subconjuntos, cada uno de los cuales tiene dos alternativas:

1) llave única y genérica
2) Llaves seriales y no seriales
3) Registros de una sola llave y de llaves múltiples.

La longitud de registro se indica como fija o variable.

En el lado derecho de la tabla se indica con una X la organización de archivo más apropiada para un conjunto de atributos de archivo que está indicado por una X en el lado izquierdo.

La cuarta línea de la tabla indica que un sistema con una llave serial única, con una llave simple, y de longitud fija (o un numero fijo de registros por bloque) puede ser implantada apropiadameante por un archivo aleatorio mapeado con mapeo de identidad.

CAPITULO 2

Sistemas Basados en Archivos Invertidos

2.1 Definición de archivo invertido y sus ventajas

Virtualmente la mayoría de los sistemas comerciales se basan en diseños de archivos invertidos. Es decir, cada sistema lógicamente contiene un archivo de documentos y uno o mas directorios auxiliares conocidos como índices invertidos, que permiten un rápido acceso a los términos indexados. En el índice se incluye para cada término una lista que asocia números a documentos, donde cada número especifica el documento en el cual aparece ese término. Por lo tanto, cuando se recuperan los documentos identificados por un término arbitrario, lo que realmente se esta haciendo es consultar los archivos de índices, para obtener los números de referencia a documentos. Finalmente los documentos identificados se toman del archivo de documentos.

2.2 Operadores auxiliares

2.2.1 Expresiones Booleanas

Generalmente cuando se hace una busqueda se obtienen como resultado un gran número de documentos que tratan el tema en forma general y que sería muy caro e inútil consultar; siendo necesario afinar la busqueda hasta obtener unos cuantos artículos tan específicos como se quiera. Esto se logra procesando la información que contienen los índices invertidos, haciendo uso de la lógica booleana para construir preguntas que contienen varios términos ligados por los operadores booleanos AND, OR y NOT. Estos operadores son implementados para el manejo de las operaciones con conjuntos de intersección, unión y diferencia.

Una pregunta que nos permite obtener los documentos que tratan sobre CIENCIA y TECNOLOGIA sería:

CIENCIA AND TECNOLOGIA

A continuación se presenta el procedimiento que se lleva a cabo internamente:

1) Utiliza el índice invertido para recuperar los números de referencia a documentos asociados con el término CIENCIA, y llama a este conjunto de números de referencia el conjunto 1. Ver la figura 2.1 (a).

2) Utiliza el índice invertido para recuperar los números de referencia a documentos asociados con el término TECNOLOGIA, llamando a este conjunto de números el conjunto 2. Ver la figura 2.1 (b).

3) Determina el conjunto de números de referencia a documentos que constituyen la intersección de los conjuntos 1 y 2; es decir, que pertenecen tanto al conjunto 1 como al conjunto 2, y lo llama el conjunto 3. Ver la figura 2.2.

4) Toma los numeros de referencia del conjunto 3 y con ellos recupera los documentos del archivo de documentos.

Con la pregunta CIENCIA o TECNOLOGIA se quieren obtener aquellos documentos identificados por el término CIENCIA, o por el término TECNOLOGIA, o por ambos términos. Los conjuntos 1 y 2 se forman de la misma manera que para el operador AND. Los conjuntos 1 y 2 se combinan en un nuevo conjunto 3, el cual contiene los números de referencia a documentos que pertenecen al conjunto 1, al conjunto 2 y los que se encuentran en ambos conjuntos (conjunto unión, figura 2.3). Finalmente, del archivo de documentos se recuperan todos los documentos identificados por el conjunto 3.

El operador NOT generalmente se utiliza para especificar que se quiere obtener algun término en particular en los documentos recuperados, pero que a su vez impida la presencia de otros términos.

Si queremos obtener documentos que traten de CIENCIA, pero no de TECNOLOGIA, la pregunta se plantearía como:

CIENCIA NOT TECNOLOGIA

Y el procedimiento sería:

1) Los conjuntos 1 y 2 se forman de la misma manera que para los operadores AND, OR. Ver la figura 2.1 (a) y (b).

2) Del conjunto 1 elimine todos aquellos números de referencia a documentos que aparecen en el conjunto 2; es decir, se construye el conjunto diferencia entre los conjuntos 1 y 2. Ver la figura 2.4.

3) Utiliza el archivo de documentos para recuperar los documentos indicados por los números de referencia que pertenecen al conjunto 1.

La mayoría de los lenguajes de consulta tienen la capacidad de combinar argumentos de búsqueda mediante los operadores booleantos AND y OR. Una consulta que incluya un operador NOT no tendrá mucha fuerza de partición para los archivos, por lo que su empleo esta muy limitado. Los operadores de comparación diferentes de igual (=); es decir (<,<=,<>,>,>=), también tienen muy poca fuerza de partición. Cuando en las expresiones aparecen operaciones de comparación se define un subconjunto restringido del archivo. Las expresiones complejas no deben ejecutarse en un solo paso, sino que deben analizarse primero para considerar las facilidades de acceso que se tengan y determinar la forma en que el archivo se puede particionar mejor.

Por ejemplo, resulta conveniente combinar las dos subexpresiones en la consulta:

(SALARIO > 150,000) .AND. (SALARIO < 250,000)

Figura 2.1 a) Documentos asociados con el termino CIENCIA
b) Documentos asociados con el termino TECNOLOGIA

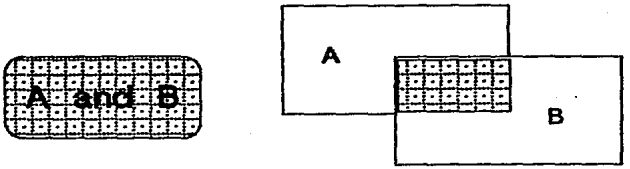


Figura 2.2 Documentos asociados con los terminos CIENCIA Y TECNOLOGIA

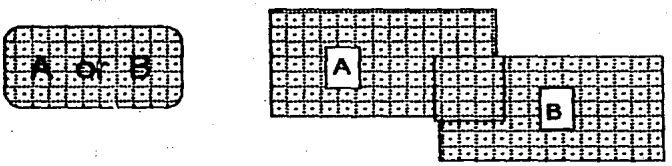


Figura 2.3 Documentos asociados con los terminos CIENCIA O TECNOLOGIA

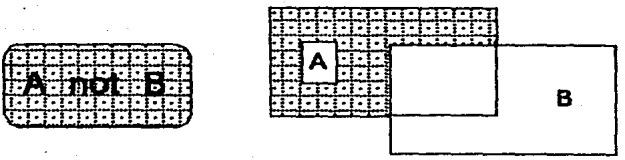


Figura 2.4 Documentos asociados con el termino CIENCIA y NO asociados con el termino TECNOLOGIA

en vez de aplicar las expresiones por separado. El empleo de operadores de comparación favorece el indexado para acceso directo, ya que se requieren búsquedas en serie.

2.2.2 Jerarquia de los operadores

La complejidad de una pregunta puede aumentar conforme se van anadiendo operadores, por lo que resulta necesario hacer reglas que aseguren que las preguntas van a ser interpretadas correctamente por el sistema recuperador.

Por ejemplo, se tiene un archivo índice como el siguiente:

Términos	Números de referencia a documentos
Automóvil	1 2 3 8
Camioneta	2 3 4 5 9
Bicicleta	4 6 8

y se tiene una pregunta como:

AUTOMOVIL AND CAMIONETA OR BICICLETA

Esta pregunta resulta ambigua, debido a que si empezamos a analizarla de izquierda a derecha tenemos que el resultado de hacer la intersección entre automóvil y camioneta es 2 3 y haciendo la unión de este resultado con bicicleta, se tiene que los números de documentos resultantes son: 2 3 4 6 8.

Mientras que, haciendo el análisis de derecha a izquierda se tendria:

Que la unión de camioneta con bicicleta daría 2 3 4 5 6 8 9 y la intersección de este conjunto con automóvil nos daría como números de documento resultantes 2 3 8.

Como puede verse, el orden en el que se ejecutan las operaciones es crítico.

La estrategia puede ser de izquierda a derecha o de derecha a izquierda; o bien, puede especificarse otro orden en el cual las operaciones se van a ejecutar. Como por ejemplo que todos los operadores OR se ejecuten primerio, seguidos por los operadores AND y finalmente por los operadores NOT. Los operadores equivalentes se van a ejecutar de izquierda a derecha.

Los paréntesis generalmente se utilizan para esquivar el orden estricto que se menciono anteriormente. Las operaciones que se encuentran dentro de los paréntesis son las primeras en ejecutarse.

A continuación se presenta el equivalente con parèntesis del ejemplo usado anteriormente, el cual presentaba el orden de izquierda a derecha.

(AUTOMOVIL AND CAMION) OR BICICLETA

Una vez que se ha establecido la regla de los parèntesis, esta puede aplicarse una y otra vez; es decir, los parèntesis pueden ir anidados, ejecutandose primero las operaciones del par de parèntesis màs interno.

Para permitir parèntesis es necesario conservar resultados intermedios. Por esta razòn algunos sistemas no permiten el uso de parèntesis y otros lo permiten pero con un limite de anidamiento de parèntesis.

2.2.3 Adyascencia

Cada sistema comercial incluye ciertas caracterìsticas que lo hacen ùnico. Con lo que se complica el aprendizaje de cada sistema.

Cuando un sistema permite hacer bùsquedas con tèrminos incluìdos en textos de documentos, resulta de gran utilidad especificar que dos tèrminos deben aparecer juntos el uno del otro en un texto y en el orden establecido. Si llamamos al operador adyascencia como ADJ, y queremos obtener los documentos que traten sobre INVERSION TERMICA, la pregunta se plantearìa como:

INVERSION ADJ TERMICA

La probabilidad de que el concepto INVERSION TERMICA estè contenido en el documento es mayor que si la busqueda se hiciera como: INVERSION AND TERMICA

Es difìcil implantar la operaciòn adyascencia utilizando la definiciòn bàsica de archivo invertido.

Para llevar a cabo la funciòn de adyascencia existen varios mètodos:

METODO 1:

1) Este primer mètodo requiere de un archivo invertido para identificar los documentos que satisfagan la pregunta

   INVERSION AND TERMICA

2) Posteriormente consulte el archivo de documentos para comparar caracter por caracter dentro de campos que fueron preespecificados y asi detectar la presencia de los caracteres INVERSION TERMICA.

3) Recupere aquellos documentos en donde por lo menos se encontró una vez el conjunto de caracteres buscado.

Este método ha sido implementado en algunos sistemas, pero solamente se utiliza cuando es muy necesario, debido a que es un proceso laborioso. Cuando se utiliza este método, el sistema previene al usuario sobre la ineficiencia del proceso. Además de que existen restricciones que limitan las condiciones bajo las cuales se va a llevar a cabo la busqueda.

METODO 2:

Este método considera un archivo invertido mejorado, el cual contiene información acerca de la posición de las palabras dentro de cada documento. Por ejemplo, si se tiene la siguiente expresión TERMICO (345 1 2 5), indica que el término TERMICO ocurre en el primer párrafo de la segunda oración y es la quinta palabra del documento 345.

Por lo tanto, si deseamos recuperar los documentos que cumplan con:

INVERSION ADJ TERMICA

Se recupera el documento 345 y se verifica que cumpla con INVERSION (345 1 2 5).

METODO 3:

Este método también considera un archivo invertido, donde a cada término se le asigna la posición que tiene con respecto al inicio del texto. Por ejemplo, TERMICO (345 13) indica que el término TERMICO ocurre 13 palabras después del inicio del documento 345. El documento 345 cumpliría con el planteamiento anterior si además el término INVERSION cumpliera con INVERSION (345 12). Los dos métodos anteriores son equivalentes en términos de recuperación.

El método 1 (búsqueda de caracter por caracter) resulta muy ineficiente. Quedando la alternativa de utilizar los métodos 2 y 3, a pesar de que estas requieren de almacenamiento extra para guardar la posición.

2.2.4 Frecuencia de la informacion

Una forma de mejorar un sistema de archivo invertido consiste en incluir información acerca de la frecuencia de ocurrencia de los términos individuales, ya que esto indica el grado de importancia que tiene cada término dentro del tema.

Si la información de frecuencia se va a recuperar, resulta conveniente incluirla en el archivo invertido.

En muchos sistemas el número de documentos en donde cada término ocurre se almacena en el archivo invertido. De esta manera el usuario puede saber rápidamente cuantos documentos podrá recuperar utilizando un término dado. Por ejemplo, el término INFORMATION tiene 53,504 ocurrencias en el banco ERIC del sistema DIALOG.

2.3 Procesamiento de preguntas

2.3.1 Planteamiento de consultas de informacion

El planteamiento de consultas de informacion es el proceso mediante el cual el usuario se comunica con la base de datos a través del sistema recuperador de informacion.

El diseno de un lenguaje de planteamiento de información depende de:

- Las necesidades y fundamentos del usuario
- El sistema recuperador de información
- La base de datos

Los usuarios varían desde usuarios ocasionales que desean recuperar hechos simples como predicciones de clima, tráfico de automóviles, etc., y para quienes resulta de gran importancia la conveniencia y la claridad, hasta especialistas que trabajan muchas horas con el sistema conociendo las capacidades y contenido de este, y para quien es de gran importancia la eficiencia y disponibilidad de herramientas para manipular los datos recuperados. También puede haber intermediarios entre los usuarios finales y el sistema; ayudantes que plantean consultas para obtener resultados que posteriormente se presentarán a los encargados de tomar decisiones, programadores que elaboran programas para procesar la información, personasd que recolectan datos y actualizan el contenido de la base de datos (capturistas), y finalmente, un administrador de la base de datos.

Antes de acceder a la base de datos, resulta de gran utilidad contar con un esquema, en donde se defina la estrategia de busqueda, como la mostrada en la figura 2.5.

2.3.1.1 Consultas interrogativas

Los sistemas interrogativos presuponen un conocimiento mínimo. Las preguntas presentadas por la computadora se resuelven en forma simple, ya sea:

- Contestando si o no
- Por elección múltiple
- Mediante la introducción de términos de identificación conocidos o mediante nombres de funciones.

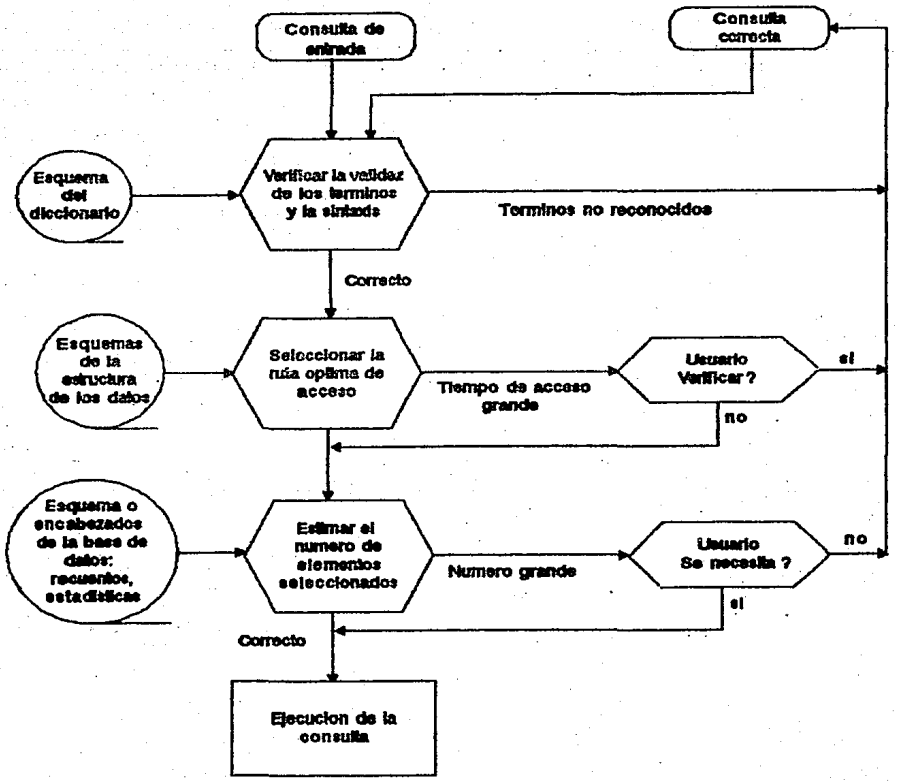


FIGURA 2.6 PLANTEAMIENTO DE UNA CONSULTA

En caso de que el usuario no comprenda los términos presentados por el sistema, se cuenta con una función de ayuda (función help). Este enfoque puede ser relativamente lento, pero resulta satisfactorio si no se utiliza durante largos períodos. Como ejemplos de este tipo de sistemas interrogativos están: Los sistemas de consulta de crédito y el de terminales bancarias de autoservicio.
Cuando se requiere un mayor flujo de datos, las terminales pueden interrogar al usuario en forma efectiva, presentando varias opciones en una interacción. El usuario puede seleccionar la opción mediante una pluma luminosa, una pantalla sensible al tacto, o tecleando el numero o letra correspondiente a dicha opción. Una selección puede llevar a la presentación de un conjunto de posibilidades más específica. Ver la figura 2.6.

La capacidad de comunicación que apoya a este tipo de planteamiento de consultas depende del equipo utilizado.

2.3.1.2 Consultas tabulares

Cuando el acceso al sistema es limitado, conviene preparar de antemano la consulta, utilizando para ello un cuestionario impreso. Cuando el usuario tiene un conocimiento adecuado sobre la base de datos, el planteamiento de la consulta no le tomará mucho tiempo. Una forma de consulta tabular contendrá nombres de atributo, y espacio para definir expresiones condicionales y de resultado simple. Comúnmente en el resultado se pueden tener varios niveles de subtotales y totales. Además se dispone de instrucciones que permiten agrupar y ordenar la información. En la figura 2.7 se presenta un ejemplo.

Cuando se requieren cálculos o criterios de selección más complejos, la estructura tabular rígida resulta inadecuada, utilizándose proposiciones y expresiones como las de los lenguajes de computación, en donde el planteamiento de consulta requiere de mayor conocimiento.

Para evitar la necesidad de conocer el contenido del esquema, se ha desarrollado una técnica que consiste en especificar consultas mediante ejemplos. En la figura 2.8 se presenta la consulta anterior, pero esta vez planteada mediante un ejemplo.

Para construir consultas mediante ejemplos debe existir interacción con el sistema . Primero se proporciona el nombre de la base de datos que se desea consultar, con lo que el sistema permite listar los elementos del esquema. El usuario puede seleccionar los elementos del esquema que desee formen parte de la consulta o de las respuestas.

En una consulta pueden incluirse varias relaciones.

Este sistema de consulta ha resultado muy práctico. Además mediante el empleo de ejemplos es posible definir, tanto el esquema de la base de datos, como los formatos de salida de los sistemas.

```
MICRO – QUESTEL                MAIN MENU              Date : 03/12/88

                0 – End

                1 – Search
                2 – Data entry
                3 – Indexing
                4 – Displays
                5 – Utilities
                6 – Down – loading
                7 – Auxiliary processing
                8 – Change of language

Number chosen :  0
```

Figura 2.6 Seleccion en pantalla

Solicitud de consulta

Titulo Verificacion de sueldos
Fecha 5 de Junio de 1996

Nombre del atributo	Seleccion		Impresion		Clasi – fica – cion	Total por atributo
	Op	valor	Col	Cond		
sueldo	>	80000.00	20			departamento
gerente	=	"S.Castro"				
empleado			1		2	
sueldo			30			
edad			40			
antiguedad			43			
cargo			48			

a)

Verificacion de sueldos

Aldana Ricardo	92,000.00	35	4	programador
Carrasco Luz Maria	95,000.00	29	1	analista de sistemas
Fernandez Rafael	123,000.00	24	1	consultor
Jimenez Silvia	80,000.00	26	5	secretaria
Mendez Carmen	90,000.00	23	1	Asesor
Narvaez Pedro	133,000.00	27	6	Administrador
Pallares Hector	162,000.00	32	7	consultor
Sanchez Jimena	80,000.00	35	8	capturista

b)

Figura 2.7 a) Planteamiento de una consulta tabular
b) Resultados de la consulta

2.3.1.3 Proposiciones interactivas de consulta

Los métodos interactivos de consulta son los que permiten mayor flexibilidad en el procesamiento de consulta. El usuario plantea una consulta utilizando un lenguaje de consulta de información. La proposición de consulta se analiza y ejecuta. La respuesta a una consulta generalmente provoca que el usuario plantee más consultas de manera que el resultado sea más específico. Se han desarrollado muchos lenguajes para el planteamiento de consultas. Estos lenguajes siempre están influídos por la estructura básica de la base de datos, por el grado de interacción en línea y por los dispositivos de comunicación empleados.

Aún cuando se emplean técnicas de lenguajes de procedimiento para el análisis y ejecución de las proposiciones de consulta, todavía no se ha aceptado un lenguaje de consulta común independiente del sistema. En la figura 2.9 se presenta un ejemplo de una consulta interactiva.

Este lenguaje y otros semejantes pueden tener interfaces con estructuras jerárquicas.

Cuando en una base de datos se requiere de muchos accesos a la base de datos para realizar operaciones complejas, algunos lenguajes de consulta permiten la creación de archivos de subconjuntos que pueden clasificarse y emplearse como entrada para procesos estadísticos.

Cuando se trata de una base de datos textual, las expresiones de consulta pueden complicarse. El procesador de consulta tiene que definir las palabras como secuencias de caracteres entre delimitadores, tales como espacios en blanco, y los símbolos (. , ; " '). Además como las palabras que aparecen en un texto tienen muchas variantes, es conveniente disponer de capacidad para el manejo de sinónimos y del reconocimiento de palabras que tengan la misma raíz. También al hacer la búsqueda se puede obligar a que las palabras aparezcan dentro de la misma frase o párrafo. En la figura 2.10 se presenta una consulta en texto libre.

Ponderación de términos.- Si además de querer obtener de una consulta todo lo referente al acero, queremos obtener otros productos que contengan aleaciones, entonces podrían asignarse pesos (ponderar) a los términos de la expresión para indicar su importancia. En la figura 2.11 se muestra un ejemplo de la ponderacion de terminos en una consulta.

Generalmente el usuario de los lenguajes orientados a las proposiciones es un intermediario entre quien solicita la información y el sistema. Se requiere de cierto entrenamiento para emplear las facilidades que proporciona el sistema. Las relaciones y conexiones que conforman la base de datos y la asignacion implantada entre estos, tendrán que entenderse para obtener un resultado efectivo.

Empleado	Departamento	Gerente	Salario	Conyuge	Edad	Antiguedad
P.Salazar	P.Sistemas	J.Cepeda	P.TOTAL; > 300,000	0	P. 25	P.3

En este ejemplo no se considera el conyuge, por lo que puede eliminarse tanto el encabezado como el campo. Es posible establecer valores constantes en la consulta; por ejemplo J.Cepeda y >300,000. El subrayado indica que los valores son muestras y el prefijo P indica que la columna se va a imprimir. El doble subrayado indica agrupamiento para la generacion de totales

Figura 2.8 Consulta mediante ejemplo

```
LIST EMPLEADO,SUELDO,EDAD,ANTIGUEDAD,CARGO FOR SALARIO > 80000
.AND. GERENTE = "S.CASTRO" .AND. DEPARTAMENTO = "SISTEMAS"
```

a)

Verificacion de sueldos

Aldana Ricardo	82,000.00	35	4	programador
Carrasco Luz Maria	95,000.00	29	1	analista de sistemas
Fernandez Rafael	123,000.00	24	1	consultor
Jimenez Silvia	80,000.00	26	5	secretaria
Mendez Carmen	90,000.00	23	1	Asesor
Narvaez Pedro	133,000.00	27	6	capturista
Pallares Hector	162,000.00	32	7	consultor
Sanchez Jimena	80,000.00	35	8	capturista

b)

Figura 2.9 a) Planteamiento de una consulta interactiva
b) Resultados de la consulta

Dale 21/09/85

El dia 19 de Septiembre de 1985 a las 7:00 am se registro un sismo en la ciudad de Mexico, causando grandes estragos en el centro de la ciudad; asi como en las delegaciones Benito Juarez y Cuauhtemoc. La cifra de los danos aun no se ha dado a conocer.

Figura 2.10 Consulta en texto libre

A los terminos que interesan en la consulta se les asignan valores dependiendo de su importancia:

PONDERACION: Acero(5), Aluminio(3), Cobre(3), Estano(3), Metalicos(2), Ferrosos(2)

En la consulta se puede incluir la siguiente expresion condicional

PONDERACION > = 5 ?

Figura 2.11 Ejemplo de la ponderacion de terminos

Con frecuencia existen diferencias de sintaxis y palabras clave entre los lenguajes de consulta y los lenguajes semejantes de programación. Estas diferencias provocan errores y frustraciones.

En algunos lenguajes de preguntas de alto nivel, las especificaciones dadas por palabras en ingles son reemplazadas por fórmulas matemáticas.

2.3.1.4 Algebra y Calculo relacional

ALGEBRA: PROYECCION, UNION, COMPOSICION, RESTRICCION, DIVISION.

II PROYECCION: El operador proyeccion sobre una relacion A bajo ciertos dominios D's da como resultado una relacion B donde B es una relacion que tiene solo los dominios especificados y no contiene renglones o eneadas repetidas.

```
EXP = II EMP-EXP      (EXPERTO, LOCALIZACION )
EMP = II EMP-EXP      (NUM:EMP,NOM:EMP,EXPERTO,EDAD)
```

EXP

EXPERTO	LOCALIZACION
BAS-DAT	CHIHUAHUA
PRO-SIS	MERIDA
SIS-OPE	D.F.
BAS-DAT	LONDRES
SIS-OPE	LOS ANGELES

EMP

NUM:EMP	NOM:EMP	EXPERTO	EDAD
1	MARTINEZ	BAS-DAT	30
2	GOMEZ	PRO-SIS	35
3	PERALTA	SIS-OPE	42
4	JIMENEZ	PRO-SIS	34
5	ORTEGA	BAS-DAT	40
6	RAMOS	BAS-DAT	50
7	PEREZ	SIS-OPE	39

* UNION: El operador union toma dos relaciones A y B que tienen un dominio comun y las une para formar una nueva relacion C, en donde cada renglon de C estara formado por la concatenacion de los renglones de A con los de B cuando coincida el dominio especificado.

EMP + EXP = EMP * EXP (EXPERTO)

NUM:EMP	NOM:EMP	EXPERTO	EDAD	LOCALIZACION
1	MARTINEZ	BAS-DAT	30	CHIHUAHUA
1	MARTINEZ	BAS-DAT	30	LONDRES
2	GOMEZ	PRO-SIS	35	MERIDA
3	PERALTA	SIS-OPE	42	LOS ANGELES
3	PERALTA	SIS-OPE	42	LOS ANGELES
3	PERALTA	SIS-OPE	42	LOS ANGELES
3	PERALTA	SIS-OPE	42	LOS ANGELES
3	PERALTA	SIS-OPE	42	LOS ANGELES

X. COMPOSICION: La composicion de dos relaciones A y B que tienen un dominio comun, da como resultado una nueva relacion C, en donde cada renglon de C estara formado por la concatenacion de los renglones de A con los de B cuando coincida el dominio especificado, excluyendo el dominio comun.

EMP X EXP = II EMP + EXP (NUM: EMP, NOM:EMP, EDAD, LOCALIZACION)

RESTRICCION: El operador restriccion toma dos relaciones A y B que tienen los dominios comunes D's, da como resultado una relacion C, en donde las eneadas de C son un subconjunto de las eneadas de A con los mismos dominios, es decir, excluyendo las eneadas que bajo los dominios D's no pertenezcan a B.

EMP -LOC = II EMP - EXP (NOM:EMP, LOCALIZACION).

EMP-LOC	NOM:EMP	LOCALIZACION
	MARTINEZ	CHIHUAHUA
	GOMEZ	MERIDA
	PERALTA	D.F.
	JIMENEZ	MERIDA
	ORTEGA	CHIHUAHUA
	RAMOS	LONDRES
	PEREZ	LOS ANGELES

EXP - EMP = EMP + EXP EMP - LOC

DIVISION: La operacion division toma dos relaciones A y B y produce una nueva relacion C. La operacion se practica sobre una relacion binaria (dos dominios) y una unaria (un dominio) y produce una relacion unaria.

LOC (LOCALIZACION)
LOC LOCALIZACION
CHIHUAHUA

```
EMP - LOC / LOC =        NOM:EMP
                         MARTINEZ
                         ORTEGA

 LOC   LOCALIZACION
       CHIHUAHUA
       MERIDA

EMP - LOC /LOC   =       NOM:EMP

 LOC   LOCALIZACION
       CHIHUAHUA
       LONDRES

EXP/LOC =                EXPERTO
                         BAS-DAT
```

OPERACIONES DEL CALCULO RELACIONAL

CUANTIFICADOR DE EXISTENCIA
CUANTIFICADOR UNIVERSAL (PARA TODO)
: TAL QUE
OPERADORES LOGICOS AND, OR, NOT.
OPERADORES DE COMPARACION.

S	S#	S NOMBRE	ESTATUS	CIUDAD
U	S1	SANTOS	20	LONDRES
R	S2	RAMOS	10	PARIS
T	S3	PEREZ	30	PARIS
I	S4	GOMEZ	20	LONDRES
D	S5	ORTEGA	30	ATENAS
O				
R				

P	P#	P NOMBRE	COLOR	PESO
A				
R	P1	TUERCA	ROJO	12
T	P2	RONDANA	VERDE	17
E	P3	TORNILLO	AZUL	17
	P4	TORNILLO	ROJO	14
	P5	GRAPA	AZUL	12
	P6	CLAVO	ROJO	19

SP

S#	P#	CTD
S1	P1	300
SI	P2	200
SI	P3	400
SI	P4	200
SI	P5	100
SI	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

```
* Obtener los numeros de las partes que son surtidas
  GET W (SP. P#)

* Obtener toda la relacion S
  GET W (S)

* Obtener los numeros de surtidor los cuales esten en "Paris" con
un estatus > 20.
  GET W (S.S#) : S. CIUDAD = "PARIS" ^ S. ESTATUS > 20

* Obtener los numeros de proveedor y estatus para los que esten
en "Paris".
 GET W (S.SH.S ESTATUS): S. CIUDAD = "PARIS".

* Obtener los numeros de proveedor para los que surtan la parte
2.
GET W (SP.S#  ) : SP.  P# = "2".

* Obtener los nombres de los proveedores que surten la parte P2.
GET W (S. S NOMBRE) : SP (SP.P# = "P2"   SP. S# = S. S#)

* Obtener los nombres de los proveedores que surten al menos una
parte de color rojo.
GET W (S. S NOMBRE) :   SP  (SP. S# = S. S#         P   (
   P.P#  = SP. P#     P.COLOR = "ROJO")   )
GETR W1 (SP.S#):     P  (P.P # = SP.  P#     P.COLOR = "ROJO")
GET W (S NOMBRE) :   W1. S# = S.S#)

* Obtener los nombres de los proveedores que no surten la parte
P1:
GET W (S. S. NOMBRE) :  SP (SP. S# = S.S#   SP.P#       "P1")
```

2.3.1.5 Procesamiento en lenguaje natural

Los documentos y los lenguajes de consulta considerados en recuperacion de informacion, estan disponibles tambien en forma de lenguaje natural. Es importante tener cuidado de los metodos automaticos actuales usados para procesar textos. En este capitulo se describiran algunos avances para el procesamiento automatico de materiales y documentos escritos en lenguaje natural haciendo enfasis en aplicaciones de recuperacion de informacion.

Se examinan varios niveles de metodos linguisticos y el papel que juegan en la recuperacion de informacion. Asi mismo se estudian sistemas modernos que entienden lenguas. Los componentes de un sistema para procesamiento de lenguaje haciendo enfasis en los procesos sintacticos que son de mucha importancia en los sistemas recuperadores de informacion.

1.- Componentes de un Sistema de lenguaje natural.

A.- Interes en el procesamiento de lenguaje natural.

Gran parte de la informacion almacenada en sistemas recuperadores de informacion bibliografica consiste en datos expresados en lenguaje natural, y la mayoria de los usuarios preferirian, si fuera posible, consultar estos sistemas utilizando su lenguaje propio haciendo las consultas de informacion necesarias a su manera.

El uso de busqueda de lenguaje natural puede elevar la eficiencia, asi como la efectividad de las operaciones de recuperacion haciendo posible la formulacion precisa de preguntas que reflejen adecuadamente las necesidades del usuario y simplificando la interaccion con el sistema.

Una segunda aplicacion del procesamiento en lenguaje natural es el uso de tecnicas complejas para la representacion del contenido de los documentos introducidos al sistema. De hecho cuando el sistema de analisis se dedica al uso de palabras individuales para la descripcion del contenido tanto de solicitudes como de documentos, una solicitud del usuario que trate con "Complejidad computacional" y que se reconozcan los terminos "Complejo" y "computo" seria facil reconocer temas extravagantes relacionados como " Calculo con numeros complejos ", como la misma area de interes. Es posible asignar frases automaticamente a los documentos y buscarlos usando estadisticas de ocurrencia y operadores de adyascencia entre palabras, pero las tecnicas establecidas son imperfectas. En particular, no pueden distinguir entre casos como "Mexico en pie" "pie en Mexico". Sin embargo, si la aproximacion de las tecnicas linguisticas fueran utilizables para combinar terminos individuales en unidades de mayor tamano, la descripcion estructurada de indices puede generarse por ejemplo, mediante la combinacion de Sustantivo - verbo - Sustantivo o por expresiones ocasionales de mayor enfoque.

Otro problema importante en recuperacion es la construccion de diccionario de sinonimos y tesauros en los cuales las palabras semejantes o relacionadas se agrupan en clases afines. Bajo las condiciones actuales los tesauros se construyen manualmente, o automaticamente utilizando la ocurrencia de palabras en los documentos introducidos al sistema. Sin embargo, si la descripcion linguistica fuera posible caracterizarla por unidades individuales de texto, las clases de los tesauros podrian definirse como el conjunto de palabras que aparecen en contextos similares en los documentos de una coleccion determinada.

Dadas las dificultades inherentes en un analisis linguistico completo de textos en lenguaje natural, muchos de estos problemas estan siendo aproximados creando situaciones simplificadas; por ejemplo, restringiendo la capacidad de area en discusion a topicos especificos del tema, o imponiendo limitaciones en la variedad de las formas linguisticas empleadas estipuladas

previamente en el sistema.

B.- Niveles de procesamiento del lenguaje.

Los diferentes niveles de lenguaje, son derivados de acuerdo a sus caracteristicas foneticas, morfologicas, lexicales, sintacticas, semanticas y pragmaticas.

Los niveles foneticos, tratan la forma de los sonidos hablados, desde como entenderlos, hasta como generarlos.

El nivel morfologico del procesamiento linguistico, concierne al tratamiento de la forma de las palabras individuales y sus partes reconocibles. El reconocimiento y la eliminacion de prefijos o sufijos de las palabras, se basan en conocimientos morfologicos.

El lexico trata con los procesamientos operantes en palabras completas. En los sistemas de informacion, este trata con la eliminacion de palabras comunes, procesamiento del diccionario de terminos, y sustituciones de palabras para la obtencion de una descripcion estructural de una oracion. Una operacion lexical separa y clasifica los terminos individuales en conjuntos predefinidos para su analisis posterior.

La sintaxis se disena de tal manera que pueda agrupar las palabras de una oracion en unidades estructurales tales como las frases preposicionales, y sujeto - verbo objeto que represente colectivamente la estructura gramatical de una oracion.

El nivel semantico agrega el conocimiento de contexto al procesamiento sintactico de manera que reestructure el texto en unidades que representen el significado real de la informacion.

Finalmente, el nivel pragmatico usa informacion adicional sobre el medio ambiente social en que existe, el documento, sobre la relacion que prevalece entre los objetos, y el mundo, para ayudar a la interpretacion del texto.

Cuando se analizan los diferentes procesamientos linguisticos, debe considerarse el grado en que se reconoce cada uno de estos niveles en forma independiente. Desafortunadamente, esta pregunta, como muchas otras crean controversia. Todos los analistas coinciden en que el procesamiento automatico del lenguaje es complicado, que el uso del contexto en el que aparecen palabras individuales es escencial para la interpretacion automatica, y que la sintaxis y la semantica se relacionan en que el conocimiento semantico se requiere para evitar ambiguedades en la sintaxis, asi como la informacion sintactica auxilia en la produccion de resultados de interpretacion semantica.

C: Sistemas que entienden lenguajes.

Desde 1950, en que se escribieron los primeros programas, se ha trabajado en la traduccion de un lenguaje a otro. Logrando resultados interesantes, se han construido diccionarios mecanizados utiles de la traduccion automatica de grandes textos no restringidos, pero en la actualidad en que se han probado estos diccionarios con textos largos como entrada, se descubrio que no funcionan tan bien como se esperaba, de donde se concluye que el trabajo de traduccion, no es tan facil, sino que es muy complicado.

Los intentos hechos en el analisis de textos, estuvo orientado a las palabras, como en los casos de traduccion directa. Se ha trabajado en la refinacion de estos sistemas permitiendo el reconocimiento de frases y logrando su conversion a su forma pasiva, por ejemplo "Juan golpeo la pelota" queda " la pelota fue golpeada por Juan". Este adelanto permite hacer la distincion entre las formas declarativas e interrogativas correspondientes.

Claramente se ve la necesidad de contar con la ayuda de un procesador semantico para la interpretacion del texto, por ejemplo, reconociendo las frases "Juan juega con la bola" y "Juan se hizo bolas," el termino bola" se refiere a cosas diferentes.

En la actualidad, la mayoria de los linguistas computacionales han aportado ideas para la integracion de un sistema completo parael procesamiento del lenguaje. Un sistema completo de este tipo para la recuperacion de informacion puede verse como una estructura de 3 partes:

1.- Se construye una representacion formal normalizada de las oraciones o unidades en discusion; usualmente las unidades de significado se obtienen a partir de un diccionario, y posteriormente se ensamblan en una representacion ocasional formal usando las reglas de la sintaxis.

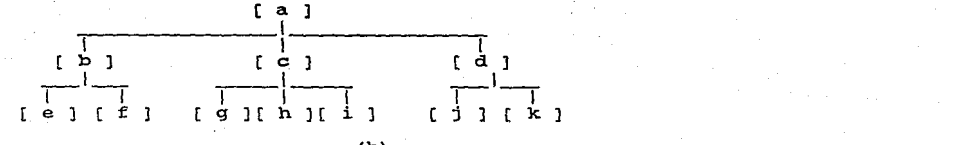
2.- Esta oracion formal, se compara contra una base de conocimientos para aumentar las descripciones iniciales e identificar las relaciones entre las unidades.

3.- Finalmente, se ejecuta la tarea deseada, la cual usa la informacion provista por la oracion de entrada, enriquecida por el conocimiento almacenado.

Uno de los problemas encontrados y resuelto parcialmente por diferentes formas es la representacion o almacenamiento del conocimiento adquirido. Para las bases de conocimiento, antes mencionadas, se supone un conjunto de reglas logicas relacionadas entre si por el mismo enviado, y en algunos casos recursivas, donde se almacenan los conocimientos primarios, esperando que con estos y sus relaciones se deduzcan casos sus alrededores y resuelvan casos de informacion indirecta. Es posible lograr una configuracion como esta en lenguajes de inteligencia artificial como LISP y PROLOG.

```
/* PROGRAMA familia */

domains
  nombre=symbol

predicates
  padre    (nombre,nombre)
  hijo     (nombre,nombre)
  hermano  (nombre,nombre)
  abuelo   (nombre,nombre)
  primo    (nombre,nombre)
  nieto    (nombre,nombre)
  sobrino  (nombre,nombre)

clauses
  padre(a,b).
  padre(a,c).
  padre(a,d).
  padre(b,e).
  padre(d,f).
  padre(c,g).
  padre(c,h).
  padre(c,i).
  padre(d,j).
  padre(d,k).

  hijo(X,Y)     :- padre  (Y,X).
  hermano(X,Y)  :- padre  (Z,X), padre   (Z,Y), X<>Y.
  tio(X,Y)      :- padre  (Z,Y), hermano (X,Z).
  abuelo(X,Y)   :- padre  (X,Z), padre   (Z,Y).
  primo(X,Y)    :- padre  (Z,X), padre   (W,Y), hermano (Z,W).
  nieto(X,Y)    :- abuelo (Y,X).
  sobrino(X,Y)  :- tio    (Y,X).
```

(a)



(b)

Figura 2.12: (a) Programa familia (PROLOG). (b) Arbol genealogico

```
primo   (j,X)                       Quienes son los primos de j ?
        X=e  -|
        X=f   |
        X=g   |-- 5 Soluciones
        X=h   |
        X=i  -|

tio     (c,X)                       De quien es tio c ?
        X=e  -|
        X=f   |__ 4 Soluciones
        X=j   |
        X=k  -|

abuelo  (X,f)                       Quien es abuelo de f ?
        X=a  -|-- 1 Solucion

nieto   (X,a)                       Quienes son nietos de a ?
        X=e  -|
        X=f   |
        X=g   |
        X=h   |-- 7 Soluciones
        X=i   |
        X=j   |
        X=k  -|

nieto   (c,a)                       c es nieto de a ?
        Falso

abuelo  (a,h)                       a es abuelo de h ?
        Verdadero

primo   (a,X)                       De quien es primo a ?
             -|-- No hay soluciones

hermano (g,X)                       De quien es hermano g ?
        X=h  -|__ 2 Soluciones
        X=i  -|

sobrino (f,X)                       De quien es sobrino f ?
        X=c  -|__ 2 Soluciones
        X=d  -|

hijo    (X,b)                       quien es hijo de b ?
        X=e  -|__ 2 Soluciones
        X=f  -|
```

Figura 2.13 Ejecucion del programa familia.

Mucha gente, argumenta por otro lado que la representacion del conocimiento en el cerebro o en los seres humanos se logra mediante las descripciones redundantes que representan a cada objeto y que la descripcion particular usada bajo cierta instancia se basa en las circunstancias y medio ambiente.

Se dice tambien que la representacion del conocimiento se basa en las diferencias que tiene un objeto, de tal manera que lo distinga de cualquier otro en cualquier situacion.

Un sistema automatico que entiende lenguajes, NO sera practico hasta que se encuentra una solucion practica para la construccion y manipulacion de las estructuras necesarias para el almacenamiento del conocimiento.

2.-Procesamiento del lenguaje y recuperacion de informacion.

Hay individuos convencidos de que para recuperar elementos "sobre" cierta area de interes, es necesario usar todos los hechos disponibles. Esta operacion requiere del analisis de significado de la informacion. En particular un buen indexado, o analisis de contenido, consiste en la traduccion del documento la solicitud en un lenguaje final consistente de conceptos y sus relaciones entre si.

En la actualidad se esta trabajando bastante en el desarrollo y uso de herramientas semanticas para analizar estructuras linguisticas -Esta en aumento la inclusion de marcadores semanticos como entradas a los diccionarios y tesauros, los indicadores de relacion especificados en las redes o arboles semanticos, son utiles para formar frases y para deshacer la ambiguedad de algunos terminos. Aun en caso de no existir las redes semanticas con sus respectivas relaciones, se utilizan algunas herramientas sintacticas, donde a las palabras reconocidas se les agregan connotaciones semanticas.

El punto de vista opuesto a la importancia del analisis del lenguaje en recuperacion lleva a concluciones diferentes. En particular se puede senalar la importancia y utilidad de la estadistica, probabilidad o tecnica vectorial para la realizacion de indices y la dosificacion. Por otro lado, se comprueba que los metodos linguisticos son efectivos en la recuperacion. La razon de esto, puede ser la existencia de diferencias fundamentales entre la recuperacion de informacion y otras tareas del procesamiento de lenguaje. En la recuperacion se necesita interpretar un documento recuperable, en vez de convertir la solicitud a una expresion con significado exacto. Es por eso que dos documentos que traten de lo mismo dentro de la misma area de interes, pero con diferentes conclusiones, son tratadas igualmente en la recuperacion, esto es tanto en la busqueda como en la extraccion. En un sistema de pregunta-respuesta, o de traduccion del lenguaje, estos documentos serian tratados en forma totalmente diferente.

Este punto de vista acerca de los sistemas recuperadores de informacion, revela la importancia del analisis del lenguaje, rechazando la idea de que la recuperacion de informacion sea una tarea facil, mas aun para los sistemas de pregunta-respuesta. Estos ultimos, requieren entender el area, con objeto de permitir la generacion de inferencias que llevan a respuestas especificas.

3.- Sistemas para el analisis sintactico.

Existen especialmente tres sistemas de importancia para el analisis de la sintaxis.

-La gramatica estructurada en frases, que permite el modelado de las propiedades estructurales basicas de los elementos linguisticos.

-La gramatica de transformacion, que contabiliza las diferentes representaciones sintacticas de fragmentos semanticos equivalentes.

-La gramatica de redes de transicion, que es la mas comunmente usada en los sistemas automaticos de procesamiento del lenguaje.

4.-Gramatica estructurada en frases.

Muchos creen que el hacer solamente un analisis sintactico del lenguaje no hace mucho por aquellos que estan interesados en el procesamiento del lenguaje. Los linguistas destacan por su trabajo en la gramatica "generativa". Esta es disenada para generar oraciones correctas gramaticalmente. Una de las propiedades mas importantes de la gramatica, es la sencillez, en el sentido de que una gramatica pequena debe prever el espacio para la generacion o el analisis de un gran numero de oraciones.

La gramatica conocida como "estructurada en frases", es util y simple para las tareas de analisis y generacion de oraciones.

Considerando reglas de la forma: S --- A + B , debe entenderse que la variable "S" se deriva como "A" seguida por "B". El simbolo "+" es un simple separador de variables. Cuando se presentan reglas de escritura, las letras mayusculas denotan elementos no-terminales, es decir, elementos que pueden sustituirse por lo que aparezca a la izquierda de alguna regla identificada igualmente. De la misma manera las letras minusculas denotan elementos terminales que no pueden ser sustituidos. El simbolo "S", denota el inicio de una expresion linguistica.

```
S --- NP + VP
NP --- T + N
T --- el
N --- nino / senor
VP --- juega / trabaja
```

1.- El nino juega
2.- El nino trabaja
3.- El senor juega
4.- El senor trabaja

Son posibles estas cuatro oraciones escribiendolas cada una como regla gramatical de la forma.

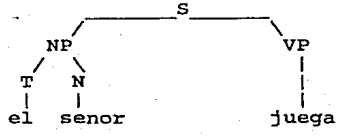
S --- oracion

Suponiendo que se le agregara una palabra a esta gramatica, solamente se requiere incorporar una regla para permitir la generacion de oraciones.
Ejemplo:

N --- perro

5.- El perro juega
6.- El perro trabaja
La incorporacion de una regla gramatical, permite evitar la escritura o definicion de reglas para cada oracion posible generada apartir de la palabra.

Es posible hacer una representacion grafica para cada oracion especifica, respetando la gramatica dada.



El arbol generado apartir de esta representacion, contiene informacion adicional sobre la estructura de la oracion, conociendo las frases que la constituyen; por esta razon, a esta representacion se le conoce como "arbol de estructura en frases" o "marcador de frases"; la gramatica descrita es una "gramatica estructurada en frases". Otra representacion del arbol puede ser : ((El nino juega) juega) que corresponde a la misma oracion.

Alas gramaticas como la descrita anteriormente se le conoce como "contexto libre" porque los simbolos no terminales de la izquierda de la regla pueden sustituirse por los de la derecha; no obstante, del contexto en que aparezcan; es decir, no se aplican restricciones de contexto a ninguna regla.

Las gramaticas de contexto-libre tienen problemas en la utilizacion de constituyentes discontinuos, ademas de que no reflejan la complejidad del lenguaje.

B.- Gramaticas de transformacion.

La diferencia basica que presenta es la introduccion de reglas sensibles al contexto del tipo:

w A x ---> w @ x

donde "A" es una variable no terminal de la gramatica y "@" es una cadena de caracteres terminales o no terminales.

La regla especifica que cuando aparece la "A" entre el contexto de "W" y "x", entonces "A" puede reemplazarse por la cadena "@".

1.- Juan probo el teorema
2.- el teorema fue probado por Juan
3.- Juan no probo el teorema
4.- acaso Juan probo el teorema ?
5.- fue el teorema probado por Juan ?
6.- el teorema no fue probado por Juan
7.- acaso no fue Juan el que probo el teorema ?
8.- no fue el teorema probado por Juan ?

Estas oraciones fueron generadas a partir de la primera de ellas a traves de una secuencia de transformaciones, particularmente empleando los modos activo-pasivo, positivo-negativo y declarativo-interrogativo, como se muestra en la siguiente tabla:

ACTIVO	PASIVO	DECLARATIVO
S	S	S
N	S	S
S	N	S
S	S	N
N	S	N
N	N	S
S	N	N
N	N	N

Suponiendo que la oracion inicial pudo generarse por una regla como

S --> NP1 + V + NP2

donde

NP1 <-- Juan,
NP2 <-- el teorema y
V <-- probo|probado

donde NP1 y NP2 denotan instancias especificas de una frase para un sustantivo particular, es facil notar las reglas de transformacion sensibles al contexto que generaron las oraciones transformadas:

ACTIVO-PASIVO:
NP1 + V + NP2 ---> NP2 + fue + V + por + NP1

POSITIVO-NEGATIVO:
NP1 + V + NP2 ---> NP1 + no + V + NP2

DECLARATIVO-INTERROGATIVO:
NP1 + V + NP2 ---> acaso + NP1 + V + NP2

El analisis del lenguaje, o el proceso de reconocimiento, usando una gramatica de transformacion es mas complejo. Este tipo de gramaticas puede separarse en dos partes.

- Componente base: que genera la llamada estructura profunda de una oracion reflejando la interpretacion que hace sobre la sintaxis y la semantica.

- Componente de transformacion: que opera con la salida del componente base y genera la estructura superficial de la oracion reflejando la representacion fonetica.

Para el analisis del lenguaje natural, es necesario seguir los pasos descritos a continuacion

1. Un sistema comun de "parsing", usado para obtener uno o mas arboles, que exhiban la estructura superficial de la entrada.

2. Aplicar transformaciones inversas a las estructuras superficiales para obtener la estructura profunda.

Este proceso se repite tantas veces como haya arboles que se generen a partir de la aplicacion de las transformaciones inversas existentes.

La experiencia indica que el proceso inverso, no se cumple excepto cuando el numero de arboles superficiales generados es muy pequeno o cuando se alimento informacion suficiente para cada nodo en los arboles de superficie para seleccionar solo aquellas transformaciones inversas que se ajusten correctamente a las estructuras profundas.

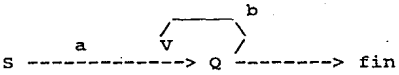
REDES GRAMATICALES DE TRANSICION

Este tipo de gramatica ofrece todas las facilidades inherentes de las gramaticas de tranformacion.

Ademas su estructura es suficientemente simple para permitir su utilizacion en la practica; Como resultado de ello, actualmente estas gramaticas son las mas usadas en los sistemas de procesamiento del lenguaje.

Los sistemas que utilizan este tipo de gramaticas, emplean la facilidad de las maquinas de estados finitos representadas como una grafica. Cada nodo de la grafica representa un estado de la maquina, y las ramas son las transiciones de un estado a otro.

Considerando como ejemplo, la siguiente grafica



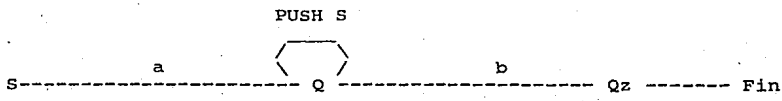
El comienzo es en el estado S, si el simbolo de entrada es una "a", entonces se hace una transicion al estado Q., si no fue una "a", el proceso de reconocimiento falta debido a que la grafica no considera esta posibilidad, esto es, no hay una trayectoria que permita la salida de S para estos casos. Cuando el simbolo de entrada cumple con la condicion de ser "a", entonces, se generan dos nuevas posibilidades en el estado Q., cuando ocurre una entrada "b", para lo cual la maquina regresa a Q., aceptando una cadena del simbolo "b", o el proceso de reconocimiento termina cuando al estado Q, no le llega una "b". La grafica del ejemplo permite la generacion de oraciones de la forma

$$a^n b^n \quad \text{para } n > 0$$

abbb...b o a b para n > 0

Una grafica tan simple como la anterior no es capaz de aceptar lenguajes de contexto libre. Seria deseable que en vez de eso se tuviera una coleccion de graficas donde fuera posible pasar de una a otra con el simple hecho de llamar ciertas ramas con el nombre del estado inicial de otra.

Un ejemplo de estas graficas recursivas es la siguiente:



En este ejemplo se observa que una vez alcanzado el estado Q, el resultado actual puede almacenarse y debe regresarse al estado inicial S nuevamente. Es comun utilizar una estructura del tipo stack para el almacenamiento de las condiciones iniciales al comenzar nuevamente un estado (recursividada).

En el ejemplo, la rama en el estado Q, se llama PUSH S para indicar que el resultado en este momento se almacena en el stack y antes de regresas al estado S, al menos de que se reconozca una "b" en la entrada, en cuyo caso se trasladara al estado Qz.

Cuando se llega al simbolo "fin" el stack esta vacio, el resultado acarreado es el correcto, pero si el stack no esta vacio, debe lerse el ultimo simbolo insertado, decrementar el stack y regresar al estado donde se origino el almacenamiento de ese simbolo.

Puede verificarse que esta ultima grafica es capaz de aceptar oraciones del tipo

a^n b^n:--> : es decir una cadena de "a" s con el mismo numero de "b" s.

Estado actual	Cadena para reconocer	Contenido del Stack
S	aabb	-
Q	abb	-
S	abb	Q1
Q1	bb	Q1
Q2	b	Q1
Fin (POP)	b	-
Q1	b	-
Q2	-	-
Fin (POP)	Aceptado	

Reconocimiento de la cadena "aabb" por una grafica de estados finitos.

En las redes gramaticales practicas, puede existir ambiguedad en los nodos; es decir, que un estado dado permita muchas transiciones posibles a otros estados. Seria logico pensar en arreglar y buscar las trayectorias posibles de acuerdo a la probabilidad determinada para cada caso, de manera que se optimice la busqueda de la trayectoria para el simbolo de la entrada. Esto se presenta de manera importante en las redes recursivas donde se analizan cadenas largas de caracteres, y por alguna circunstancia ocasionada por un simbolo en el proceso del reconocimiento es necesario regresar hasta el punto donde existe ambiguedad, y se tomo una decision o trayectoria gramaticalmente valida.

Las operaciones de construccion estructural pueden ser muy complejas. Por ejemplo, para reacomodar un arbol de estructura de frase siguiendo el analisis de una oracion pasiva. Se han sugerido sistemas alternativos para el analisis gramatical, donde el barrido de simbolos dentro de una oracion se hace de derecha a izquierda en vez de izquierda a derecha, analisis simultaneo del crecimiento de muchas trayectorias en vez de utilizar un rastreo hacia atras cuando se detecta ambiguedad. Mucha gente opina favorablemente sobre las redes gramaticales de transicion como el mejor medio para el reconocimiento sintactico.

4 ANALISIS SINTACTICO EN RECUPERACION DE INFORMACION

Los metodos de analisis sintactico pueden utilizarse principalmente de dos formas para la recuperacion de informacion bibliografica:

1. La identificacion sintactica puede mejorar la operacion de indexado haciendo posible la asignacion de documentos y solicitudes de frases sintacticamente correctas reemplazando los terminos individuales usualmente empleados.

2. Es posible obtener una vision detallada sobre el contenido del documento, usando aproximaciones sintacticas, tratando las autoridades de descripcion directa sobre porciones de oraciones y/o parrafos del documento.

Este segundo metodo es comunmente llamado recuperacion de pasajes, donde se requiere de un analisis del texto completo para la extraccion del contenido. Este metodo proporciona mayor informacion dado que responde las preguntas directamente en vez de mencionar las referencias solamente.

El uso mas inmediato de las tecnicas sintacticas en recuperacion de informacion recae sobre tareas de indexado y particularmente la eleccion de frases de sustantivos o frases de preposiciones para propositos de indexado. Una posibilidad es usar un analisis sintactico simplificado que asigna una o mas marcas sintacticas a cada palabra del texto y definiendo una frase de indexado consistente de un conjunto de palabras contiguas que representan una secuencia especifica de marcas sintacticas.

Una tecnica particular para la deteccion de frases de indexado utilizan el siguiente proceso para indexar.

1. Se usa un diccionario de reconocimiento para asignar una de 16 posibles categorias sintacticas a cada palabra.

2. Un diccionario de formato que almacena un total de 77 formatos sintacticos permitidos en la construccion de frases.

3. El ciclo de indexado consiste en acumular secuencias de terminos indispensables de hasta cinco palabras de la entrada, la longitud de la secuencia se determina por la ocurrencia de delimitadores, como signos de privacion.

Las marcas sintacticas correspondientes a las palabras de la entrada, se obtienen del diccionario de reconocimiento, mientras que el diccionario de formato se usa para determinar si la secuencia de marcas de entrada corresponde a uno de los formatos permisibles en el diccionario, en cuyo caso, se acepta la frase correspondiente como una frase de indexado. En caso contrario, se repite el proceso eliminando una palabra de la secuencia.

Un metodo alternativo para generar frases de indexacion, consiste en realizar un analisis sintactico completo del texto, o extractos del texto como en un resumen, y en asociar aquellas frases como terminos de indexado cuyos componentes exhiban relaciones sintacticas especificas entre ellos.

La conclusion es que el papel de los metodos linguisticos en general y el analisis sintactico en particular, no se ha definido para los sistemas recuperadores de informacion.

2.3.2 Estrategias de procesamiento

Todos los lenguajes de consulta a bases de datos cuentan con las siguientes operaciones:

1) Comandos para modificar archivos, tales como insertar, borrar y modificar.

2) Capacidades aritmèticas como sumas y restas.

3) Comandos de asignaciòn e impresiòn.

4) Operaciones que permitan obtener valores globales de los registros de un archivo, incluyendo los valores de suma, promedio, total, màximo y mínimo.

Los procesos de bùsqueda y actualizaciòn de archivos pequeños cuyo contenido permanece constante durante largos perìodos de tiempo no presentan problema, y pueden hacerse mediante la exploraciòn secuencial de todo el archivo. Pero desafortunadamente en la realidad los archivos no son pequeños ni estàticos; por lo que no es viable que los procesos de bùsqueda y actualizaciòn se hagan mediante la exploraciòn secuencial, ya que esto resultaría muy lento y costoso. Para ello se han creado varias tècnicas:

1) Manejo de índices que permiten modificar porciones del archivo sin tocar el resto.

2) Un índice invertido se utiliza para identificar todos los registros que cumplan con valores particulares de ciertos atributos. Los registros resultantes se pueden recuperar rápidamente si estàn agrupados; es decir, si se encuentran contiguos físicamente.

Cuando el nùmero de tèrminos que identifican un registro, o el nùmero de grupos utilizados para particionar la coleccìn crecen, entonces el índice se puede partir en una jerarquía de índices. De manera que el índice de mayor nivel sirve para proveer el acceso a otros índices màs detallados, que a su vez permiten el acceso a otros niveles y asi sucesivamente hasta que los registros almacenados en el archivo principal sean identificados. Este mètodo es el que se utiliza en la busqueda sobre àrboles agrupados (cluster tree search).

Tanto la jerarquía de índices como los registros del archivo principal deben buscarse y mantenerse en forma eficiente bajo las operaciones de anadir, borrar y modificar su contenido.

Cuando los registros se conectan mediante apuntadores, las operaciones de anadir y borrar pueden hacerse fácilmente mediante un cambio en los apuntadores tal y como se muestra en la figura 2.14 a y b.

Entre más cambios se hagan resulta indispensable reorganizar el área de almacenamiento, para compactar el espacio, almacenando la información en forma contigua y eliminando los espacios vacíos dejados por los archivos borrados.

Las reorganizaciones de archivo pueden involucrar alteraciones en el nivel conceptual que requieren cambios en el esquema del archivo. Como en el caso de alteraciones en arreglos jerárquicos o de red, para lo cual se pueden hacer cambios físicos al crear un nuevo nivel de indexación, o cambiar el método de acceso para recuperar registros. Para dicha reorganización debe copiarse la base de datos a un almacenamiento auxiliar y hacer los cambios necesarios fuera de línea, recargando la version correcta en el almacenamiento principal, siguiendo la operación de actualización; o bien pueden hacerse las correciones en línea, bloqueando el acceso a ciertas partes de la base de datos, mientras se hacen las revisiones.

Aun cuando existan métodos que minimicen las dificultades inherentes a la actualización y al mantenimiento de la base de datos, cualquier reorganización puede requerir recursos sustanciales, sobre todo cuando la base de datos es muy grande.

También es importante considerar la eficiencia de las operaciones de búsqueda y recuperación. El problema básico en la optimización de preguntas consiste en determinar el orden en que se van a ejecutar las operaciones para recuperar la informacion como resultado de una formulacion de pregunta. El problema de optimizacion es mas importante en las bases de datos relacionales, debido a que no siempre existen en el sistema trayectorias de acceso fácil provistas por estructuras de apuntadores entre registros.
El provechos de construir un índice sobre los atributos de unión u ordenar las relaciones antes de la unión depende del tamano de las relaciones, de la dificultad de las operaciones de indexación u ordenamiento, del tamano de la memoria interna utilizada por el sistema en estudio y de la frecuencia con la cual se debe ejecutar una operación de unión.

Las siguientes reglas generales de optimizacion son útiles bajo la mayoría de las condiciones operacionales:

1) Ejecutar todas las operaciones de selección tan pronto como sea posible, debido a que reduce el tamano de los archivos.

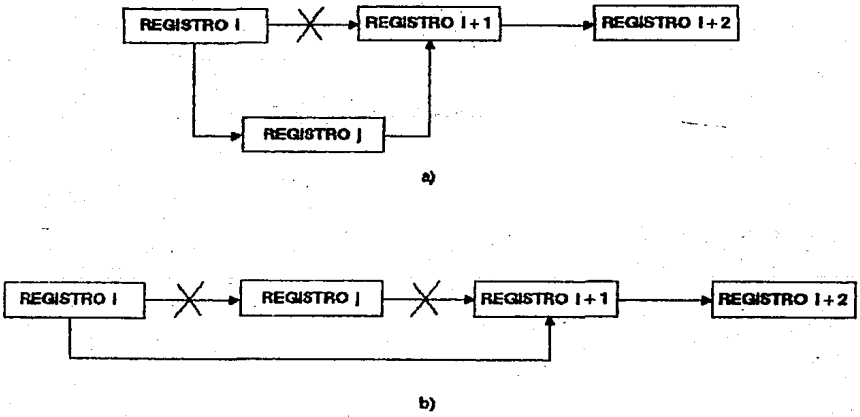


Figura 2.14 Manejo de apuntadores para insertar y eliminar registros.
a) Insercion del registro J entre los registros I e I+1.
b) Eliminacion del registro J.

2) Preprocesar los archivos antes de hacer una unión, creando un índice sobre los atributos de unión u ordenando donde se indique.

3) Ensamblar secuencias de selecciones o secuencias de operaciones de proyección en una selección o proyección simples; una secuencia de estas operaciones puede ejecutarse en una exploración simple del archivo.

4) Combinar las operaciones de proyección con otras operaciones binarias, involucrando varias relaciones para reducir el número de atributos que necesitan procesarse.

5) Combinar operaciones de selección con un producto cartesiano previo para generar una unión. Generalmente una unión natural es menos costosa de ejecutar que un producto cartesiano.

Actualmente se han optimizado las funciones de procesamiento de preguntas en algunos sistemas de base de datos relacionales.

En muchas circunstancias, los sistemas jerárquicos y reticulares que proveen acceso directo para la mayoría de las combinaciones de registro importantes, pueden sin embrago ser preferidas a un sistema relacional optimizado.

2.3.3 Dinamica de los sistemas recuperadores de informacion

El diseno de recuperadores de información empleando Bases de datos debe hacer coincidir tres componentes que son:

Metodología de consulta, interfases del usuario y la organización de la base de datos.
Algunas posibilidades de elección de los componentes de una base de datos están íntimamente relacionadas con el contenido de los datos y con las herramientas para manejarlos.
La elección de la interfase puede basarse tanto en la adecuación técnica como en que sea apropiada para la relación usuario-sistema. El tipo de sistema de bases de datos va a depender de la forma en que esten estructurados los datos con respecto a la interfase y del desempeno de los diferentes tipos de uso. En la tabla 2.A se presenta un resumen de los tres componentes básicos que deben considerarse para el diseno de un sistema recuperador de información.

Metodologia de resultado	Interfases del usuario	Organizacion de la base de datos
	Interrogativa	
Recuperacion de hechos	Interactiva	Relacional
Inferencia estadistica	Tabular	Jerarquica
Inferencia deductiva	De procedimiento	Reticular
	Lenguaje controlado	
	Lenguaje natural	

Tabla 2.A (10-1)

DINAMICA DEL USUARIO

Puede resultar imposible construir sistemas únicos que satisfagan toda la gama de necesidades. Sin embargo, es conveniente especificar a nivel conceptual que elecciones se tomaron en el diseño; así como en la implementación del sistema recuperador de información.

DINAMICA DEL SISTEMA.

Generalmente cuando un usuario hace una consulta sobre información especifica lo hace una sola vez, a menos que se le hayan olvidado o perdido los resultados de la consulta; por lo tanto, un sistema recuperador de información continuará siendo valioso siempre que exista una actualizacion en el contenido de los datos, en la capacidad de análisis de los mismos.

Para que un sistema recuperador de información funcione adecuadamente es necesario vigilar o adecuar la salida para que proporcione información útil. Además debe establecerse un mecanismo que mejore tanto la organización de los datos como del contenido.
El uso de la base de datos puede cuantificarse con estadísticas y en base a ellas efectuar cambios sustanciales en el manejo de la información. Sin embargo, es importante considerar que es más fácil que adaptarse al sistema, que adaptar el sistema a las conveniencias de todos los usuarios. Por lo que no es conveniente confiar exclusivamente en las estadísticas de uso.

En la práctica, el sistema de bases de datos se juzgará tanto por su calidad intrinseca como por los resultados que pueda proporcionar.

A los usuarios les resulta frustrante tanto la existencia de datos inadecuados y erróneos, como los sistemas de comunicación inadecuados y poco confiables que están conectados a un buen sistema recuperador de informacion.

2.4 Sistemas comerciales basados en archivos invertidos

En la actualidad, la mayor parte de los sistemas comerciales, estan basados en archivos invertidos. A continuacion se describen algunos sistemas comerciales para la recuperacion y administracion de informacion agrupados de acuerdo a sus requerimientos y caracteristicas.

========== MAIN FRAMES ==========

1.- Sistema DIALOG

Es un producto de Lockhead en California. Cuenta con aproximadamente 300 bases de datos disponibles en linea.
El sistema DIALOG se basa en una estructura de archivos invertidos. El sistema crea conjuntos con los numeros de referencia de documentos que se forman a partir de una solicitud de informacion, que consta de los terminos a buscar, intercalados los operadores AND, OR, NOT, (F), (W), etc. que a su vez pueden priorizarse mediante el uso de parentesis. El ooperador (W) es de adyascencia entre los terminos, el (F) para recuperar terminos dentro del mismo campo.
Cada solicitud de informacion genera un conjunto nuevo que adopta como nombre el numero de paso en que fue solicitada la informacion, lo que permite afinar la estrategia de busqueda, combinando terminos nuevos con los resultados retenidos en el conjunto de algun paso anterior. La secuencia de solicitudes de informacion, puede ser revisada y utilizada nuevamente. En el momento en que se desee, se pueden desplegar los documentos a que se refiere en el conjunto resultante de cualquiera de los pasos de la secuencia o historia de la consulta. Para lo cual existen varios formatos de impresion. Dentro de los comandos de este sistema, se encuentra uno muy utilizado en la actualidad y es el de truncacion o sustitucion de caracteres de un termino; su exito se debe a la facilidad de captacion o recuperacion de documentos utilizando las raices de las palabras (en algunos casos esto permite la recuperacion en varios idiomas).

EJEMPLO DOCUMENTADO DE DIALOG.

```
?ss cable (W) (tv or television?)
    S1     5979     CABLE
    S2     6747     TV
    S3    21879     TELEVISION?
    S4     4346     CABLE(W) (TV OR TELEVISION?)
```

```
?ss franchis? or licens?
     S5     1540     FRANCHIS?
     S6     4178     LICENS?
     S7     5664     FRANCHIS? OR LICENS?

?ss s4 and s7
           4346     S4
           5664     S7
            278     S4 AND S7

?ss s8 and py=1985
            278     S8
     S9  118659     PY=1985
    S10      20     S8 AND PY=1985

?t 10/8/1-3

10/8/1
1268411     DATABASE: TI File 148 *Use Format 9 for FULL TEXT*
  Columbia Pictures Industries Inc. enters into new licensing
agreement with Home Box Office Inc.
DESCRIPTORS: Columbia Pictures Industries Inc.- contracts and
specifications; Home Box Office Inc.-contracts and
specifications; cable television-contracts and specifications;
moving-picture industry-contracts and specifications

10/8/2
1268404     DATABASE: TI File 148 *Use Format 9 for FULL TEXT*
  Time Inc. approve new film licensing agreement between its
Home Box Office Inc. subsidiary and Columbia Pictures Industries
Inc.
DESCRIPTORS: Home Box Office Inc.-contracts and specifications;
Columbia Pictures Industries Inc.-contracts and specifications;
moving-picture industry-contracts and specifications; cable
television-contracts ans specifications; Tri-Star Pictures Inc.-
contracts and specifications


10/8/3
1262254     DATABASE: MI File 47
  Cal. court ruling splits franchising process open. (cable TV)
DESCRIPTORS: cable television-cases; minicipal franchises-cases;
suscription television-cases; Preferred Communications-case;
Access Cable-cases
```

2.- STAIRS (Storage and Information Retrieval System).

Es un producto de IBM para almacenar y recuperar informacion. Dado que IBM no hace bases de datos, el usuario debe adquirirlas o desarrollarlas por su cuenta para ponerlas a disposicion comercial o privada. STAIRS esta formado por dos partes fundamentales:

1) Programas de utileria para la creacion y mantenimiento de bases de datos.

2) Un sistema recuperador de informacion en linea llamado AQUARIUS, el cual se compone de un sistema interactivo de recuperacion y lenguaje de consulta o QUERY.

El sistema recuperador funciona en un ambiente multiusuario que permite dialogar con el usuario, donde el dialogo trata de la busqueda y recuperacion eventual de informacion almacenada. La diferencia principal entre STAIRS y DIALOG es que STAIRS no solamente incluye funciones de procesamiento de texto y recuperacion de documentos, sino que ademas cuenta con un administrador de bases de datos para procesamiento numerico en forma tabular. La recuperacion de registros se basa en los valores de los atributos particulares de cada registro. STAIRS utiliza modos de operacion separados para el manejo de texto y datos estructurados conocidos como SEARCH y SELECT respectivamente. Es necesario un sistema de recuperacion de texto para la creacion de un archivo invertido, un indice de texto y uno o mas archivos de texto. La figura 2.15 muestra la estructura basica del sistema STAIRS.

El archivo de texto contiene los documentos almacenados bajo un formato especial en el cual son presentados al usuario. El indice de texto incluye apuntadores a los registros del archivo de texto, asi como a informacion privada o datos estructurados asociados con subconjuntos del registro en el archivo de texto. Uno de los archivos que genera STAIRS es un diccionario de terminos, conceptualmente similar al archivo invertido, donde se puede localizar una palabra y siguiendo los apuntadores respectivos, los sinonimos y apuntadores al archivo maestro. Ademas, cada apuntador al archivo maestro, especifica la localizacion de la palabra dentro del documento a traves de un codigo del parrafo, numero de oracion y numero de palabra.
El sistema STAIRS permite la recuperacion en texto libre o en resumenes de documentos, cuenta con los operadores ADJ, WITH, SAME, truncacion y permite la utilizacion de parentesis; asi como los operadores comunes AND, OR, NOT y XOR.
En el modo SELECT para informacion tabular, permite la utilizacion de operadores de relacion, igual, mayor, menor o sus combinaciones. Tambien cuenta con un comando para recuperar la informacion contenida en algun rango, comando RANK, este rango se refiere a los documentos recuperados en el modo SEARCH, su operacion consiste en otorgar un valor a cada termino en funcion de la frecuencia de utilizacion, frecuencia del termino en el documento, numero de documentos recuperados, etc., un algoritmo particular para evaluar terminos es el siguiente:

$$\text{termino} = \frac{(\text{frecuencia de ocurrencia en el documento})\,(\text{frecuencia del termino en el conjunto recuperado})}{\#\text{ de documentos recuperados que contienen el termino}}$$

STAIRS es un sistema muy poderoso que fue disenado para el manejo de informacion tanto de texto libre como de informacion

estructurada en campos espeacificos. La desventaja de este sistema tan completo, es que requiere de grandes dispositivos de almacenamiento, en particular una computadora IBM de gran capacidad.

3.- BRS (Bibliographic Retrieval Service)

Es un sistema comercial que cuenta con mas de 100 bases de datos, esta basado en los principios del sistema STAIRS. Se origino en la red de comunicaciones biomedicas de la Universidad de Nueva York.
BRS opera exclusivamente como un sistema recuperador de informacion y no como administrador de bases de datos. Por lo tanto, muchos de los comandos de STAIRS fueron eliminados. Dado que no existe el modo SELECT siempre estara en modo SEARCH. Para encontrar valores numericos como fechas, etc se puede utilizar el operador LIMIT. El comando RANK tambine fue eliminado.
Incluye la utilizacion de campos para la informacion, evitando que sea texto libre, alun cuando es posible recuperar en cualquier campo del registro y mas aun limitar a la informacion a que se encuentre en un campo determinado.
La ventaja que tiene BRS sobre STAIRS es la simplicidad y eficiencia. La eliminacion de comandos de STAIRS, logro un sistema simple y especializado en la recuperacion de informacion.

```
1_: microcomputer   or   microprocessor
RESULT     8823 DOCUMENTS

2_: 1 with  (digital   adj (interface or satellite or switching))
RESULT       25 DOCUMENTS

3_: 2.ti.
RESULT        2 DOCUMENTS
-----------------------------------------------------------------
1_: ..set detail=on
SET HAS BEEN COMPLETED

2_: (mainlining or mainstreaming) with retarded
MAINLINING                5 DOCUMENTS
MAINSTREAMING          3595 DOCUMENTS
RETARDED               7169 DOCUMENTS
RESULT                   71 DOCUMENTS
-----------------------------------------------------------------
1_: birth adj order with risk
RESULT                    9 DOCUMENTS

2_: ..sort
  USING SEARCH STATEMENT 001
PLEASE ENTER PARAGRAPHS TO BE SORTED
  _: au
  SORT COMPLETED SUCCESSFULLY
BRS SEARCH MODE - ENTER QUERY
```

```
2_: ..p s au/doc=1-5
     1
AU ATKINSON-SUE. STANLEY-FIONA-J.

     2
AU BROWN-MARY-M.

     3
AU ESTEVE-RONALD-J.

     4
AU HICKS-ROBERTA-A.
   PELLEGRINI-ROBERT-J.
   EVANS-ELIZABETH-A.
-----------------------------------------------------------------
1_: housing adj (start$1 complet$4 finish$2)
RESULT                12 DOCUMENTS

2_: economic adj indicator$1
RESULT                 9 DOCUMENTS

3_: 1 and 2
RESULT                 1 DOCUMENT

4_: ..sdi
SPECIFY STATEMENT NUMBER_: 3
```

4.- MEDLARS

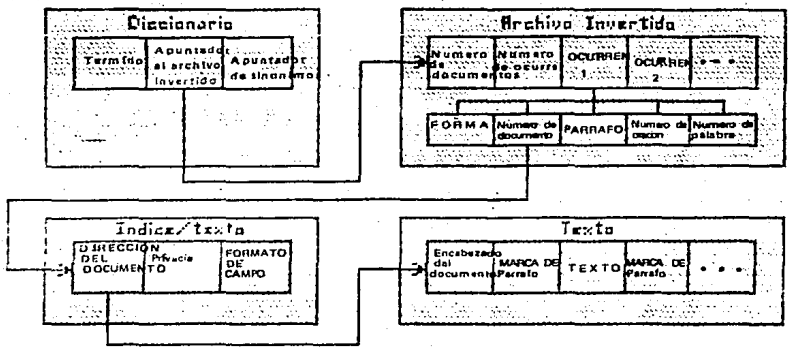
Este sistema fue disenado para la recuperacion de informacion bibliografica en el area biomedica, fue concebido en 1971 por la Biblioteca Nacional de Medicina (NLM) en Estados Unidos. La estructura de Medlars se basa en la utilizacion de archivos invertidos. Esta constituido por tres archivos diferentes: El archivo de indices, el archivo de ocurrencias y el archivo de datos. El archivo de datos almacena completamente la informacion asociada a cada registro; esto incluye toda la informacion que se le presenta al usuario. Cada registro se identifica por un numero de referencia unico, el cual es asignado por la computadora. En la figura 2.16 se puede apreciar la organizacion de archivos del sistema MEDLARS.

5.- TEXT-TRIEVE

Es un sistema recuperador de documentos. Desarrollado para Burroughs Corporation.

Se encuentra disponible para computadoras de las series B 5000, B 6000 y B 7000.

Text-Trieve permite organizar, almacenar, buscar, recuperar, desplegar, imprimir y distribuir la informacion. Ademas ofrece:



Organizacion de Archivos del sistema STAIRS

Figura 2.15

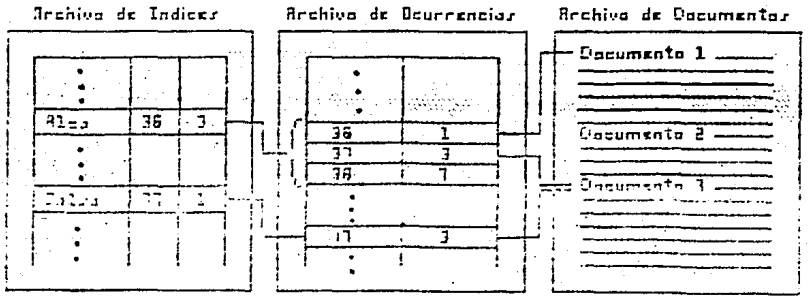


Figura 2.16

Organizacion de Archivos del sistema MEDLINE

- almacenamiento de informacion centralizado
- organizacion eficiente de informacion
- indexado por llaves
- busquedas de informacion flexibles y rapidas
- opciones de despliegue e impresion convenientes
- distribucion selectiva de informacion
- ayuda cuando se solicite
- opciones de seguridad.

El sistema Text-Trieve es rapido, flexible y facil de manejar. Permite la incorporacion de documento completo o referencia a el. Usando su modulo de Entrada de informacion, permite que un grupo autorizado de usuarios den de alta registros simultaneamente. Para las bases de datos existentes, se han implantado ciertos programas de conversion de formatos.

========== MINI COMPUTADORAS ==========

1.- FINDER
Permite la recuperacion de documentos mediante terminos ubicados en cualquier parte de las bases de datos textuales con capacidad de cientos de miles de registros con tiempos de minimos. Permite combinar conjuntos de terminos usando logica Booleana. Puede mezclarse recuperacion de texto completo con palabras llaves para afinar la busqueda.

FINDER crea archivos invertidos para el indexado de la base de datos, pudiendo contener todas las palabras si asi se desea. Cuando se busca cierta informacion, primero busca sobre los archivos de indices evitando una busqueda secuencial, el resultado logra un sistema virtualmente independiente del tamano de la base de datos y con una potencialidad de recuperacion en texto completo. Es una combinacion nunca antes utilizada en microcomputadoras de 64K de RAM.

Caracteristicas adicionales de FINDER:- permite truncacion de terminos tanto a la izquierda como a la derecha,
- permite enmascarar ciertos caracteres en palabras de interes,
- permite la utilizacion de logica Booleana para combinar conjuntos de resultados,
- contiene diferentes opciones para el despliegue de la informacion,
- permite la busqueda simultanea sobre cualquier campo del registro o restringida a un campo,
- permite que los indices de terminos contengan una o varias palabras,
- permite ordenar la informacion contenida en el conjunto resultante, de acuerdo a las especificaciones seleccionadas,
- permite la busqueda de valores en ciertos rangos,
- no requiere memorizar los comandos del sistema,
- permite elegir opcionalmente, palabras no deseadas en los indices (STOPWORDS).

========== MICRO COMPUTADORAS ==========

1.- DBASE II
Es uno de los sistemas mas populares en el mercado de sistemas administradores de bases de datos, producido por Ashton-Tate. Aun cuando dBASE II no presenta todas las caracteristicas necesarias para un ssitema de informacion bibliografica, se puede utilizar debido a la flexibilidad de sus archivos con estructura relacional.

dBASE es una herramienta muy util para el indexado de documentos ya que permite accesar la informacion por puntos multiples o campos del documento.

Unas de las principales ventajas son
- su flexibilidad para definir los archivos a utilizar,
- su facilidad para agregar registros
- cuenta con un sistema para validar informacion en forma simple,
- facil de indexar la informacion y
- cuenta con un generador de reportes.

Una caracteristica interesante es que permite capturar informacion fuera de linea y posteriormente copiar la informacion a un formato estandar para agregarla a alguna base de datos mediante un proceso BATCH en otra computadora mas grande.

Las desventajas de dBASE II como sistema recuperador de informacion son:
- El generador de reportes no es muy flexible,
- utiliza campos de longitud fija en sus registros, lo cual provoca el desperdicio de espacio en los archivos,
- No tiene comandos para el respaldo de la informacion y puede perderse accidentalmente,
- no permite la repeticion de campos,
- no cuenta con mensajes claros para la identificacion de errores.

2.- FOXBASE
Es un sistema administrador de bases de datos, que pretende mejorar ciertas deficiencias de otros programas similares, cuenta con una utileria de ayuda de excelente calidad, la cual es muy descritiva y proporciona para cada comando, ejemplos de como debe usarse.

Es un programa que acelera el tiempo de ejecucion del dBASE II en una proporcion de 5 a 22 veces, ya que cuenta con un compilador de los comandos definibles por el usuario y ademas permite la utilizacion de coprocesadores aritmeticos en las computadoras compatibles con la IBM PC.

3.- SCI-MATE

Es un recuperador personal de datos.
Los archivos de SCI-MATE son muy flexibles, permiten adaptar informacion textual o resumenes, notas de investigaciones, citas bibliograficas, etc.

No necesita aprenderse algun lenguaje tecnico para poderlo utilizar, solamente se le debe indicar (en ingles) al programa que formato de impresion se utiliza para cada registro del texto. Pueden crearse campos etiquetados para capturar en base a ellos, o tambien puede capturarse texto completo sin respetar ningun formato. Los registros pueden capturarse y editarse directamente desde el teclado, de algun otro archivo de SCI-MATE, de algun procesador de palabras o si se tiene el modulo opcional de comunicaciones, de alguna base de datos en linea que permita su descarga.

Los registros quedan indexados durante la captura; por lo tanto, son recuperables y no requieren de ninguna utileria adicional, ni de palabras-llave.

Permite buscar por caracteres, palabras o frases que se encuentren en cualquier parte del registro. Esta busqueda de texto-libre es unica en los sistemas recuperadores de informacion. Para las estrategias de busqueda, acepta operadores booleanos, truncacion y cadenas de palabras.

Caracteristicas:
- Almacenamiento con acceso aleatorio.
- Sistema controlado por menus y subsistema tutor.
- Almacenaq desde uno hasta 1900 caracteres sin requerir de una pre-indexacion o seleccion de palabras-llave para encontrar la informacion.
- Formato de registro flexible, con un maximo de 20 campos definibles por registro.
- Generador de reportes.
- Actualizacion inmediata del archivo maestro.
- Actualizaciones en linea.

Creado en 1978 por el Instituto de Investigacion Cientifica (ISI) para resolver problemas a los bibliotecarios.

4.- MICRO QUESTEL
Permite la recuperacion de informacion, captura de documentos, busqueda y despliegue en pantalla o impresion.
Permite definir la estructura de la informacion incluyendo la validacion, repetitividad y obligatoriedad de cada campo.

El modulo de captura, cuenta con los elementos necesarios para la validacion de la informacion contra valores o rangos predefinidos en la definicion de la base de datos.
La indexacion que ofrece MICRO QUESTEL actualiza automaticamente el diccionario que puede consultarse en cualquier momento de la busqueda.

La busqueda de informacion por modo conversacional permite la toma de decisiones y cambios de estrategia para utilizar los terminos mas apegados a la informacion solicitada.

Los despliegues de informacion pueden dirigirse a la pantalla, la impresora o a un archivo en disco, de acuerdo a las necesidades del usuario. La presentacion o formato de la informacion desplegada, se especifica previamente (margenes, lista o encabezados, odenamiento, etc.).

Los archivos relacionados con el recuperador de informacion (tablas de verificacion, mensajes, diccionario de terminos, etc.) pueden desplegarse o imprimirse. Los mensajes pueden tambien personalizarse.

Cuenta ademas con un modulo que permite almacenar la informacion de una consulta externa para que pueda usarse fuera de linea y con esto reduzca el costo de telecomunicaciones.

La opcion de Carga permite crear, corregir y actualizar localmente la informacion y posteriormente transmitirla a un servicio externo de consulta.

MICRO QUESTEL permite la carga de informacion via su modulo de captura, por descarga de informacion o con una interfase semejante a WORDSTAR.

5.- INMAGIC
Es un sistema administrador de archivos que ha sido disponible para minicomputadoras desde 1980. Actualmente existe la version para microcomputadoras. El paquete es poderoso, flexible y relativamente facil de utilizar. Fue desarrollado por Warner-Eddison Asociados, cuya firma se especializa en aplicaciones para bibliotecas.

El sistema acepta comandos directos o despliega menus y pantallas de ayuda si se necesitan. INMAGIC acepta operadores Booleanos AND, OR y NOT, y puede restringir una busqueda a aquellos registros que satisfagan el criterio de seleccion solicitado, (por ejemplo, mayor, menor, igual, etc.). Tambien permite recuperar por palabras o raices de palabras que ocurren dentro de un campo. Acepta parentesis para determinar el orden de ejecucion. En particular, INMAGIC ofrece 3 comandos adicionales, CW (containig word), CS (containing stem) y ST (start with stem).

Las busquedas pueden realizarse en pasos, cada cual modificando el resultado obtenido en el paso anterior. Las estrategias pueden almacenarse y ser llamadas posteriormente. Puede buscarse aun en campos no indexados, pero con la advertencia del tiempo que tardara en recorrer el archivo en forma secuencial.

La salida de la informacion puede hacerse de 3 formas, a la impresora, a la pantalla o a un archivo ASCII. Para cualquiera de las salidas elegidas, puede ordenarse la informacion de acuerdo a

algun campo, ya sea en forma ascendente o descendente, o solicitarse que se respete algun formato previamente disenado.

INMAGIC puede indexar hasta 50 campos, compuestos por terminos, por palabras claves o ambos. Permite la captura en BATCH de la informacion al archivo maestro.

6.- BIBLIOTK

Es un programa de Apple II o IIe para la administracion de informacion bibliografica de una coleccion personal, permite la creacion, edicion,almacenamiento, busqueda, recuperacion, ordenamiento e impresion de listas de fichas. Es un sistema manejado a traves de menus y facil de operar, ademas es copiable con propositos de respaldo.

BIBLIOTEK fue disenado basicamente para investigadores y escritores. Una de sus principales caracteristicas, es la posibiblidad de imprimir las fichas recuperadas, bajo algun formato programado por el usuario y posteriormente imprimirla con otro formato diferente.

Un disco permite el almacenamiento de aproximadamente 500 fichas, pero es multivolumen, por lo tanto, permite usar varios discos con informacion de la misma base de datos.

La informacion puede buscarse usando palabras llave, nombre de autores o cualquier otro criterio. La lista de documentos, generada por la busqueda bajo cierto criterio puede editarse o imprimirse.

CAPITULO 3

Consideraciones Generales para el Diseño del Sistema

3.1 ESTUDIO DE VIABILIDAD DEL SISTEMA

Se entiende por estudio de viabilidad, al conjunto de investigaciones orientadas al establecimiento de una base que permita decidir sobre la posibilidad y conveniencia de utilizar sistemas de computacion, sustituyendo o auxiliando a los procesos manuales o en la mejora de los sistemas existentes.

Es en escencia un instrumento de planeacion y control en el desarrollo y aplicacion de la informatica.

La ejecucion de las actividades necesarias para el desarrollo del estudio, sin una base metodologica adecuada, puede representrar riesgos de diversa indole debido a la magnitud y complejidad de las tareas a desarrollar.

Las tareas por realizar pueden dividirse en tres modulos.

Modulo I. Diagnostico de la situacion actual.
Modulo II. Determinacion de los requerimientos.
Modulo III. Seleccion del sistema de computacion.

Modulo I. DIAGNOSTICO DE LA SITUACION ACTUAL.

Se debe realizar un estudio detallado de las actividades y procedimientos que se realizan para determinar las necesidades actuales, se recomienda hacerlo de lo general a lo particular.

La investigacion que debe llevarse a cabo exige contemplar los siguientes puntos:

- Funciones y objetivos
- Atribuciones legales
- Reglamentos internos y lineamientos generales
- Sistemas de trabajo
- Areas funcionales
- Estructura organica
- Estructura programatica
- Recursos humanos, materiales y funcionales
- Sistemas de organizacion

Para la determinacion de requerimientos de informacion se deben identificar los tipos de datos que maneja cada area de la institucion asi como la relacion que existe entre ellas.

los diferentes tipos de informacion que maneja cada area podrian fijarse po la aplicacion de la misma, esta puede estar orientada a la administracion, produccion, investigacion cientifica, etc..

La interrelacion que puede existir entre estas areas en materia de informacion, se puede clasificar como:

1 - Unidad generadora
2 - Unidad usuaria
3 - Unidad tratadora
4 - Unidad diseminadora

Para el desarrollo de una base de datos publica es importante desglozar las unidades generadoras y usuarias para permitir la ubicacion perfecta del contexto, contenido de informacion, beneficio social, comercializacion, etc.

Para las unidades generadoras deben estudiarse los siguientes puntos:

- Informacion que generan
- Forma de generar informacion
- Vida util
- Niveles de agregacion
- Sistemas de actualizacion y periodicidad de los mismos
- Tratamientos que requiere
- Origen de la informacion
- Volumen de informacion
- Procesos a que se somete la informacion
- Canales de transmision
- Metodos de clasificacion
- Formas de representacion
- Sistemas que se emplean para la generacion.

Para las unidades usuarias deben estudiarse los siguientes puntos:

- Informacion que utilizan
- Frecuencia con la que la utilizan
- Problemas en la obtencion de la informacion
- Sistemas de archivo
- Uso que le dan
- Estimacion del numero de usuarios
- Necesidades adicionales que no se resolveran
- Deficiencias en oportunidad calidad, cantidad y presentacion.

Para lograr identificar los problemas existentes en la unidad de informatica hay que elaborar un diagnostico detallado para encontrar las funciones que desarrolla, los instrumentos que utilizan y la relacion que guardan con el resto de la institucion. Los puntos que se recomiendan desarrollar son los siguientes:

- Organizacion
    - Adecuacion de la unidad
    - Objetivos, funciones y atribuciones
    - Estuctura organica interna
    - Reglamentos de trabajo internos
    - Estructura programatica de la unidad
    - Instrumentacion administrativa
- Recursos humanos
    - Del personal
    - De la unidad
- Medio ambiente de trabajo
- Infraestructura fisica y logica de informatica
    - Sitemas de programacion
    - Soporte (del proveedor)
    - Equipo fuera de linea
    - Equipos de apoyo
    - Mantenimiento
    - Bienes de consumo
- Contratacion de servicios externos
    - Asesorias y consultorias externas de servicios y costos promedio
    - Renta o utilizacion de equipos externos (razones y

costos)
  - Servicio de mantenimiento, instalacion o reparacion de equipo
- Compromisos y erogaciones
- Cobertura y eficacia de los servicios de informatica
  - Clave y nombre de la aplicacion
  - Tipo de dispositivos fisicos que utiliza
  - Sistemas especiales de manejo de datos que emplea
  - Numero de programas y lenguajes utilizados
  - Frecuencia de uso de la informacion
  - Medios de consulta mas frecuentes
  - Tipos de acceso a la recuperacion
  - Diferentes tipos de salida (formatos)
  - Volumen aproximado de impresion por proceso
  - Tiempo promedio de proceso por programa
  - Numero de archivos de la aplicacion
  - Frecuencia de obtencion de respaldos de cada archivo
  - Sistema al que pertenece
  - Usuarios de la aplicacion
  - Antecedentes, descripcion y justificacion
  - Tiempo de vigencia
  - Fecha de instalacion de la aplicacion
  - Descripcion y numero de los archivos que maneja
  - Cantidad, longitud y formato de los registros de cada archivo
  - Forma de captura de los datos
  - Periodicidad de la actualizacion y volumen aproximado
  - Procesos a que se someten los datos
  - Nivel de documentacion
  - Analisis de costos
  - Problemas que aun no han sido superados
  - Lineamientos de seguridad en general
- Estadisticas de funcionamiento del equipo
  - Indice de fallas del equipo
  - Tipos y causas mas frecuentes de las fallas
  - Frecuencia de elaboracion de trabajos no previstos
  - Existencia de rutinas de contabilidad de uso del sistema
  - Promedio de aplicacion mas usada
  - Frecuencia de obtencion de estadisticas del sistema.

Modulo II. DETERMINACION DE REQUERIMIENTOS.

El producto de esta fase debera reflejar en terminos de cantidad y calidad las caracteristicas indispensables y deseables del equipo fisico, sistemas de programacion, equipo adicional o complementario, recursos humanos, materiales, servicios externos y tiempo requeridos para el desarrollo, implantacion y operacion de los sistemas.

A continuacion se describen algunos puntos de utilidad para el desarrollo:

1. Determinar la capacidad promedio de memoria necesaria y su distribucion para multiproceso y/o multiprogramacion con base en los calendarios de operacion de las aplicaciones

analizadas y tiempos estimados para pruebas de las nuevas aplicaciones a desarrollar, horarios de operacion del sistema, etc.

2. Determinar el numero y tipo de unidades de entrada y salida en linea necesarias, lectoras, perforadoras e impresoras, con base en los volumenes de datos de entrada e informacion de salida de las aplicaciones.

3. Determinar el numero y caracerisiticas de unidades de cinta magnetica, con base en las necesidades de las aplicaciones coincidentes en proceso.

4. Determinar la cantidad de unidades, caracteristicas y distribucion de areas en disco, con base en: el volumen de informacion por archivo, necesidades de extraccion de datos, potencial de crecimiento, volatilidad de los datos, indice de actividad, organizacion de los archivos, convergencia de aplicaciones en proceso, concurrencia simultanea a un mismo archivo, las caracteristicas del proceso de cada actividad, espacio ocupado por el sistema residente, areas de trabajo para clasificacion, archivos de paso y areas para colas de entrada y salida.

5. Determinar el tipo y numero de terminales con base en los requerimientos de los diferentes usuarios, su obtencion, su ubicacion geografica, el tipo de aplicacion de cada uno de ellos y los volumenes de entrada y salida de informacion.

6. Determinar el tipo y cantidad de unidades especiales de acuerdo al volumen, tipo de informacion y frecuencia de uso.

7. Determinar los tipos y cantidad de cada uno de los diferentes equipos de digitacion, con base en el volumen y la frecuencia de captura de datos, la productividad del personal en digitacion y las jormadas de trabajo del mismo.

8. Determinar el tipo de sistema operativo, con base en los diferentes requerimientos de las aplicaciones y el tipo de configuracion del equipo fisico, caracteristicas de adelanto y grado de confiabilidad de los disponibles en el mercado.

9. Seleccionar los diferentes lenguajes necesarios, con base en el tipo de funciones y caracteristicas especificas de las aplicaciones, compatibilidad con otros equipos y nivel de revision.

10. Determinar las necesidades en programas de servicio (utilities) con base en los requerimientos de apoyo de las aplicaciones, sistemas de seguridad en general, facilidades de utilizacion, grado de confiabilidad y eficiencia.

11. Determinar los requerimientos de paquetes epeciales, con base en las caracteristicas de aplicaciones, archivos, procesos y equipos.

12. Determinar todos aquellos requerimientos necesarios para las comunicaciones, en caso de ser sistemas de teleproceso, tales como tipo y numero de modems, controladores, caracteristicas de las lineas de comunicacion, modos de transmision, etc.

13. Estimar, de acuerdo a los volumenes de informacion de entrada y salida, almacenamiento, periodicidad y vida util de la misma, el consumo y las existencias permanentes necesarias: de tarjetas, formas continuas, discos y cintas de repuesto.

14. Elaborar el programa de actividades para la implantacion de los sistemas, siendo este, un resumen de los que fueron presentados analiticamente en la seccion anterior.

15. Determinar los recursos humanos que requiere el desarrollo, la implantacion y la operacion de los sistemas y en general un resumen de lo expuesto analiticamente en el inciso h) de la seccion anterior.

16. Caracteristicas de las instalaciones del centro de procesamiento.
    a) Dimensiones y caracteristicas del local.
    b) Caracteristicas de equipo adicional: piso falso, aire acondicionado, planta de luz, equipo de emergencia, estanteria, cortadoras y separadoras de formas continuas, etc.

Alternativas de solucion

Si ya se conocen los requerimientos de equipo, deben considerarse como altenativas de solucion las siguientes:

1. Utilizacion de equipo del sector al que pertenece la dependencia, o de alguno disponible dentro de otro sector inclusive.

2. Incrementar en lo necesario el equipo actual de informatica de la dependencia.

3. Substituir el equipo actual de la dependencia por aquel que reuna las caracteristicas necesarias.

4. Contratar un equipo adicional para satisfacer los requerimientos.

Implicaciones y presentacion de alternativas.

De acuerdo a lo tratado anteriormente, es necesario fijar las necesidades de organizacion, incremento o capacitacion de recursos humanos y materiales, nuevos sistemas de trabajo, necesidades adicionales de instalaciones e implicaciones de tipo presupuestal, entre otras.

MODULO III. SELECCION DEL SISTEMA DE COMPUTACION

Esta fase es muy delicada, ya que el equipo que se decida adquirir, en caso de ser la mejor solucion, debera reunir los requisitos y satisfacer todas las necesidades actuales.

Para esta tarea se sugiere

- Hacer un concurso de proveedores de sistemas de computacion existentes en el mercado,
- Proporcionar a cada proveedor participante, la informacion suficiente para la elaboracion de su propuesta, misma que ha de tener mas de una alternativa en configuracion, soporte y tipo de operacion, renta, compra, renta con opcion a compra, etc.,
- Que se establezcan las condiciones de presentacion y fecha limite para la entrega de propuestas,
- Analizar y evaluar cada una de las propuestas en forma detallada documentando los resultados parciales y totales del analisis y evaluacion
- Seleccionar la propuesta que en todos sus aspectos cumpla plenamente con las condiciones requeridas.

Con el objeto de facilitar el analisis de las propuestas y garantizar los mejores resultados, se ha juzgado conveniente dividirlo en cuatro grandes grupos:

a) equipo fisico
b) sistemas de programacion
c) soporte
d) presupuesto

A continuacion se presenta una lista de los puntos mas importantes que deberan ser confrontados para cada concepto de los cuatro grandes grupos.

1. Equipo fisico
   - Unidad central de proceso
   - Unidades de entrada
   - Lectoras de tarjetas
   - Lectoras opticas de caracteres
   - Unidades de salida
   - Impresoras
   - Perforadoras de tarjetas
   - Graficadores
   - Terminales graficadoras
   - Unidades de entrada y salida
   - Terminales de rayos catodicos o pantallas de video
   - Consolas de impresion
   - Terminales de audio-respuesta
   - Unidades de almacenamiento
   - Unidades de cinta magnetica
   - Unidad de discos magneticos, acceso directo
   - Equipo de digitacion
   - Perforadoras de tarjetas
   - Grabadoras de cinta

Grabadoras de discos
Entrada directa
Equipo especial

2. Sistemas de programacion
   Sistema operativo
   Lenguajes de programacion
   Programas de servicio (utilities)
   Programas operativos del sistema
   Paquetes especiales

3. Soporte
   Caracteristicas del proveedor
   Asistencia tecnica
   Asistencia educacional
   Soporte de mantenimiento
   Soporte de maquina

4. Presupuesto
   Recursos humanos
   Recursos materiales
   Otros gastos

Prueba de los sistemas propuestos
Una vez identificadas las caracteristicas e implicaciones de los diferentes sistemas de computacion, es conveniente realizar una serie de pruebas conocidas como "BENCHMARK", sobre la productividad de los mismos en tiempos de proceso, con la ejecucion de una mezcla de aplicaciones que sean representativas de las necesidades del usuario tanto en condiciones de procesamiento como en volumen.

Finalmente deberan analizarse los resultados obtenidos y seleccionar aquellas propuestas que cumplan con los requisitos.

3.2 BASES DE DATOS SEPARADAS

Un sistema puede estar compuesto por dos grandes conceptos de bases de datos; es decir, bases de datos como soporte de informacion y bases de datos como administrador de movimientos, ya que hay razones suficientes para no intentar incorporar la informacion en un solo tipo de estructura de bases de datos.

Las razones son las siguientes:

1) Las estructuras fisicas usadas comunmente en un sistema de informacion, son de una complejidad y magnitud, que la actualizacion de la informacion es lenta, debida a la generacion de indices y apuntadores usados. Mientras que, en los sistemas operantes o administradores de movimientos, las estructuras de datos usadas son muy simples y esto permite la actualizacion de los datos en tiempo real.

2) Es muy dificil insertar datos nuevos y borrar los anteriores de la base de datos del sistema de informacion, excepto por

una operacion secuencial fuera de linea. Los sistemas operantes tienen sus estructuras de datos de tal manera que los registros pueden insertarse o borrarse facilmente.

3) Los sistemas operantes usualmente contienen la ultima transaccion. Mientras que, los sistemas de informacion pueden ofrecer informacion con mas de 24 horas de diferencia sin tener consecuencias.

4) Los sistemas operantes deben manejar un alto tope de transacciones, tal que, las estructuras de datos que permiten el acceso rapido toman una gran importancia para competir con el volumen. Los sistemas de informacion que contienen los mismos datos, usualmente manejan un numero relativamente menor de preguntas.

5) Los sistemas de informacion deben contener la informacion mas relevante o en forma resumida sin todos los detalles y precision que se manejan en los sistemas operantes.

3.3 TELEPROCESAMIENTO DISTRIBUIDO

3.3.1 Bases de datos distribuidas

Estas son algunas facilidades y ventajas que presentan las configuraciones de bases de datos distribuidas.

- COSTO. El costo de almacenamiento de datos, en pequena escala, es menor que el costo de su transmision.

- CARGA. La carga por trafico de tareas y uso de los datos es generalmente menor que la capacidad del software.

- ADMINISTRACION LOCALIZADA. La organizacion local de los datos es totalmente responsable de su precision en contenido y custodia.

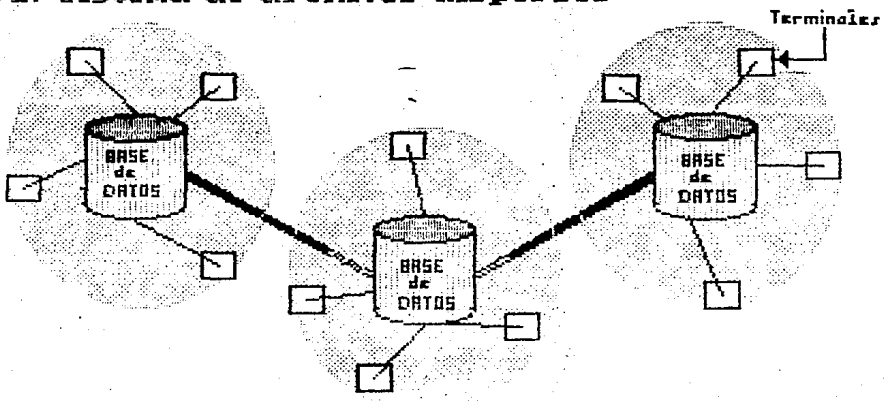
- DISPONIBILIDAD. Dado que la actualizacion, operacion, falla, o cualquier otro tipo de interrupcion del sistema de informacion no ocurriria simultaneamente, una computadora puede respaldar a otra y continuar en la normalidad.

- SEGURIDAD. Con un esquema de soporte entre computadoras es posible regenerar los archivos de otra computadora, que por alguna causa haya destruido los datos.

- REDES DE COMPUTADORAS. Es posible conectar computadoras muy diferentes entre si a grandes distancias, permitiendo la consulta o vinculacion logica de la informacion en un sistema total e integrado.

De las razones anteriores surgen cuatro diferentes categorias en el uso de bases de datos distribuidas, en la figura 3.1 se muestran los cuatro esquemas mas comunes en configuraciones

1: Sistema de archivos dispersos



2: Inteligencia distribuída

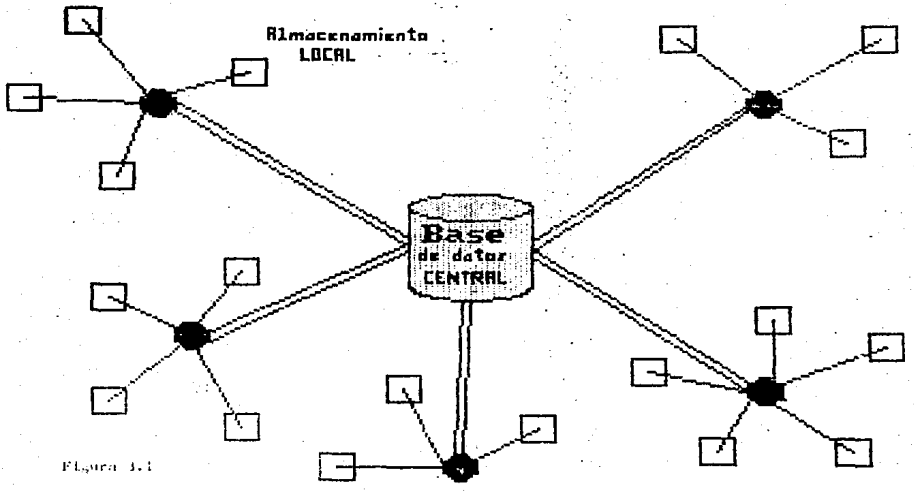
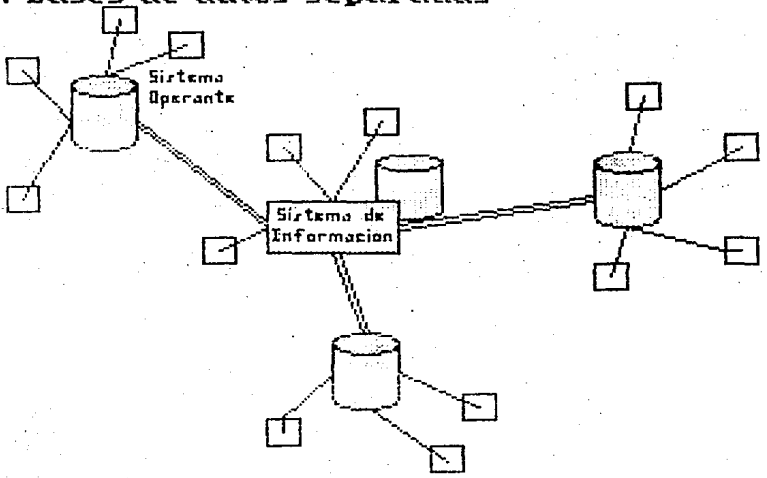


Figura 3.1

3: Bases de datos separadas



4: Red de computadoras

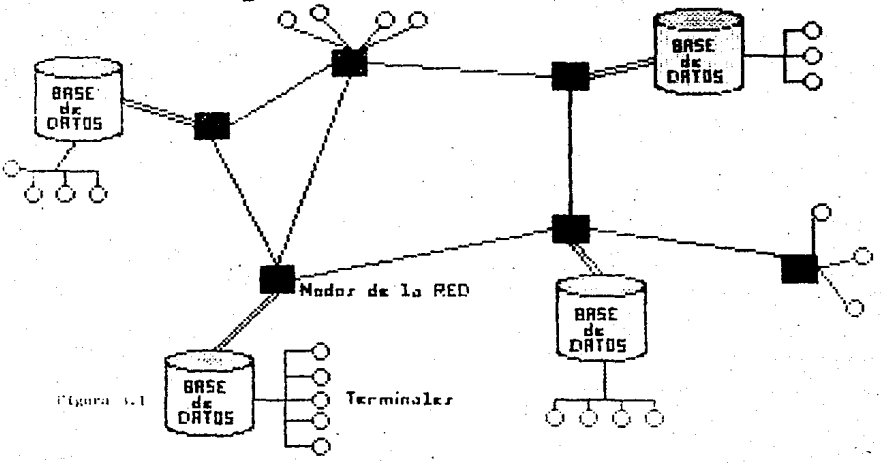


Figura 1.1

geograficamente distantes.

1) Los tipos de datos y las estructuras son iguales, pero el almacenamiento fisico esta disperso. Puede usarse el mismo esquema en cada una de las bases de datos separadas.

2) Se emplean pequenos perifericos para enlazarse a una base de datos central. Los subesquemas usados en las unidades de almacenamiento perifericas pueden derivarse de la estructura de datos del sistema central.

3) Pequenos sistemas que alimentan parte de sus datos, a un sistema central. Cada sistema tiene su propio esquema, sin embargo, los formatos y tipos de datos estan relacionados estrechamente y planeados de manera integral.

4) Es una red de computadoras en la que tanto las computadoras y las bases de datos son totalmente heterogeneas.

Los programas necesarios para permitir la independencia fisica de los datos con datos distribuidos son muy complejos. La mayoria de los programas para bases de datos no permiten aun una verdadera independencia de datos en un sistema distribuido.

La situacion se complica aun mas si los esquemas y subesquemas se encuentran en sitios separados geograficamente, tal y como se muestra en la figura 3.2

Paulatinamente vendran desarrollandose programas para bases de datos distribuidas, ofreciendo independencia logica, fisica y posiblemente geografica.

Cuando una solicitud de informacion requiere de datos no almacenados en la computadora donde se origina la transaccion, existen tres posibles soluciones:

1) Transmitir la transaccion al sitio donde se encuentran los datos y procesarlos ahi.

2) Transmitir los datos al lugar donde se origina la transaccion para procesarlos localmente.

3) Transmitir tanto los datos como la transaccion a un tercer sitio para su procesamiento.

La eleccion de una de estas posibilidades, dependera del volumen de los datos y la frecuencia con que se requiera transmitirlos. Es muy importante en algunos casos, mantener el control de los datos, principalmente si los datos se actualizan por transacciones que provienen de diferentes sitios, deben permanecer en un solo lugar para poder controlar los conflictos y llevar control de su lectura y actualizacion ("dead locks") eficientemente.

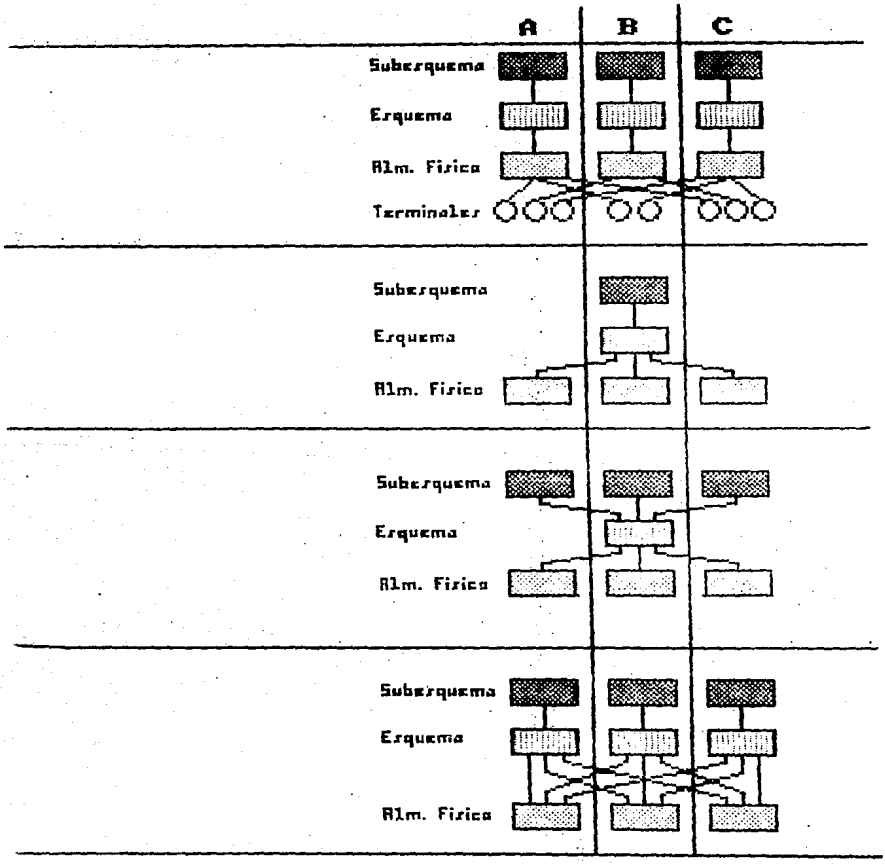


Figura 3.2 Esquemas y subesquemas en sitios separados geográficamente

3.4 ADMINISTRACION Y SEGURIDAD DEL SISTEMA

3.4.1 CONFIABILIDAD

3.4.1.1 Concepto de Confiabilidad

La confiabilidad se logra cuando un sistema de computacion, tanto el equipo como los programas, siempre producen resultados correctos.

" Los metodos actuales de depuracion de programas mediante pruebas son evidentemente inadecuados, ya que por lo comun estan limitados a la verificacion de unos cuantos calculos muestra. El analisis de la estructura de un programa puede crear un plan de prueba que asegure que cada seccion de programa se ejecute una vez, pero rara vez puede lograrse probar cada posible secuencia de ejecucion, ya que casi siempre el numero de combinaciones es demasiado grande. Los valores extremos de datos de entrada a menudo pueden inducir errores y son utiles para efectuar pruebas.

La aplicacion de tecnicas de programacion estructurada debera llevar a una reduccion y a una mejor identificacion de los errores ocultos, pero aun asi no se eliminaran. La verificacion formal de programas intenta eliminar fallas. Estas tecnicas dependen de una especificacion detallada de las transformaciones de datos, por lo que resulta esencial un modelo comprensivo de la base de datos."

En lo sucesivo se analizaran los tipos de fallas mas comunes en las bases de datos. Los procedimientos de vigilancia de la integridad tmbien pueden ayudar a mantener la confiabilidad de una base de datos.

Para producir resultados correctos se requiere tanto de datos correctos como de programas correctos, forzando al sistema a efectuar los algoritmos correctamente. Dentro de cada una de estas areas es necesario dividir el problema en dos sub-problemas:

1. Detectar la existencia de una falla o mostrar la ausencia de fallas.

2. Dada la deteccion del error, es necesario disponer de un medio para corregirlo y recuperarlo.

PROBABILIDAD DE FALLA.

En los sistemas de computacion compuestos por muchas partes expuestas a fallas, y donde el exito de una operacion exitosa depende de que cada una de ellas funcione correctamente. Sean q partes con probabilidad individual de falla pF, la probabilidad de operacion con exito es

$$pEXITO=(1-pF)^{q}$$

En la practica, muchas de las fallas de circuitos de computadoras son transitorias y no se repetiran durante largo tiempo. Tales errores pueden deberse a una rara combinacion del estado del sistema, variaciones de la energia, ruido electrico o acumulacion de electricidad estatica, etc.. En estos casos puede intentarse nuevamente la ejecucion del proceso y se obtendra el resultado esperado.

3.4.1.2 Redundancia

La redundancia se obtiene cuando las unidades de datos se expanden para proporcionar mas informacion que la estrictamente necesaria. Puede verificarse la consistencia interna de las unidades de datos. Si se encuentra que estan en error, pueden corregirse.

PARIDAD

Es una forma sencilla de verificacion de la informacion, y con el minimo costo ya que implica solamente agregar un bit en las unidades elementales del dato. Por lo general se agrega un bit de paridad a los caracteres en cinta o en la memoria de la computadora. Al numero de bits con valor 1 dentro de un caracter, se le llama peso Hamming. Si el caracter esta representado por ocho bits, el noveno sera el de paridad. Cuando la combinacion de bits del caracter con el bit de paridad es non, se dice que la paridad es non; cuando la combinacion de los bits del caracter con el bit de paridad sea par, se dice que la paridad es par. Es mas frecuente encontrar sistemas con paridad non para forzar a que los datos cuyo peso Hamming sea igual a cero, cambien su estado al menos una vez en la transmision del caracter.

DUPLICACION

La duplicacion de informacion es una forma muy sencilla de lograr redundancia. Por ejemplo en el caso de una cinta, se pueden hacer pasar los datos por diferentes canales y hacer una comparacion de ellos, facilitando una alta determinacion de errores. Desde luego que el costo de duplicar la informacion es bastante alto.
Rara vez resulta factible el mantenimiento de archivos totalmente duplicados en disco en las computadoras de proposito general. El costo de esta duplicidad es relativamente alto, dado el grado de confiabilidad de los dispositivos de almacenamiento actuales. La redundancia total de los archivos aumenta el tiempo necesario para la actualizacion notablemente.

Existen varios metodos para la deteccion de errores en secuencias y evitar el arrastre del error.

Los datos de un archivo son mas vulnerables cuando se encuentran en la memoria principal, especialmente cuando los sistemas de computacion se emplean para muchas tareas paralelas o sufren cambios frecuentes.

Las sumas de verificacion tomadas en el contenido de un buffer pueden utilizarse para verificar que los datos no se hayan modificado con otro procedimiento que no sea el aprobado. El mantenimiento y frecuente calculo de sumas de verificacion puede ser bastante costoso.

Otra posibilidad para proteger los buffers es el empleo de barreras. Estas consisten en un par de codigos que tengan posibilidades de ocurrir con poca frecuencia. Esto nos permite comparar los bits de las barreras y hacer calculos de una manera mas rapida y barata, dado que no se compara la longitud completa del carater.

3.4.1.3 Confiabilidad de las transacciones

La confiabilidad de una base de datos se selecciona con la conclusion exitosa de las transacciones. Una transaccion se define como el programa que modifica el estado de la base de datos de una manera que, si el estado de dicha base de datos era correcto y la transaccion no introduce un error debido a una falla o a datos incorrectos de entrada, el estado de la base de datos vuelve a ser correcto despues de que la transaccion haya concluido con exito. Si la transaccion no concluye adecuadamente ninguno de sus efectos debera ser visible en la base de datos.

Transacciones de dos fases.
Para minimizar el efecto de una falla es mejor separar en dos fases las acciones de transaccion. Durante la primera fase, la transaccion adquiere todos los recursos necesarios para su ejecucion, incluyendo aquellos bloques que necesite actualizar. Tambien realiza las calculos solicitados previos al ingreso. Todos los bloques que no vayan a modificarse, se pueden escribir en la primera fase. Durante la segunda fase, se ejecutan todos los bloques con modificaciones, liberando los recursos asignados en la primera fase.

En el transcurso de la primera fase, la transaccion puede concluir sin haber afectado en forma significativa la base de datos.

Al entrar a la segunda fase, la transaccion debe concluirse. Una falla en esta fase puede afectar la base de datos. Esta falla pudo haber sido provocada por fallas internas en la transaccion, o por alguna causa externa, por el sistema, etc.. Las fallas en la segunda fase se evitan tanto como sea posible, sin embargo, no pueden eliminarse por completo.

En ciertas transacciones puede resultar complicado definir un solo punto de conclusion. Las transacciones largas y complejas pueden estar anidadas en una jerarquia.

MANEJO DE TRANSACCIONES

Usualmente se espera que el sistema operativo cuente con un

modulo manejador de transacciones. Si no lo contiene, esta sera una tarea que debera administrar el sistema de bases de datos. El manejador de transacciones recibe los siguientes mensajes de control durante la ejecucion de una transaccion

- inicio de transaccion
- realizacion de la transaccion o aborto de ella
- transaccion realizada
+ datos.

El manejador de transacciones debera reconocer cada mensaje y actuar de la manera apropiada para cada uno de ellos.

Aborto: El programa ejecutor de alguna transaccion puede iniciar un aborto al encontrar un error. La unica condicion es avisar al usuario y al sistema de ello. El aviso al sistema, en ocasiones, provee las herramientas para una recuperacion y reinicio posterior.

Conclusion: Es la posibilidad de terminar la ejecucion de una transaccion con exito, que por alguna causa fue interrumpida. Se presentan tres alternativas para lograr la conclusion de una tarea.

1. La bitacora de transacciones es un archivo separado que proporciona almacenamiento seguro para los bloques de datos creados en una primera etapa y deberan escribirse en otra posterior. A los bloques que se colocan en la bitacora de transacciones para su escritura posterior, se les llama imagenes posteriores.

   Cuando se solicita la conclusion de reinicio, el manejador copia las imagenes posteriores a la base de datos.

2. Las transacciones pueden mantener sus propias imagenes posteriores. En este caso el manejador ejecuta simplemente un reinicio de la transaccion, cuando se solicita la conclusion.

3. Consiste en la reejecucion de toda la transaccion. Al esperar el punto de conclusion, la transaccion ya ha realizado cualquier verificacion que pudiera haber provocado un aborto interno, de manera que no se reinicien las transacciones erroneas.

Si no es posible concluir una transaccion comprometida, esta debera eliminarse para evitar dejar la base de datos en un estado inconsistente y la transaccion con el requerimiento de recursos.

Cancelacion: Para cancelar o deshacer una transaccion, es necesario que se haya conservado sin cambios un duplicado al menos, del estado anterior de la base de datos. Las facilidades comunmente empleadas para admitir cancelaciones son:

1. Creacion de una nueva version. Todos los datos nuevos se incorporan a un duplicado de la informacion anterior, para no

tener que destruir la version anterior.

2. Bitacora de respaldo. Todos los datos anteriores se introducen en un archivo de bitacora de transacciones. Los bloques guardados en este archivo previo a la actualizacion, se les llama imagenes anteriores.

3.4.1.4 Bitacora de actividades

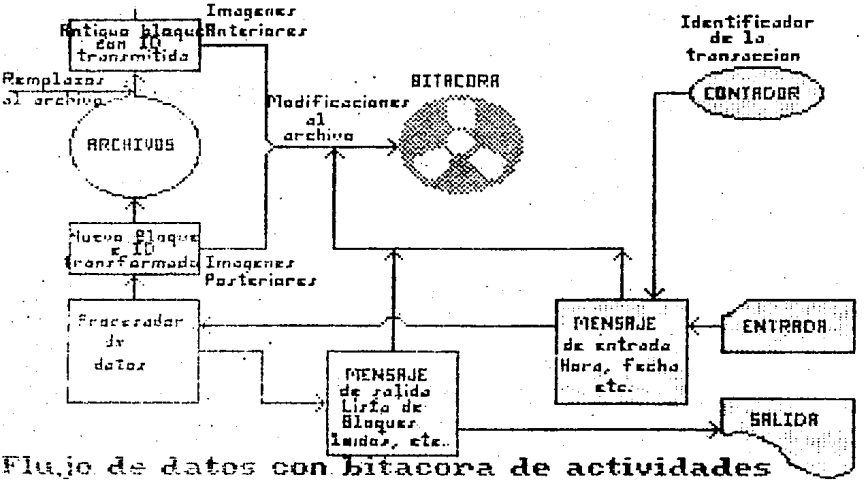
En la restauracion de una base de datos danada, es necesario volver a establecer los archivos sin los efectos de un error.

La restauracion de los archivos al estado anterior al error, borrara el efecto de otras transacciones que hayan provocado cambios concurrentes posteriores a los archivos de la base de datos. Sin embargo, es posible que las transacciones subsecuentes hayan utilizado datos erroneos colocados en la base de datos provenientes de un error primario y que puedan generar errores secundarios.

Para permitir la restauracion total puede conservarse una bitacora que contenga todas las actividades que afectan a la base de datos. Los datos se agregan en las bitacoras en forma secuencial, impidiendo que esta se regrese y acepte una modificacion, a menudo se utilizan las cintas como almacenamiento principal de las bitacoras aun cuando tambien la tendencia de usar discos para reducir costos laborales y tiempo empleado por registro esta en aumento. Esto se ilustra en la figura 3.3.

El objetivo fundamental de una bitacora es reestablecer el archivo con error a su estado anterior o cuando menos identificar el error, esto se hace mediante el seguimiento inverso de los pasos que hayan modificado los archivos y hayan quedado registrados en la bitacora de actividades. Para lo cual, los elementos requeridos en la bitacora son: las imagenes anteriores como apoyo para verificar que todos los datos que han modificado el estado de la base de datos se han identificado, y las imagenes posteriores para un posible punto de reinicio del proceso de la transaccion. Los elementos adicionales que deberan colocarse en la bitacora son la entrada de transacciones, la salida de transacciones y una guia del proceso interno de la transaccion, esta ultima se refiere a una lista de direcciones de memoria secundaria o registros de los archivos modificados.

Respaldo: En la practica, la restauracion mediante imagenes posteriores depende de la disponibilidad de una copia de respaldo de alguna version anterior de la base de datos, preferentemente la ultima. Es posible generar periodicamente copias de respaldo y conservar una serie de versiones anteriores. Cada copia debera estar identificada por la fecha y hora de la ultima transaccion incluida. Una copia de respaldo debe generarse mientras la base de datos esta en reposo, ya que las actualizaciones durante el copiado pueden provocar que la copia sea inconsistente. En las bases de datos muy grandes es posible que no haya tiempos de



Flujo de datos con bitacora de actividades

Figura 1.1 Bitacora de actividades

reposo lo suficientemente largos como para realizar una copia. En este caso se recomienda usar bitacoras de movimientos.

3.4.1.5 Escenario para la recuperacion de errores

Los cuatro pasos elementales para la recuperacion de un estado de error son:

1. Deteccion del error,
2. Determinacion de la fuente de error,
3. Ubicacion de errores secundarios y
4. Aplicacion de correcciones.

La deteccion y determinacion de la fuente del error.
Es posible distinguir una variedad de puntos de entrada en el proceso de recuperacion de un estado de error.

Si el problema es una caida del sistema, este se detecta facilmente, ya que la maquina se detiene y las terminales ya no responden de la manera esperada. La ventaja que presenta este tipo de fallas es que el error se detecta de inmediato.

Si el error incide en algunos de los mecanismos basados en redundancia, el error probablemente no se haya propagado en exceso, si fue a causa de un registro equivocado, se rastrea en forma inversa hasta saber la ubicacion fisica del registro y permitir la determinacion del error. La correccion manual del elemento incorrecto tiene que relizarse si el error constituye una fuente de error de entrada.

Si el error se debe al mal funcionamiento del dispositivo de almacenamiento, resulta suficiente con localizar la imagen posterior que se guardo al iniciar la transaccion y se puede continuar remplazando el bloque incorrecto.

Seguir el proceso en sentido inverso se complica cuando el error fue el resultado de un calculo, en vez de un valor elemental de entrada. En este caso debe contarse con los programas que realicen las funciones inversas a partir del resultado y algunos datos independientes.

3.4.2 PROTECCION DE LA PRIVACIA

Este tema cubre dos aspectos importantes:

- El Impedimento de acceso a los usuarios no autorizados a la informacion.

- Compromiso de ofrecer un servicio adecuado que no estropee la calidad de la informacion, sino que la refleje fielmente.

3.4.2.1 Componentes del problema de proteccion

Los tres elementos fundamentales en un esquema de proteccion son:

- Los usuarios con acceso a la base de datos,
- El tipo de acceso deseado y
- Los elementos a los que realizara el acceso

La combinacion de estos elementos permite la identificacion, alcances y rutas de acceso permitidas y limitadas del sistema de informacion.

Se definiran algunos conceptos para la homogeneizacion de los terminos que se emplearan en adelante.

Entorno: Existe un area con perimetro definido, conocido como sistema de la base de datos.

Usuarios e intrusos: Dentro de esta area puede haber individuos aceptados y preferentemente identificados, individuos disfrazados de usuarios validos, e intrusos.

Alcance limitado: El sistema desconoce la identidad de los individuos del mundo exterior.

Privilegios: Todos los elementos que estan protegidos hasta cierto punto mientras se encuentren dentro del area del sistema de la base de datos, y perderan toda su proteccion al salirse de ella.

Confiabilidad: Un prerrequisito para lograr que la proteccion de la base de datos sea de alto nivel dentro del sistema.

La proteccion de los datos requiere establecer un control en la lectura, escritura y empleo de la informacion.

Mientras mas proteccion se implante para reducir las violaciones accidentales y deliberadas de acceso, mayor sera el costo del sistema. Cuando el costo de la proteccion excede el valor de los elementos protegidos, se ha llegado a un extremo.

El valor de la proteccion de un dato puede determinarse de acuerdo al beneficio obtenido por un intruso deliberado o de acuerdo a la perdida sufrida por el propietario por cualquier causa. El vinculo entre la cantidad de elementos perdidos en el caso de una violacion de privacia y la perdida de informacion puede ser en muchas situaciones no lineal. En un caso, la perdida de un solo hecho podria resultar sumamente incriminatoria en una aplicacion, y la perdida de datos adicionales podria no importar gran cosa despues. En otros casos, el valor de la informacion aumenta con el numero de datos liberados.

En sistemas comerciales se ha estimado que el valor de la proteccion adecuada es del 10% al 20% del costo basico del

procesamiento de datos.

Actualmente se ha reconocido que la proteccion de la privacia y la seguridad del sistema son preocupaciones importantes; de manera que se puede esperar un desarrollo incremental en esta area, tal que se logren sistemas con mejor apoyo para estos puntos y bajen como consecuencia el costo para el usuario.

3.4.2.2 El usuario con acceso a la base de datos.

La identificacion externa de los usuarios con acceso a la base de datos es en primer lugar el nombre, en la forma en que lo introduzcan al sistema. Un usuario con derecho de acceso tambien puede identificarse mediante una clave de acceso. Todas las autorizaciones y privilegios dados a los usuarios con acceso a la base de datos dependeran estrechamente de la clave de acceso o llave del sistema.

Para evitar la entrada de intrusos al sistema, la clave de acceso debe ser muy dificil de imitar o copiar.

Con el fin de proteger el proceso de obtencion de una clave del sistema, cuando la maquina (el sistema operativo) solicita al usuario introduzca su clave de acceso, la clave debera introducirse sin que el sistema la exhiba con el fin de protegerse de los observadores. En general, esta clave de acceso consistira en unas cuantas letras, elegidas por el usuario. Un intruso podria utilizar un metodo de ensayo y error para introducir posibles claves de acceso y lograr entrar. El tiempo necesario para realizar un ensayo sistematico es el principal elemento para desanimar a posibles intrusos. El tiempo esperado para abrir un seguro especifico sin ningun conocimiento previo es:

T(entrar)=(1/2)*(C^d)*t(ensayo)

donde C^d es el numero de combinaciones posibles y t(ensayo) es el tiempo necesario para probar una combinacion.

Ejemplo:
Para una clave de tres letras; d=3, C=26 y el tiempo para la interaccion con el sistema podria ser t(ensayo)=3 segundos, de manera que T(entrar) es aproximadamente siete horas.

Si a esto se le agregan sistemas adicionales como que despues de n intentos se rompa la comunicacion o no permita la introduccion de otra clave hasta despues de un lapso determinado, podria empeorarse la situacion para el intruso notablemente.

Un problema practico es que con las claves de acceso grandes aumenta el grado de complejidad para memorizarlas, provocando que se escriban y probablemente pierdan su privacia; otro problema es que se escogen claves con relacion a iniciales, nombres, diminutivos, animales u otros mnemotecnicos auxiliares, por lo

que resultan faciles de adivinar.

3.4.2.3 Tipos de acceso a los datos

Comunmente encontramos distinciones entre autorizacion para leer y autorizacion para escribir datos. Las cintas, cartuchos y algunos discos tienen algun mecanismo fisico que impide la escritura si no existe la presencia de un objeto. Este control tambien puede seleccionarse desde el sistema operativo.

Se pueden definir siete tipos de privilegios de acceso diferentes, que combinados permiten un alto grado de control de proteccion.

1.- Lectura. El acceso otorga el privilegio de copiar datos contenidos en el dispositivo de quien realizo el acceso. Este privilegio protege a los datos almacenados, pero no a la informacion que contienen.

2.- Ejecucion.- Otorga el privilegio de un programa o procedimiento.
Pueden asociarse privilegios de acceso ampliado a los datos. El texto de procedimientos y los datos leidos por dicho procedimiento no quedan disponibles ni es posible modificar el programa sin contar con privilegios adicionales, lo cual permite proteger la informacion de manera selectiva.

3.- Modificacion.- Este acceso otorga la capacidad normal de escritura. Permite la destruccion de datos almacenados.

4.- Eliminacion.- Permite la destruccion de la informacion que exista. En ocasiones es muy sutil la diferencia entre la eliminacion y el acceso con privilegios de modificacion, pero la diferencia radica en que la eliminacion varia el tamano del archivo.

5.- Ampliar. Permite la adicion de datos al archivo sin la capacidad de destruir otros datos del mismo archivo, sin el privilegio de leer el contenido.

6.- Trasladar.- Es un privilegio que no se tiene comunmente, ya que permite mover datos de un archivo evitando la lectura de su contenido.

Hay casos tipicos en que este privilegio es adecuado para muchas tareas del sistema operativo. El traslado de los datos del usuario a los buffers de salida se realiza a nivel de bloque sin importar su contenido o estructura.

7.- Verificacion de existencia. Permite al usuario conocer de antemano a la lectura la presencia de un dato en el archivo o base de datos. En ocasiones la existencia de un dato desencadena una serie de procesos.

La combinacion de estos privilegios de acceso al sistema ofrecen un sistema completo para el control de la privacia. Para darse una idea de las combinaciones posibles, solo hay que generar una tabla representando la existencia o ausencia del privilegio.

1	2	3	4	5	6	7	NIVEL PRIVACIA
NO	NO	NO	NO	NO	NO	NO	CERO
NO	NO	NO	NO	NO	NO	NO	DIEZ
SI	SI	SI	SI	SI	SI	SI	CIENTO VEINTISIETE

(2 COMBINACIONES)^(7 PRIVILEGIOS) = 128 NIVELES DE PRIVACIA

3.4.2.4 Elementos que deben protegerse

El espacio de datos, al que se dirige un usuario solicitando un tipo de acceso especifico, contiene los siguientes elementos: datos, rutas de acceso, programas de la base de datos y anotaciones de esquemas.

Datos: Un dato es un hecho registrado. Un hecho resulta interesante para el usuario cuando esta relacionado con un conjunto de atributos que lo identifiquen y lo relacionen con el mundo real.

Un dato en si puede requerir de un esquema de proteccion, pero es mas frecuente encontrar agrupaciones de datos que tengan las mismas restricciones. Es tambien como el encontrar sistemas operativos que controlan la informacion a nivel de registro, por lo que es imposible pensar en aislar un elemento en si mismo.

Rutas de acceso. La relacion de un dato con otro, se logra a traves de apuntadores.
Tambien existen apuntadores en la utilizacion de indices. Estos presentan una clase diferente de elementos que se debe asegurar. En muchos casos, es necesario tener diferentes privilegios de proteccion para apuntadores, objetos y para el valor de los objetos.

Existe cierto traslape conceptual entre los apuntadores considerados como objetos de proteccion y la disponibilidad de verificacion de existencia como tipo de acceso. Sin embargo, la omision de cualquiera de ellos puede dejar ciertas brechas en un sistema de proteccion.

Programas. Los programas pueden ser elementos activos en un sistema de base de datos, por lo tanto, su proteccion debe ser

importante. Cuando la proteccion de los datos este completa, se debe proceder de la misma forma con los programas, bajo las mismas reglas. Sin embargo, la clasificacion de los datos como registros con atributo no se aplica. La existencia de texto de programa debe reconocerse en forma especifica.

Esquemas. El esquema de una base de datos es el corazon de la misma y debe protegerse bajo una cierta estructura jerarquica. En este caso, hay que ocuparse de la asignacion adecuada del objeto en un medio de base de datos, cuando la proteccion se proporcione a traves del esquema mismo.

3.4.2.5 Entorno de proteccion

Ya que los servicios y archivos de la base de datos son una parte integral de las facilidades de un sistema de computacion, no es posible limitar las consideraciones de proteccion solo al sistema de la base de datos, pero aun asi habra un area mas alla de la cual la responsabilidad de la proteccion ya no este en manos de la gerencia de la base de datos. Con frecuencia, se descuida la definicion del entorno en que se mantiene la proteccion. Omitir una especificacion del entorno, junto con las especificaciones acerca de la seguridad disenada en la operacion interna del sistema, puede provocar equivocos. Los usuarios de los sistemas de computacion tal vez esten excesivamente deseosos de creer que existe una proteccion adecuada.

Componentes del sistema de proteccion. Con frecuencia sera necesario incluir en las consideraciones el sistema operativo, los subsistemas de entrada y salida, el subsistema de originalidad, asi como al personal implicado en la operacion. En el caso ideal, es posible proteger la base de datos como si fuera una unidad separada. Si se incluye el sistema operativo, es posible que exista menor confiabilidad debido a las cuentas en gran parte y al mayor numero de usuarios. En el estado actual de las medidas de seguridad de los sistemas existentes de computacion, el trabajo realmente secreto se realiza en computadoras no compartidas y en edificios protegidos, mediante personal verificado y asegurado.

Se considera siempre que el acceso a la base de datos solo se logra a traves de dicha base de datos o de un sistema de archivo que reconozca las fronteras de proteccion. En los sistemas en que no exista una bitacora de transacciones como parte de el, se deberan crear archivos de respaldo utilizando programas de utileria que a menudo no tomaran en cuenta las convenciones de proteccion. Las cintas de respaldo, asi como la salida proveniente de ejecucion no autorizada de programas de utileria proporcionan buenas oportunidades para que se burlen las fronteras de proteccion.

Perdida de control sobre los datos. Otro problema en la definicion del entorno de proteccion se debe al intercambio rutinario de informacion.

Las herramientas mas importantes de que actualmente se dispone para impedir el acceso no autorizado son la definicion rigurosa de los archivos, el control de entrada y salida, y la identificacion del personal que ingresa al entorno protegido. Ademas el sistema lleva una bitacora de todos los intentos fallidos o no para lograr acceso, y ayudar en el analisis posterior a los hechos en caso de violaciones. Es necesario que la gerencia este realmente preocupada para que el personal sea consciente de la importancia de la proteccion.

Respuesta a las violaciones. Es necesario considerar cuidadosamente cual sera la accion a realizar cuando se detecte una violacion del mecanismo de proteccion.

Gran numero de violaciones de acceso se debe a depuraciones que realicen los programadores o a oficinistas que se equivoquen al teclear una consulta. Por otra parte, debera llevarse una bitacora adecuada del comportamiento incorrecto en el acceso, para permitir detectar las acciones ilicitas y atrapar en su caso a quienes las realicen.

3.4.2.6 Organizacion de la llave de acceso

El efecto producido por un gran numero de usuarios puede reducirse al establecer categorias de usuarios. Esto da lugar a una tabla que mapee la identificacion individual del usuario a una identificacion de la categoria. Una reduccion del numero necesario de categorias se logra si un individuo puede ser miembro de mas de una categoria, por lo que se necesitaran menos categorias especiales. Por otra parte, los privilegios inesperados pueden ser el resultado de multiples vinculos concurrentes de la categoria. Este problema se puede evitar haciendo que el usuario deba especificar el area de la base de datos a la que desea tener acceso, de manera que solo este activa una categoria a la vez. Con esta organizacion por categorias, es posible identificar de una manera mas efectiva al usuario, por lo que puede resultar mas conveniente que la identificacion por claves. Si el sistema proporciona descripciones detalladas del contenido y enlaces de un registro (como en un esquema), es posible una proteccion detallada. Si no se proporcionan facilidades de este tipo, en general la proteccion estara limitada al nivel del archivo.

El metodo empleado para la proteccion del archivo siempre lo determina el sistema operativo. Dentro de un sistema para identificar los archivos en forma unica, sus nombres recibiran automaticamente un prefijo con un termino de calificacion construido a partir del nombre del usuario o su identificacion de entrada. Por lo tanto, el sistema operativo puede verificar si este nombre compuesto esta permitido para consultar especificamente ese archivo o fue concedido el permiso de tener acceso a esos datos, determinado por el privilegio de acceso.

En los sistemas de bases de datos existen diferentes apoyos del sistema para la proteccion de un archivo, mientras que los sistemas de archivos convencionales se basan exclusivamente en el sistema operativo y en el hardware de la computadora. En los sistemas de bases de datos, se pueden encontrar

- ATRIBUTOS DEL ARCHIVO
- MECANISMOS DE HARDWARE
- ASIGNACION JERARQUICA DE PRIVILEGIOS
- PROTECCION A NIVEL DE OBJETO
- MECANISMOS DE SOFTWARE
- DISTINCION DE USUARIOS

3.4.2.7 Criptografia

Es una tecnica alternativa de proteccion, consiste en transformar los datos en una forma que no proporcione informacion al ser interceptada. Basicamente se dispone de tres metodos:

1.- Codificar la informacion
2.- Transposicion de los codigos que representan los datos
3.- Sustitucion de los codigos que representan los datos

La codificacion de la informacion consiste en representar ciertos datos conocidos por un codigo, ejemplo
1=HOMBRES, 2=MUJERES, impidiendo que la informacion se entienda sin la tabla de conversion.

A las operaciones en que los simbolos basicos se trasponen o sustituyen a fin de volver ininteligibles los datos se le denominan Cifras. Una transformacion de este tipo crea textos cifrados a partir de un texto simple.
Para entender el mensaje, el receptor debera descifrar el texto de acuerdo a las reglas y a una llave utilizada al cifrar los datos. En la figura 3.4 se muestran los componentes de la criptografia.

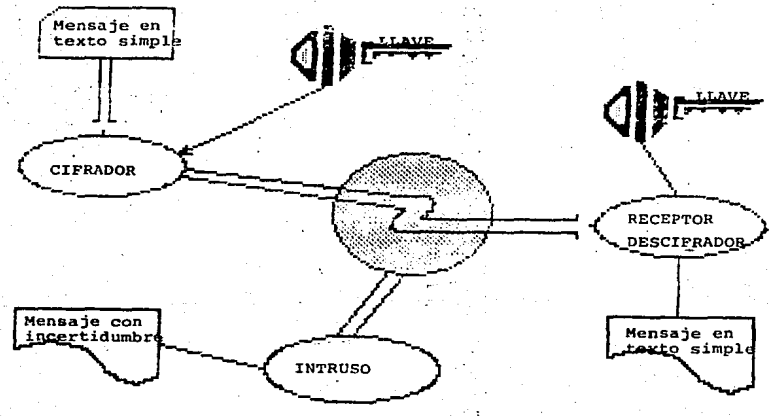


Figura 3.4
ELEMENTOS DE CRIPTOGRAFIA

CIFRADO

Una cifra de transposicion opera sobre un grupo de caracteres. Una llave indica el patron de reacomodo de los bits.

TEXTO	D (68)	E (69)	M (77)	O (79)
ASCII	100,0100	100,0101	100,1101	100,1111
AGRUPACION 5 BITS	10001,00	10001,01	10011,01	10011,11
LLAVE	45231	45231	45231	45231
TEXTO CIFRADO	01001,00	01001,01	01101,01	01101,11
AGRUPACION ASCII	010,0100	010,0101	011,0101	011,0111
TEXTO	4 (36)	% (57)	5 (54)	7 (55)

Tambien existen metodos donde la llave de transposicion es variable de acuerdo a ciertos parametros. Esto logra la informacion bajo un formato muy dificil de descifrar sin el conocimiento de la llave.

DECODIFICACION

La decodificacion es posible dependiendo de ciertos factores:

- Es un metodo conocido?
- Se conoce el lenguaje fuerte?
- Cuantas posibles codificaciones e presentaron?
- De cuanto texto codificado se dispone empleando la misma llave?
- Se dispone de cualquier texto simple que coincida?

Cuando el lenguaje fuente es conocido, pueden emplearse metodos estadisticos para el remplazo de las letras mas comunes, hasta lograr un mensaje con sentido. Tambien es importante recordar que muchas de las secuencias pseudoaleatorias con las que se forman las llaves cambiantes, dependen exclusivamente de tres factores por las que son facilmente reproducibles, los tres factores mencionados son:

- La condicion inicial
- El multiplicando y la cantidad de corrimiento
- La longitud de los registros del CPU.

Para evitar la decodificacion, se recomienda tener un numero alto de codigos de los datos, reducir la redundancia, reducir el numero de caracteres transmitidos, evitando espacios y signos de puntuacion para que el analisis sea mas dificil.

Generar llaves de longitud variable aumenta notablemente la complejidad de traduccion de la informacion encriptada, aun para descifradores con mucha experiencia.

Desventajas de la criptografia.

La traduccion de la informacion toma tiempo que puede ser vital en operaciones en tiempo real, por lo que hay que evaluar y hacer un compromiso del costo beneficio de proteger los datos contra el tiempo de respuesta.

3.4.3 INTEGRIDAD DE LA BASE DE DATOS

La integridad de una base de datos indica la ausencia de datos inconsistentes.
Mientras se realiza una actualizacion, la consistencia de la base de datos puede estar alterada, esto es posible notarlo en datos elementales que se hayan modificado aun cuando haya otros que no. Cuando el nivel de redundancia es bajo, se reducen las violaciones potenciales a la integridad, ya que un hecho aparecera rara vez en otro lugar. Hay archivos (archivos invertidos) que gracias a la redundancia permiten obtener eficiencia en el sistema, por otro lado, es claro notar que se debe establecer un compromiso entre la integridad y la redundancia para lograr el sistema optimo.

La competencia que realizan los procesos para obtener los resultados de sus transacciones con la base de datos, dependera fuertemente del sistema operativo, ya que este controlara todo flujo de informacion y seleccionara las partes importantes de la computadora para efectuar las operaciones necesarias, evitando los conflictos de actualizacion o consulta.

El empleo de seguros proporciona los medios par impedir que las transacciones interfieran unas con otras. Un problema serio que se presenta en el empleo de seguros, es el "aseguramiento mutuo" ya que genera un estado caotico, en donde solamente se reducen mediante la intervencion del sistema operativo, a esto se le conoce como "punto muerto".

3.4.3.1 Seguros

Consiste en proporcionar un mecanismo de seguro, que garantice la exclusion mutua de transacciones interferentes. Es decir, que el primero que solicite algun dato, sera el responsable de el, bloqueando a todas las demas solicitudes, hasta que lo libere.

El mecanismo de los seguros se logra mediante la utilizacion de semaforos que estan directamente relacionados con el recurso o dispositivo y nos indica en forma binaria, si esta ocupado o disponible. La transaccion que coloque al semaforo en un estado ocupado, se convierte en la responsable del recurso y solamente esta podra cambiar su estado nuevamente a desocupado. Al tamano del dato que queda asegurado por el semaforo, se le llama granularidad del seguro. El contar con granulos muy pequenos aumenta considerablemente el desempeno del sistema, ya que la posibilidad de interferencia sera menor, aunque hay que recordar que mientras mas pequenos sean los granulos, habra mas de estos y se complicara su manejo.

Dentro de una actualizacion se le conoce como lectura auditada, a aquella que requiera seguros y lectura libre a una consulta que no requiera de seguros.

Cuando en una consulta auditada, se solicita el mismo elemento por diferentes tareas, el semaforo se asigna a la primera y las demas se incorporan a una linea de espera, hasta que termine el uso por parte de la tarea que tiene asignado el seguro. Cuando el semaforo se libera, el manejador cronologico de transacciones puede reiniciar sus funciones, sacando la siguiente solicitud de la linea de espera debido a ese semaforo. Este manejador cronologico debe atender de inmediato la lista de semaforos para evitar un retraso excesivo, llamado hibernacion.

Para aquellos sistemas que por los metodos de seguros o para evitar la interferencia de las transacciones, permite realizar las operaciones hasta la conclusion de otras, se ha determinado que serian muy ineficientes, por lo tanto, se han logrado instrumentar funciones paralelas o que no requieran de la terminacion de otra para su ejecucion, a esto se le conoce como "seriabilidad", dicho de otra manera, es la propiedad de las transacciones que se traslapan de tener una programacion cronologica correspondiente en serie.

La interferencia en las rutas de acceso puede provocar que los programas de transacciones fallen. Trasladar un dato puede provocar la perdida de apuntadores, sustituyendolos con basura, cuando se buscaban apuntadores posteriores.

Para evitar localizar y recuperar datos incorrectos se coloca un letrero en el punto anterior. En muchos sistemas operativos solo se proporcionan seguros para archivos completos. Colocar un seguro implica no poder utilizar ninguna porcion de ese archivo aun cuando sea para una consulta sin modificaciones. Esto entorpece el tiempo de respuesta del sistema. Dado que tanto el tamano de la base de datos como su jerarquia no fueron posibles de determinar por anticipado, se recomienda que estos se manejen dinamicamente y recordando que por lo general seran muchos.

3.4.3.2 Hibernacion y punto muerto

Hibernacion: Cuando una transaccion esta bloqueada o dormida y el sistema esta demasiado ocupado para despertarla dentro de un lapso razonable.

Punto muerto: Cuando una transaccion esta bloqueada por otra y esta a su vez esta bloqueada por la primera. El circulo puede involucrar a muchas transacciones y ninguna de ellas puede ser activada sin posibles violaciones a la integridad.

Para el usuario estas dos fallas son iguales ya que el sistema no funciona.

La hibernacion ocurre cuando una transaccion no recibe recursos durante un periodo excesivamente largo y el usuario que presento la solicitud tiene razon para creer que el sistema no esta funcionando.

Puede crearse una cadena infinita de actualizaciones si ocurre una circularidad en las transacciones o en las restricciones de una base de datos. El tener transacciones con distintas prioridades es una invitacion a que las transacciones de baja prioridad entren en hibernacion.

Los puntos muertos ocurren cuando dos o mas transacciones solicitan recursos en forma incremental y se bloquean mutuamente impidiendose una a otra la conclusion.

Es posible que un punto muerto se cree cuando se realice el acceso a un solo objeto mediante dos transacciones y se aceptan solicitudes incrementales. Este tipo de punto muerto tiene mucho mayor probabilidad de ocurrencia si la granularidad es alta, es decir, el seguro cubre grandes porciones de informacion.

Tambien existen puntos muertos debidos a recursos compartidos del sistema. En el caso general de recursos compartidos, se consideran multiples dispositivos ya que el punto muerto puede verse provocado por una transacion que este bloqueada por un grupo finito de buffers, mientras que la otra transaccion esta ocupando unidades de cinta, etc., Las transacciones que se comunican a fin de procesar conjuntamente datos de proceso tambien estan sujetas a puntos muertos.

La posibilidad de que existan puntos muertos solamente es verdadera cuando se cumplan las cuatro condiciones siguientes:

1. Seguros. La interferencia de acceso se resuelve posicionando y respetando los seguros.

2. Bloqueo. El propietario de un objeto esta bloqueado cuando solicita un objeto asegurado.

3. Garantia de conclusion. Los objetos no pueden quitarse de sus propietarios.

4. Circularidad. Existe una secuencia circular de solicitudes.

Todas las tecnicas para manejar puntos muertos intentan modificar una de estas condiciones. Sin embargo, las tecnicas que resuelven puntos muertos tambien afectan a la hibernacion, por lo que la tecnica que se decida, debera someterse a pruebas de su efecto en el desempeno del sistema.

La capacidad para evitar puntos muertos puede simplificar muchas elecciones altenativas. Sin embargo, los esquemas para evitar estos casos imponen a los usuarios restricciones que pueden resultar dificiles de aceptar.

Para evitar los puntos muertos se deben contemplar las siguientes sugerencias:

1. Reparacion posterior. No utilizar seguros y arreglar despues las fallas por inconsistencia.

2. No bloquear. Solamente informar a quienes efectuaron solicitudes que provocaron conflictos de informacion.

3. Asignacion previa. Si existe algun conflicto, quitar los objetos a sus propietarios.

4. Secuencia previa. no permitir secuencias circulares de solicitud.

5. Aseguramiento de dos fases. Se realizan primero todas las solicitudes y si ninguna esta bloqueada se inician todas las modificaciones.

3.4.3.3 Mantenimiento de la integridad

La integridad continua de una base de datos es esencial para su operacion exitosa.

para lograr prevenir la interferencia de las actualizaciones, se propone la siguiente organizacion del sistema:

1. El sistema de proteccion del acceso. Permite que un solo usuario, el propietario de los datos, modifique un objeto protegido o posicione un seguro para el objeto.

2. El sistema de seguridad de acceso. Permite que solo exista un caso del usuario,

3. El sistema manejador de archivos. Asegura aquellos objetos que son sinonimos con los objetos identificados por el sistema de proteccion al acceso para su obtencion.

El problema general para asegurar la integridad de las bases de datos aun no se ha resuelto. Un paso inicial consiste en la definicion de las necesidades de aplicacion al no recolectar restricciones semanticas.

Ya que una falla de consistencia puede copiarse gradualmente a traves de toda la base de datos, es esencial llevar a cabo una estrategia de vigilancia periodica para que se conserven los datos un largo plazo. Si los resultados basados en datos almacenados durante muchos anos estan equivocados, la confianza del usuario en todo el trabajo realizado se pierde repentinamente. La vigilancia se puede realizar en dos niveles: estructural y orientado al contenido.

La vigilancia estructural la puede realizar el sistema de archivos sin participacion del usuario.

La vigilancia orientada al contenido requiere que el usuario proporcione afirmaciones referentes a los vinculos entre datos.

En ambos casos, la vigilancia solo se podra efectuar cuando exista cierta redundancia.

La vigilancia estructural se combina en forma conveniente en operaciones periodicas de realizacion de vaciados. Una de las areas que deben verificarse es la asignacion de almacenamiento. Cuando existen apuntadores, su logica puede verificarse siguiendo sus trayectorias, confirmando su adecuada operacion.

Vigilancia de contenido. Se realiza mediante procesos que permiten validar la informacion, considerando por ejemplo, los limites superior e inferior de algunas variables o contra tablas predefinidas.

3.4.4 SEGURIDAD Y PRIVACIA

Seguridad se refiere a la protecion de los datos contra perdidas accidentales o intencionales, que puedan causar personas no autorizadas o modificaciones y destrucciones no autorizadas.

Privacia se refiere a los derechos de los individuos y organizaciones para determinar por ellos mismos el cuando, como y que extension de su informacion puede transmitirse a los demas.

La privacia es algo que generalmente va mas alla de un centro de computo, llegando en ciertas ocasiones a sociedades donde queda establecida legalmente.

Es mas dificil mantener la privacia en un sistema de base de datos que en los archivos convencionales, ya que el medio ambiente que rodea a cada uno de estos sistemas es muy diferente y en el caso del sistema de base de datos, se hace dificil debido a que la informacion es compartida y se utiliza para diferentes aplicaciones, donde para algunas de ellas debemos ofrecer la

informacion integral, mientras que para otras es informacion restringida. En los antiguos sistemas de archivos convencionales, se definen publicos los archivos que asi lo requieren y protegidos los otros, y esto es posible, dada la desvinculacion de los datos, caso contradictorio en una base de datos.
Con esto no se pretende decir que en un sistema de bases de datos sea imposible definir niveles de seguridad y privacia, ya que por lo general estos manejadores cuentan con las rutinas adecuadas para ofrecer ciertas restricciones de informacion inclusive al nivel de un dato individual.
La seguridad en un sistema de bases de datos debe contemplarse cuidadosamente en el diseno de las estructuras internas de la informacion, para que se contemplen todas las alternativas y se obtenga la mejor solucion, por lo general es una tarea que desarrolla el analista y administrador de la base de datos.

Los siete puntos esenciales para la seguridad de una base de datos son:

Una base de datos debe ser

1. PROTEGIDA
2. RECONSTRUIBLE
3. AUDITABLE
4. A PRUEBA DE TRAMPAS

Sus usuarios deben ser

5. IDENTIFICADOS

Sus transacciones deben ser

6. AUTORIZADAS
7. MONITOREADAS

Al hablar de Seguridad de la informacion, es posible establecer niveles para diferentes contextos de esta, los cuales se resumen en la figura 3.5.

Cada uno de estos niveles puede incluir alarmas, puertas, guardias u otros mecanismos para evitar los accesos no autorizados, precauciones a fuego, protecciones a los archivos, etc. No es suficiente generar los respaldos de la informacion, ya que estos pueden ser robados o destruidos por fuego.

En algunas configuraciones de proteccion y respaldo de la informacion, solamente se duplican porciones de la base de datos para que si existe alguna falla, esta pueda regeneranse a partir de la informacion minima, la causa de guardar solamente porciones y no toda la base de datos, es optimizar el tiempo de duplicacion de archivos y poder ofrecer un sistema que sea capaz de procesar en tiempo real, sin menospreciar la importancia de la proteccion de la informacion. Esto se muestra en la figura 3.6.

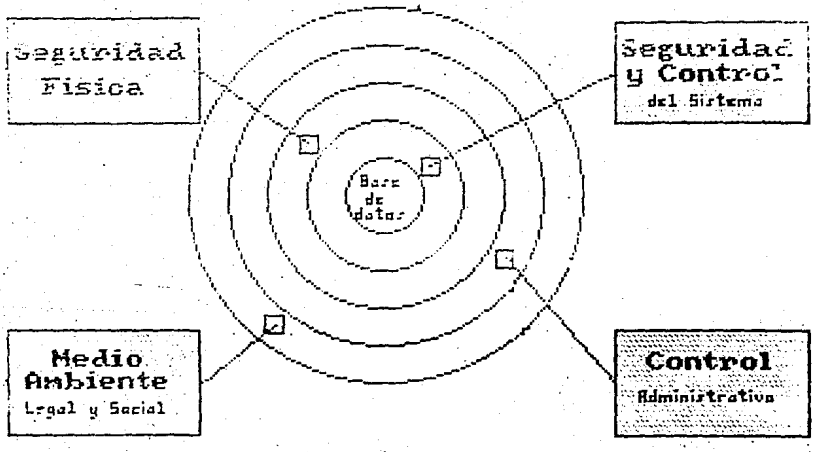


Figura 3.5 Seguridad de la información

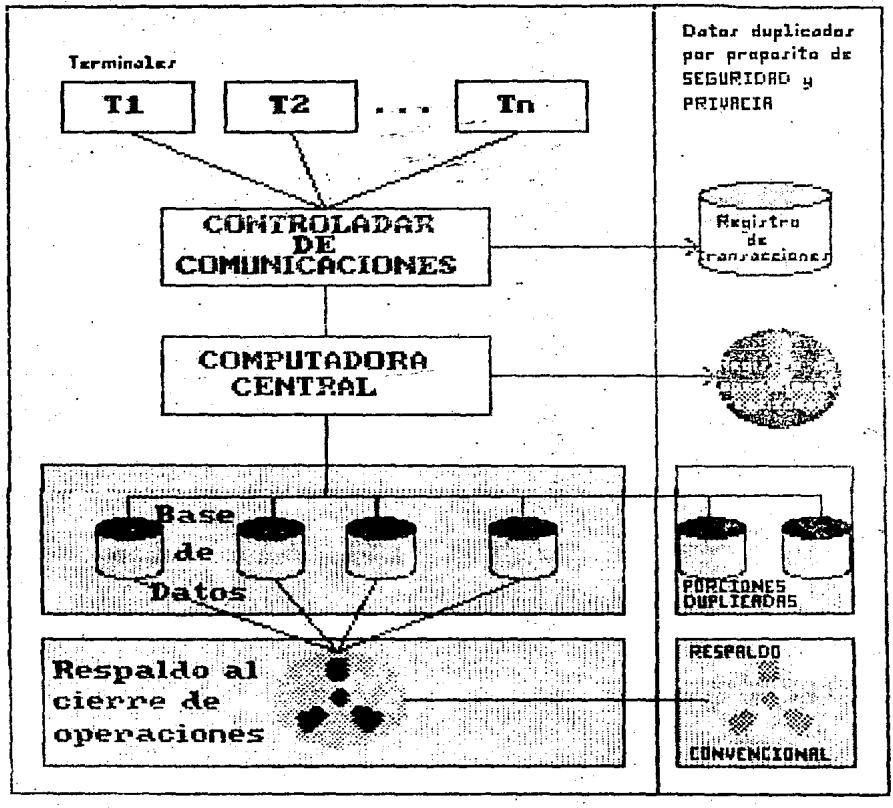


Figura 3.6 Protección de la información

3.5 CODIFICACION

El contenido de una base de datos es una representacion del mundo real. Para lograr una representacion, primero se hace una abstraccion de la realidad, despues esta se traduce a numeros o caracteres que a su vez son transformados a códigos binarios de computadora. Las decisiones que se tomen en el nivel de representación ligarán la base de datos con un contenido especifico durante muchos anos de su operación; por lo tanto, se debe estar consciente de las alternativas de representación, de manera que el sistema apoye la diversidad y solo restrinja lo irrealizable.

3.5.1 Representacion del conocimiento

La capacidad de la base de datos para representar el mundo real se reduce al considerar solamente el conocimiento disponible, que a su vez es analizado para considerar solo aquellos aspectos que seran mas utiles en una decision para reconstruir la realidad. En el diseno de bases de datos la categorización se basa en la tradición y la disponibilidad de los datos. Los datos pueden recolectarse en forma detallada o restringida. La recolección de datos detallada aumenta el costo de recolección, almacenamiento y procesamiento. El beneficio de tener datos muy detallados resulta dificil de cuantificar.

El resultado de una decisión taxonómica requiere que se utilice documentación en la captura o entrada de los datos, de manera que se limite el rango de respuestas, como por ejemplo: "el número de hijos se refiere a los hijos de cualquiera de los padres que vivan en la misma casa y que sean económicamente dependientes"

Si no se hicieran este tipo de especificaciones, se registraría información diferente.

CLASIFICACION

Una vez que se han seleccionado las entidades y sus atributos, entonces se clasifican las posibilidades de observación, las cuales pueden ser tan simples como contestar a: "Que opción desea: 0) Salir, 1) Introducir datos, 2) Corregir, 3) Calcular ?" ,o pueden no ser tan obvias como lo seria el tratar de diagnosticar una enfermedad, para lo cual se requiere tomar otras decisiones. En la tabla 3.A se presenta una lista de posibilidades de clasificación.

CLASIFICACION DE VALORES DE ATRIBUTOS

Capacidad de clasificacion	Capacidad de medicion	Elecciones de valor	Dominio de la muestra
Con rangos	Métrica	Continuas	Peso
		Enteras	número-almacenado
	Ordinal	-	amistad
Sin rangos	Nominal	Ilimitadas	nombre
		Limitadas	grupo de frutas

TABLA 3.A

La clasificación de un tipo de observación no siempre es obvia; sin embargo, una vez que un dato se ha clasificado se establecen aspectos importantes del dominio para los atributos que permitan seleccionar una representación adecuada.
Los valores que puede tomar este elemento tienen un orden natural para establecer rangos, que tengan una secuencia que se explotará en la codificación y en el procesamiento de datos.
En caso de no existir un orden natural en las observaciones, se puede establecer un orden artificial.

Los dominios pueden especificarse mediante una lista como la siguiente: frutas(manzana, pera, mango, platano) y el orden que tiene cada uno de ellos dentro de la lista define el rango. Sin embargo, no es conveniente plantear consultas con operaciones de relación (mayor que o menor que), a menos que la definición del orden o secuencia de clasificación resulte clara para el usuario.

Si los valores son métricos, los valores se determinan por medición y es posible efecturar operaciones aritméticas con ellos.

Los valores dentro de un dominio peso pueden sumarse, restarse, compararse, etc.
Cuando los datos son contínuos, es necesario adoptar decisiones acerca del intervalo y la precisión debido a que el hardware de la computadora impone límites, los cuales no parecen importantes, sino hasta el momento de hacer cálculos subsecuentes a través de los cuales se van perdiendo exactitud.

Para los datos ordinales o nominales se deberán restringir las operaciones aritméticas, tomando como ejemplo la medición de la inteligencia (IQ), en donde se obtiene un promedio mediante el cual es posible asignar porcentiles significativos a los individuos. La siguiente proposición

"Julio es el doble de inteligente que Pedro"

no puede basarse en esa métrica artificial. La moda y la mediana se pueden calcular tanto en datos ordinales como en datos enteros, ya que se pueden obtener por comparación y conteo.

Cuando los datos son estrictamente nominales también es posible determinar la moda, por ejemplo el apellido "González" puede ser el apellido más frecuente en Toluca, pero ningún rango esta implicado por el valor "González". El color de ojos de Sergio no puede considerarse mayor o menor que el de Ramón a menos que se introduzca algún tipo de rango como claro comparado con oscuro.

Los valores nominales que son ilimitados permiten el manejo de texto. Dentro de este tipo se encuentran los nombres de personas, nombres de instituciones, etc., Resultando difícil controlar la exactitud y los valores solo pueden utilizarse por comparación. Los valores nominales no siempre pueden evitarse, pero si es posible establecer un dominio limitado, esto permitirá tener muchas posibilidades para verificación y control de los datos. La lista de valores puede llegar a ser tan grande que para definir el dominio se necesite de un léxico. Algunas veces una definición puede estar disponible como parte de una convención de codificación.

CODIFICACION

Después de que los dominios de los valores que se van a utilizar se han clasificado, se procede a seleccionar la representación real de los datos en la computadora.

Para los valores que tienen una clasificación métrica la elección es directa. Los datos continuos pueden representarse mediante enteros o fracciones binarias si lo permite el intervalo de valores. La posición de un punto binario o decimal puede hacerse de diferentes maneras. Una de ellas consiste en emplear una notación de punto flotante, o la representación de valores en forma de fracciones racionales, empleando dos enteros para el dividendo y el divisor. La representación adecuada de valors indefinidos o desconocidos puede resultar difícil. En sistemas que utilizan una representación de signo-magnitud para valores numéricos, se ha utilizado el valor "-0", pero esta convención es recomendable sólo cuando la generación de valores numéricos sea controlable por el diseñador del sistema recuperador.

Con frecuencia los datos ordinales se representan como enteros, requiriendose de un léxico interno o externo al sistema para la traducción. En caso de no traducir los datos ordinales a códigos enteros o alfabéticos clasificables se requiere de un léxico para realizar las operaciones de comparación.

Los datos nominales de un dominio limitado por lo regular se asocian con un léxico para verificar la captura de los datos. Los datos pueden ser representados en forma de enteros para facilitar el manejo y almacenamiento. Cuando se necesita producir el valor como resultado, se requiere de hacer la conversión inversa; es decir convertir la representación de enteros a datos nominales; esto se hace utilizando el mismo léxico pero en sentido contrario.

Cuando los valores codificados llevan dígitos de verificación o son redundantes, es posible detectar algunos errores internos de procesamiento, mediante cálculos que generan dígitos de verificación que se consultan en el léxico. Cuando estos dígitos de verificación no se encuentran en el léxico se detecta el error.

Los valores nominales ilimitados sólo pueden traducirse a un código adecuado de caracteres. Debido a que los códigos nominales ilimitados casi nunca intervienen en el procesamiento. Algunos sistemas han optado por reemplazar los valores nominales por códigos de referencia, almacenando las cadenas de caracteres en un archivo remoto. Teniéndose acceso a este archivo sólo durante los procedimientos de salida.

3.5.2 Representacion de maquina

Para todas las categorías, con excepción de los datos nominales ilimitados, un elemento dato puede codificarse como un número entero o de punto flotante. Generalmente estos elementos dato son de tamaño fijo. Cuando los datos permanecen en forma de cadena de caracteres, el dato se representa mediante una secuencia de caracteres binarios codificados, generalmente de longitud variable.

La codificación estándar para caracteres conocida como tabla ASCII (American Standard Code for Information Interchange), que es un código estándar americano para intercambio de información, en este código, cada caracter se representa mediante siete bits; en máquinas con espacio para caracteres de 8 bits se desperdicia un bit. En la tabla 3.B se muestran los códigos de caracteres alternativos y su tamaño.

Nombre Común	(organización)	Tamaño	Número de símbolos Datos	Control	Indefinido
Baudot	(CCITT)	5 bits+ Corrimiento	50	5	3+
BCD	(CDC)	6 bits	64 o 48	0	0 o 16
Fidelata	(US Army)	6 bits	48	16	0
ASCII	(ANSI)	7 bits	95	33	0
EBCDIC	(IBM)	8 bits	133	64	59

ANSI.- American National Standards Institute
ISO .- International Standards Organization

Para uso internacional algunos caracteres pueden representar símbolos alternos.

TABLA 3.B

En la comunicacion de datos se utilizan caracteres de control para denotar espacios vacios, separar unidades de datos, controlar las funciones de los dispositivos de salida, tales como alimentacion de linea, cambio de linea, etc. Los caracteres de control en las bases de datos resultan de gran utilidad para delimitar los campos de longitud variable y representar "caracteres indefinidos", este tipo de uso requiere de mucho cuidado para evitar conflictos con funciones asignadas por hardware.

CONVERSION

Despues de que se capturan los datos, eston son convertidos a su representación binaria. Este proceso de conversion es costoso. Las técnicas de captura de datos pueden llegar a influir en la elección de representación de los datos. Tiene especial importancia el manejo de ments, en donde el código de la opción elegida se introduce en el sistema de manera que si la presentación del menú cambia con el tiempo, el código no represente la posición del letrero, sino que se haya traducido a un código que pueda conservarse.

Estas formas de autocodificación o cuestionarios de elección múltiple mejor conocidos como ments son utilizables en cualquier campo de aplicación.

Es posible generar formas prototipo utilizando datos de la definición del esquema (nombres y longitudes de cada campo), asegurando con ello que los datos capturados coincidan con las especificaciones del archivo.

CODIGOS ESTANDAR

Con frecuencia los datos nominales se representan mediante códigos enteros, con lo que se ahorra gran cantidad de tiempo en el diseno de la captura, en espacio de almacenamiento y en el diseno del sistema. Además simplifica la integración de la base de datos con otras áreas. Cuando los datos nominales van a ser utilizados por un grupo de usuarios grande, se tiene que llegar a un acuerdo con respecto al código que se les va a asignar, a fin de que esta codificación sea completa y no exista ambiguedad ni redundancia. Esta tarea puede requerir de mucho tiempo.

La definición de codificaciones estandares ha sido encabezada por grupos en el Bell System, organizaciones comerciales, agencias de la defensa, agencias federales y el Bureau de Census. También se están desarrollando algunos estándares internacionales.
La mayoría de los códigos estándares están disponibles en cinta para facilitar su incorporación a sistemas procesadores de datos. En la tabla 3.C se presentan algunas referencias a tablas estándar de código.

Area tema: Entidades nombre del codigo	Organizacion Lugar	Numero de anotaciones: Formato del
Estados de E.U. Identificador USPS	National Bureau of Standards FIPS 5-1 Gaitherburg, MD	56:AA o :NN
Paises	NBS, FIPS 9	
Codigos postales	POs britanicos y canadienses	:ANA-NAN
Ocupaciones	Bureau of the Census	:XXXX
Automoviles	Consumers Union	Todos los automoviles desde 1970
Publicaciones MARC	Washington, DC	(3 estandares 100 elementos)
Negocios D-U-N-S	Dun and Bradstreet, Inc. New York, NY	3000000 :NNNNNNNNN

TABLA 3.C Muestrario de codigos estandar

CODIFICACION DE NOMBRES DE INDIVIDUOS

La codificacio de nombres es uno de los problemas principales en el procesamiento de datos, debido a que el empleo de un numero que represente a cada individuo va a estar limitada por el tipo de proceso, por ejemplo la representacion de personas mediante su RFC esta restringido a los procesos relacionados con el pago de impuestos, nomina, etc.

El empleo de nombres se dificulta por dos razones:

1) Conglomerados.- El problema que surge al emplear nombres y apellidos es que algunos aparecen con gran frecuencia. Los conglomerados provocan la recuperacion de demasiados nombres.

2) Inconsistencia en la forma de escribir nombres y apellidos.- Dado que la forma de escribir un nombre o un apellido no siempre es igual, esto provoca fallas al recuperar la informacion.

Es posible emplear referencias avanzadas para localizar simultaneamente los nombres de sonido semejante. A continuacion se define el metodo Soundex.

Se eliminan todos los caracteres no alfabeticos (*,-,(,etc.)
Se convierte todo a mayusculas
La primera letra se traslada al termino resultado
El resto se convierte de acuerdo con los siguientes pasos:
- Se eliminan las consonantes H y W que con frecuencia no se pronuncian.
- Se hacen las siguientes sustituciones:

Labiales	: B,F,P,V	- 1
Guturales, sibilantes	: C,G,J,K,Q,S,X,Z	- 2
Dentales	: D,T	- 3
Largoliquidas	: L	- 4
Nasales	: M,N	- 5
Cartoliquidas	: R	- 6

- Se combinan dos o mas digitos adyascentes
  LL - 4, SC - 2, MN - 5
- Las letras restantes (vocales: A,E,I,O,U e Y) se eliminan.
- Los primeros tres digitos se encadenan con el resultado.

Este metodo resuelve gran parte del problema de la inconsistencia en la escritura, ya que lo que realmente maneja es una representacion fonetica de las palabras. Por ejemplo:

McCloud,MacCloud,McLoud,McLeod,M'Cloud	M243,M243,M243,M243,M243
Ng, Eng, Ing	N2,E52,I52
Rogers, Rodgers	R262,R326
Smith, Schmid, Smid, Smyth, Schmidt	S53,S53,S53,S53,S53
Wiederhold, Weiderhold, Widderholt	W364,W364,W364

Este esquema parece estar sesgado hacia los nombres anglosajones, aunque tambien existen otros codigos.

El empleo de codigos Soundex como llave primaria aumenta el numero de colisiones. Este codigo proporciona hasta 26 X 7 X 7 X 7 o 6,734 posibilidades, y las alternativas no se distribuyen uniformemente. La primera letra provoca una mala distribucion, como se muestra en la tabla 3D. Por lo tanto, cuando resulta necesario conocer la identidad de un individuo tal vez se requiera informacion secundaria como direccion, fecha de nacimiento, profesion, lugar de nacimiento; nombres de los padres, etc. El uso de los parametros adecuados dependera de las circunstancias.

Resulta muy costoso que un individuo cambie su nombre,dado que con ello se complica su identificacion.

La preocupacion acerca de la proteccion de la privacia en todos los aspectos, impide el empleo de numeros personales de identificacion; por lo que, algunos paises europeos estan descontinuando su empleo. El uso de numeros como identificadores no va a desaparecer. Los individuos tendran muchos numeros como identificadores: Numero de licencia, numero del seguro social, numero de empleado, numero de cuenta bancaria, etcetera.

3.5.3 Compresion de los datos.

El almacenamiento de los datos sigue siendo uno de los factores primordiales del costo. Con frecuencia la representacion de los datosproduce desperdicio de espacio y redundancia. Aun cuando cierta redundancia, resulta de gran importancia para mejorar la compatibilidad, es posible hacer una compresion de los datos, siempre que se presente cualquiera de las condiciones siguientes:

Solo se consideran esquemas que permitan una reconstruccion completa de los datos, para que los datos comprimidos se puedan volver a expandir.

Existe cierto intercambio entre el grado de compresion y el costo del tiempo de procesamiento, ya que la compresion reduce las condiciones de transferencia de datos y con ello se incrementa la rapidez de las operaciones de archivo. Por lo tanto debe hacerse una comparacion entre el tiempo que utiliza la unidad central de proceso (CPU) en hacer la compresion y la velocidad de acceso al archivo; asi como el costo de almacenamiento.

Compresion de elementos numericos.

Para fines de calculo es conveniente que los valores numericos ocupen palabras completas, ya que frecuentemente los datos forman matrices para permitir que los calculos se hagan rapidamente a traves de renglones y columnas. Debido a que los datos son representaciones de fenomenos del mundo real, no resulta raro asignar posiciones para valores que aun no se han recolectado o

que tengan un valor de cero. El almacenamiento de valores de datos diferentes de cero con sus coordenadas, dejando comprimidos los valores cero fuera del archivo, por ejemplo, se tiene un registro con longitud = 20 espacios de datos numericos

registro original:

0,2,0,0,0,0,5,7,8,0,0,0,3,0,0,0,1,0,0,4, y

Representacion coordenada:

2:2,7:5,8:7,9:8,13:3, 16:1,19:4420:u

Ahora el registro requiere solamente de 8 espacios de datos.

Cuando ocurren tanto valores cero como valores indefinidos, el vector bit puede utilizarse para describir las 3 siguientes posibilidades:

0 valor cero
10 valor indefinido
11 valor verdadero distinto de cero y que se va a almacenar. El vector bit para el mismo registro, sin considerar a "u" como dato.

01100001111110001100110011102. 2,5,7,8,3,1,4,4,u

En este caso no hubo beneficio, donde realmente se ve su efecto es donde abundan los valores indefinidos. El tamano de este ultimo vector bit depende de los valores en el registro. Para evitar esto y tener con vector bit de tamano fijo se pueden codificar 4 posibilidades
00 valor cero
01 valor repetido
10 valor indefinido
11 valor almacenado

quedando el vector bit de tamano fijo como"

0011000000001111110000001100001100000l1 102:2,5,7,8,3,1,44,u

La conversion de valores binarios a digitos decimales de tamano fijo no tiene sentido, ya que resulta mejor analizar los registros para determinar el mayor tamano de espacio, el cual sera fijo y podra conservarse como otro prefijo en el registro. utilizando los mismos datos se encuentra que el dato de mayor valor puede representarse mediante (9) bits, quedando con base en los valores que se van a almacenar, para lo cual se puede hacer un rastreo del registro, para determinar el mayor espacio necesario, conservando este tamano como otro prefijo en el registro.

0011000000... 11000011102 9,002/005/007/008/003/001/4/4

El simbolo / es un indicador de las fronteras de las unidades de 9 bits [En este caso se necesita 1 palabra 36 bits en vez de 4 palabras). El metodo fracasa si existe un solo valor grande en el registro.

Una alternativa para compresion puede ser un esquema que especifique el intervalo de valores. Con el proposito de que el registro pueda formatearse de acuerdo con el intervalo. Por ejemplo si el intervalo de valores es de 0 a 20, almacenando estos valores en campos de tamano $[\log_2 20]=5$ bits cada uno.

Las posiciones inferiores de los numeros de punto flotante pueden truncarse si no se requiere de mucha precision. Esta decision tambien depende de las especificaciones proporcionadas en el esquema.

Cuando los datos representan atributos que cambian lentamente, puede convenir almacenar las diferencias de valores sucesivos.

Por ejemplo, se tiene la secuencia
502,503,505,504,507,508,511
puede representarse como:
502,4,+1/+2/-1/+3/+1/+3

Con este metodo de compresion el ejemplo anterior solo requiere de 4 bits para almacenar las diferencias sucesivas.

Compresion de cadenas caracter

La compresion de cadenas caracter es mas frecuente que la compresion de valores numericos. Las abreviaturas se utilizan con frecuencia en forma independiente del procesamiento de los datos. El manejo de abreviaturas presenta un grado de compresion de los datos previo a su captura. Las abreviaturas pueden reducir el volumen de captura de datos; asi como los costos de almacenamiento. Cuando se desarrollan abreviaturas especificas para trabajar con la base de datos, es necesario considerar los efectos que se pueden generar.

Ademas debe tomarse en cuenta que la compresion de terminos dentro de la computadora puede resultar mas economica que el uso de abreviaturas que simplifican el proceso de captura pero no hay que olvidar que una reduccion de errores en el procesamiento manual resulta de mucha utilidad.

Recodificacion

La tecnica mas utilizada para comprimir datos textuales es la representacion de los datos en un conjunto de caracteres de menor tamano, por ejemplo, los datos de un conjunto de 8 bits pueden recodificarse a un conjunto de 7 0 6 bits.

Con 6 bits se pueden representar los digitos.

La compresion al recodificar simbolos, en un numero especifico de bits sera optima si el numero de simbolos a representar, es menor que una potencia de dos.

La recodificacion de grupos de unos cuantos caracteres y una sola representacion puede producir una compresion mas densa, como se muestra en la tabla 3.E. El limite para un grupo es 2 bits.

El ahorro se calcula en base a una codificacion no agrupada y densa.

El numero de bits para los grupos puede no coincidir con las fronteras de palabra,de manera que unidades de palabras multiples tendran que manejarse a fin de explotar este tipo de compresion. De manera que el agrupamiento de 5 caracteres alfanumericos permite que una palabra de 32 bits represente 5 caracteres con un ahorro del 20%, con respecto a la representacion comun del codigo de 8 bits/caracter.

Eliminacion de espacios en blanco

En datos formateados existe la tendencia de incluir largas secuencias de espacios en blanco. Una tecnica sencilla para comprimir dichas secuencias consiste en reemplazar una cadena de i espacios en blanco mediante una secuencia de dos caracteres ' i' como se muestra a continuacion.

Edgar	3	523.12	Edgar 3523.12
Maritza	1	012.32	Maritza 112.32
Sergio	2	1.28	Sergio 21.28
Jose	4	432.12	Jose 4432.12

El valor de i esta limitado al mayor entero que pueda conservarse en un espacio caracter (63,127 o 255).

CODIFICACION DE SIMBOLOS DE LONGITUD VARIABLE.

Los caracteres no ocurren con la misma frecuencia a lo largo del texto, como puede observarse en la tabla 3.F.

La tecnica de codificacion Huffman aprovecha esta caracteristica para comprimir datos. A los simbolos mas frecuentes se les asignan los codigos mas cortos y todos los codigos largos se contruyen de manera que los codigos cortos no aparezcan como secuencias iniciales de bits de los codigos largos. No se requiere de otros bits marca para denotar la separacion de un campo caracter y su sucesor.

Con los datos de la tabla anterior, se empieza a codificar el espacio optimo, como se muestra en la tabla 3.G.

Obteniendose los codigos, como se muestra en la tabla 3.H. Generalmente la construccion de arbol puede utilizarse para minimizar el acceso a elementos de frecuencia desigual, siempre y

A	3.051	15	12.111	2	6.229	4
B	9.357	3	4.129	9	5.550	7
C	7.267	5	3.916	10	9.722	2
D	4.783	10	2.815	13	6.016	5
E	1.883	17	1.838	18	4.386	11
F	3.622	13	3.911	11	5.162	9
G	5.103	8	1.960	16	3.086	16
H	7.440	4	6.937	5	3.842	12
I	0.387	23	8.061	3	3.707	13
J	2.954	16	0.427	23	0.776	21
K	3.938	12	0.576	21	0.602	22
L	4.664	11	2.746	14	3.474	14.5
M	9.448	2	4.429	8	4.560	10
N	1.785	18	2.114	15	1.844	18
O	1.436	19	6.183	7	2.271	17
P	4.887	9	2.897	12	7.801	3
Q	0.175	25	0.199	24	0.427	23
R	5.257	7	1.880	17	5.317	8
S	10.194	1	6.787	6	12.886	1
T	3.450	14	15.208	1	5.608	6
U	0.238	24	1.008	20	1.417	20
V	1.279	20	0.428	22	1.436	19
W	6.287	6	7.643	4	3.474	14.5
X	0.003	26	<0.001	26	<0.001	26
Y	0.555	21	1.794	19	0.369	24
Z	0.552	22	0.002	25	0.039	25

TABLA 3.D Distribucion de nombres, palabras y tipos de palabras de acuerdo con las letras iniciales

Conjunto de caracteres	Tamano del gpo.	Grupo	Amplitud del gpo.	Bits	Limite	%Ahorro
Digitos	10	3 digitos	1 000	10	1 024	16.6
Alfabeto de un solo tipo	26	4 letras	456 976	19	524 288	5.0
Alfanumericos+28	90	2 caracteres	8 100	13	8 192	7.1
Alfanumericos+18	80	3 caracteres	512 000	19	524 288	9.5
Alfanumericos+22	84	5 caracteres	4.29X10	32	4.29X10	9.25

TABLA 3.E Codigos alternativos para grupos de caracteres

E	133	D	43	G	14
T	93	L	38	B	13
O	85	C	31	V	10
A	81	F	29	K	5
N	75	U	28	X	3
I	71	M	27	J	2
R	70	P	22	Q	2
S	65	Y	15	Z	1
H	61	W	15	Total	1032

TABLA 3.F Frecuencia relativa de las letras en textos en ingles

1 Se inicializa una lista con una anotacion para cada simbolo, su frecuencia y un espacio para su codigo.

2 Se toman las dos anotaciones de la lista que tengan la menor frecuencia; se les asignan los bits 0 y 1.

3 Si una de las anotaciones es el resultado de una combinacion previa, entonces el nuevo bit se encadena al frente de cada codigo, de lo contrario se inicializa el campo de codigo con el bit.

4 Se quitan de la lista las dos anotaciones empleadas y se inserta una sola anotacion combinada que tiene como frecuencia la suma de las dos anotaciones y se ligan las dos anotaciones empleadas.

5 Se repiten los pasos 2,3 y 4 con las dos anotaciones que ahora tienen las frecuencias mas bajas, hasta que se hayan procesado todos los simbolos.

6 Los valores del codigo pueden obtenerse a partir del arbol de anotaciones.

a)

Elemento siguiente	Pasos de puesta en codigo		Frecuencia combinada	Nuevo codigo para elementos previamente puestos en codigo
fr(Z)=1				
fr(Q)=2	Z->0	Q->1	fr(ZQ)=3	
fr(J)=2	J->0	ZQ->1	fr(JZQ)=5	Z->10 Q->11
fr(X)=3	X->0	JZQ->1	fr(XJZQ)=8	J->10 Z->110 Q->111
fr(K)=5	K->0	XJZQ->1	fr(KJZQ)=13	X->10 J->110 ... Q->1111
fr(V)=10	V->0	KXJZQ->1	fr(VKXJZQ)=23	K->10 ... Z->11110 Q->11111
fr(B)=13				
fr(G)=14	B->0	G->1	fr(BG)=27	
fr(W)=15				
fr(Y)=15	W->0	Y->1	fr(WY)=30	
fr(P)=22	P->0	VKXJZQ->1	fr(PVKXJZQ)=45	V->10 ... Z->111110 Q->111111
fr(M)=27	M->0	BG->1	fr(MBG)=54	B->10 G->11
etc.				

b)

TABLA 3.G a) Algoritmo para construir un codigo Huffman
b) Construccion de un codigo Huffman

E	100	D	11011	G	001111
T	000	L	11010	B	001110
O	1111	C	01110	V	001010
A	1110	F	01011	K	0010110
N	1100	U	01010	X	00101110
I	1011	M	00110	J	001011110
R	1010	P	00100	Q	0010111111
S	0110	Y	011111	Z	0010111110
H	0110	W	011110		

TABLA 3.H Codigo Huffman para texto en ingles

cuando el costo de acceso por enlace sea igual.

Con las frecuencias de ocurrencia fr(i) mostradas en la tabla 3.F la longitud promedio de caracter se calcula como:

$$lc = \frac{fr(i)\,lon(i)}{fr(i)} = 4.1754 \text{ bits}$$

donde: lon(i) es el numero de bits de simbolo i en el codigo Huffman.

Manejo de cadenas de longitud variable.

Los nombres de personas, organizaciones, libros, etc. varian mucho en longitud,de tal forma que si se empleara una longitud fija de cadena, dificilmente se lograria acomodar sin tener que buscar reemplazo.

En la mayoria de las maquinas el uso de los codigos Huffman es complicado: los esquemas para utilizar caracteres libres que remplacen cadenas largas que se presentan con frecuencia. Los caracteres libres pueden resultar faciles y efectivos. De manera que si se desea repetir muchos caracteres como espacios en blanco, es conveniente remplazar dichos caracteres por caracteres libres.

Pueden ser aquellos caracteres tanto de control como de datos no utilizados.

O cuando las palabras aparecen con cierta frecuencia; tambien pueden reemplazarse mediante caracteres libres unicos. Tal es el caso de palabras como: y, o, que, la, el, son, a, con,.... etc.

REEMPLAZO DINAMICO

Cuando se dispone de tiempo de procesador durante la captura de datos, es posible seleccionar un lexico que controle el reemplazo en forma dinamica.

El lexico que sea especifico para un registro tendra que llevarse junto con dicho registro. Por ejemplo:

"Lanina habla con ninas que no hablan" (37 caracteres)
(1) nina (2) habla

La (1)(2) con (1)s que no (2)n
(1) y (2) son caracteres de control

La compresion puede ser del 40% al 60% en cadenas que varian de 400 a 1500 caracteres utilizando hasta 10 patrones en el lexico. Esta misma tecnica es aplicable a cadenas de bits arbitrarias.

Tambien podria generarse dinamicamente un codigo Huffman, pero el

arbol de codigo tendra que agregarse al registro, resultando dificil representar el arbol en forma compacta. Ademas, un programa general de decodificacion para codigos variables de caracter tambien seria mayor y mas lento que un programa orientado a una codificacion Huffman especifica.

IMPLANTACION DE LA COMPRESION

Los esquemas de compresion pueden evaluarse mejor cuando se aplican a una muestra representativa de los datos.

Los parametros a evaluar son:

- Reduccion de las necesidades de almacenamiento
- Modificaciones en el tiempo de transferencia de datos
- Costo de procesamiento de la compresion y expansion.

Debido a que es posible utilizar metodos combinados, habra que considerarse el orden en que se apliquen los algoritmos de compresion.

Punto de procesamiento para la compresion.

La compresion o expansion puede realizarse en diferentes puntos del proceso de adquisicion, almacenamiento y recuperacion de datos. Cuando la compresion se lleva a cabo inmediatamente despues de la captura de los datos y si la expacion se realiza antes de presentar la informacion, el volumen de datos que maneje el sistema sera menor. Los datos que solo sirven para seleccionar otros elementos no tendran que expandirse, ahorrandose con ello tiempo de procesamiento por la expansion. El uso generalizado de datos comprimidos requerira que todas las rutinas de procesamiento puedan manejar datos comprimidos. Cuando se requiere expander varias veces los datos para hacer calculos resulta conveniente utilizar elementos de tamano consistente en vez de trabajar con elementos comprimidos.

Es posible equipar a los sistemas de archivo con algoritmos de compresion mediante el empleo de procedimientos de base de datos o mediante otros procedimientos, de manera que el usuario no se de cuenta para nada de la compresion y la expansion, o puede ser que un administrador de base de datos ajuste el grado de compresion realizado de acuerdo con las razones entre el almacenamiento y el costo de CPU aplicables a un sistema. El manejo automatico de la compresion del sistema puede disminuir la relacion entre el usuario y el hardware y reducir la asignacion de los programas a caracteristicas del sistema.

COMPRESION Y DISENO DE ARCHIVO.

El hecho de que los archivos comprimidos generen registros de longitud variable y que la actualizacion de registros de archivos comprimidos haga que los registros cambien de tamano, muestra claramente que la compresion de los datos tiene un efecto importante en el diseno del archivo.

CAPITULO 4

Técnicas Estadísticas para el Diseño de Sistemas de Información

4.1 TECNICAS ESTADISTICAS.

A menudo, el proceso de analisis y diseno puede utilizar informacion recolectada a partir de la observacion de sistemas relacionados. La estadistica es un medio para condensar y simplificar los datos detallados. Los resultados que se obtienen permiten transmitir la experiencia a sistemas mejorados o nuevos.

A fin de aplicar las tecnicas estadisticas, se requieren algunas actividades en secuencia.

1) Es necesario comprender el proceso que se este analizando.
2) Se listaran los parametros que afectan a la operacion.
3) Se recolectaran datos acerca de la operacion, y se exhibiran con los parametros empleados.
4) Se desarrollaran funciones de transferencia para permitir la aplicacion de la informacion recolectada en sistemas nuevos.
5) La validez de la transferencia se probara en la nueva aplicacion.

Se cubriran sin demasiada profundidad estos puntos utilizando ejemplos que provengan de aplicaciones de bases de datos. El conocimiento de la estadistica se encontrara en los textos especializados.

Modelos y parametros. - A menudo resulta dificil medir los parametros operativos basicos y es frecuente que las medidas de desempeno obtenidas representen los efectos combinados de muchos parametros. La fuerza del empleo de tecnicas estadisticas hace posible analizar observaciones de fenomenos complejos y aleatorios a partir del modelo base. Los resultados obtenidos pueden utilizarse para el desarrollo posterior del sistema.

4.1.1 Distribuciones de demanda comunes

Estas evaluaciones exigen que se tome en cuenta que los registros promedio y los archivos promedio son tan raros en la realidad como las personas promedio. En la mayoria de los casos en que se utilizaron promedios, las medidas obtenidas eran mas o menos fuertes; es decir, no se veian muy afectadas por cierta cantidad de variacion con respecto al valor promedio. Sin embargo, a menos que se encuentren rigidamente restringidos, los valores observados de elementos tales como la longitud de registro, la longitud de campo y la frecuencia de consulta se desviaran de la media.

Con frecuencia, las mediciones obtenidas para datos variantes se presentan con exito mediante tecnicas graficas adecuadas para la exploracion de datos. Los datos presentados en forma de programa pueden mostrar las frecuencias de los eventos clasificados por tipo de valor. Se dice que el histograma representa una distribucion de ocurrencias. En esta descripcion todos los valores de las ordenadas se consideran frecuencias, y

los valores localizados en las abscisas, son valores categorizados de los eventos. Si no existe variacion, todos los eventos caeran en la misma columna del histograma y se tendra una distribucion constante.

Es posible muestrear y graficar datos provenientes de operaciones del sistema de archivo para hacer una exploracion de los datos. Cuando las frecuencias de los eventos se categorizan con detalle, la grafica podra tener la forma de alguna de las distribuciones mas comunes. Rara vez sera exacta la coincidencia, pero si los eventos muestran patrones consistentes, las reglas asociadas con el tipo de distribucion de coincidencia puden utilizarse para desarrollar tacticas adecuadas de analisis.

Como se presentan las distribuciones.- Las distribuciones uniformes son aquellas en donde todos los eventos considerados tienen la misma probabilidad de ocurrir, se obtienen cuando se categorizan eventos independientes. La uniformidad puede obtenerse al eliminar los efectos de orden superior. El tiempo de espera para accesar un disco, dado que el tiempo entre las consultas al disco es muchas veces mayor que el tiempo de revolucion del mismo, posiblemente se distribuya en forma uniforme.

La uniformidad en la distribucion de registros en un archivo aleatorio indexado minimiza la interferencia y las coaliciones. La eliminacion de los efectos de orden superior es la idea que respalda muchas de las transformaciones de llave a direccion. El metodo del residuo de la division o modulo es representativo de este metodo.

Las distribuciones normales ocurren en forma natural cuando los eventos que se estan midiendo son sumas y productos de muchas acciones independientes, y cuando tienen una probabilidad constante de ocurrencia P. En realidad, la distribucion normal es utilizada para calcular las probabilidades de eventos discretos, cuando F (numero de muestras) tiende a infinito. Si el producto FP > 15, la curva normal resulta ser una aproximacion adecuada.

Sea un archivo consistente en bloques con un numero fijo f de registros de dos tamanos distintos que se presentan con igual probabilidad (p=0.5). Si f> 30, el espacio total esperado utilizado para los f registros se distribuira en forma normal. Si la distribucion de la longitud del registro varia en forma uniforme, la distribucion normal es una aproximacion adecuada con aun menos registros por bloque. Los registros cuyas longitudes se distribuyen en forma normal.

Si las longitudes del registro se distribuyen en forma normal, la secuencia de longitudes se distribuira normalmente ante cualquier numero de registros.

Las distribuciones exponenciales ocurren cuando la probabilidad de una serie de eventos aumenta o disminuye rapidamente conforme aumenta el valor del ultimo evento. Las distribuciones

exponenciales crecientes en las bases de datos las constituyen el numero de relaciones entre sus elementos y, por lo tanto, tambien el tiempo necesario para clasificar, ya que crecen en forma mas que lineal de acuerdo con el tamano del archivo.

Comunmente una distribucion exponencial negativa se genera a partir de eventos mutuamente dependientes, con la misma probabilidad. Tambien se presentan cuando se consideran los tiempos entre llegadas aleatorias de solicitudes de servicio. Las distribuciones exponenciales y de Erlang son importantes en el analisis de lineas de espera de solicitudes de servicio.

Las distribuciones de Poisson se presentan cuando ciertos eventos cuya contribucion es importante tienen una pequena probabilidad de ocurrencia.

Puede mostrarse que para eventos que tengan una distribucion exponencial de tiempo entre llegadas, el numero de llegadas esperadas en un periodo dado tiene una distribucion de Poisson.

Las distribuciones sesgadas se presentan siempre que las transformaciones no lineales afecten el resultado de los eventos. Estas distribuciones sesgadas se caracterizan por el hecho de que su media ocurre en un punto diferente que su mediana. Otra medida mas de tendencia central es la moda, que es la posicion del valor mas frecuente, en este caso cero.

Las distribuciones bimodales se presentan cuando los eventos que se estan midiendo se deben a dos fenomenos base separados. Un ejemplo de esto seria el tiempo de respuesta para consulta, cuando alguna puede contestarse utilizando un indice y otras requieren una lectura exhaustiva del archivo o de un subconjunto de este.

Funcion acumulada de distribucion.- Con frecuencia se utiliza una distribucion de frecuencia para obtener un calculo de la fraccion de casos que sobrepasan cierto limite. La altura de la curva es directamente proporcional al numero de eventos que ocurran con valores menores que el correspondiente.

4.1.2 Descripcion de las distribuciones

Cuando el resultado de una medicion genera una distribucion, cualquier semejanza con alguno de los tipos de distribucion mostrados proporcionara algunas claves acerca del proceso que provoco la secuencia de eventos. Como paso siguiente, pueden obtenerse algunas medidas cuantitativas. Dos parametros basicos utiles con casi cualquier distribucion observada son la media y la desviacion estandar de las F muestras tomadas. A fin de tener un calculo no sesgado, estas observaciones deben ser una muestra aleatoria de los eventos.

El numero de muestras debera ser lo suficientemente grande para que se tenga confianza en que el conjunto utilizado es representativo de los eventos que ocurren dentro del sistema.

4.1.3 Distribucion uniforme

Si una distribucion no muestra cambio en algun intervalo de valores, las tareas de calculo se simplifican generalmente. Una distribucion uniforme de busquedas sobre un disco requerira un tiempo de localizacion mayor que el promedio entre todas las distribuciones, con excepcion de alguna que sea bimodal. La probabilidad de que el siguiente registro no este disponible en el bloque actual es mayor para una distribucion uniforme de solicitudes que para cualquier otra; sin embargo, el costo promedio de las busquedas es menor cuando las inserciones de actualizacion se distribuyeron en forma uniforme.

El calculo de la desviacion estandar de las frecuencias, y la obtencion de un valor que sea relativamente pequeno con respecto a la altura o la frecuencia media indican un grado satisfactorio de uniformidad.

4.1.4 Distribucion normal.

A menudo puede esperarse que la diferencia entre frecuencias observadas y esperadas sea una variable distribuida normalmente, ya que se basa en muchas muestras independientes provenientes de los eventos observados.

Si una distribucion es aproximadamente normal, es posible aplicar muchas reglas utiles de calculo. En el diseno de archivos, a menudo resulta conveniente calcular con que frecuencia se excedera cierto limite; para una distribucion que es aproximadamente normal, el area acumulada bajo la curva normal mas alla del limite proporciona el calculo deseado.

Desde luego, aun cuando los datos se distribuyan exactamente en forma normal, pueden suceder eventos raros, como las altas tasas de actividad en los sistemas de procesamiento de datos que pueden provocar que estos eventos raros ocurran con frecuencia.

4.1.5 La distribucion de Poisson

Las distribuciones de Poisson ocurren debido a eventos independientes. La forma de la distribucion de Poisson se describe por completo mediante el valor de la media de las observaciones, ya que en esta distribucion la media y la varianza son iguales.

Los eventos que llevan a esta distribucion son tales que una curva de Poisson es muy probable: la distribucion es la suma de muchos eventos, cada uno con una baja probabilidad, y la probabilidad de los eventos en las categorias de numeros superiores tiende a cero.

Prueba de bondad de ajuste.- Puede realizarse una prueba mas formal para determinar si una distribucion se ajusta con los datos utilizando la funcion ji cuadrada. La prueba ji cuadrada o

X2, compara categorias de observaciones con sus esperanzas, pero cada categoria debe contener al menos una muestra esperada. Para evitar categorias no validas, los valores de Poisson de menos 1 pueden agruparse en la ultima categoria adecuada.

Los valores obtenidos para x2 pueden compararse con valores estandar que se basan en la consideracion de que la diferencia entre las distribuciones fue causada por eventos aleatorios. Estos valores estandar de x2 pueden calcularse segun se necesite, utilizando aproximaciones de distribuciones binomiales o empleando tablas y graficas estadisticas.
Para utilizar la distribucion x2, es necesario conocer el numero de grados de libertad.

Un valor muy alto de x2 hace poco probable que las frecuencias esten relacionadas; un valor muy bajo haria sospechar que los datos se arreglaron para mostrar un buen ajuste. La prueba ji cuadrada es muy util cuando se estan comparando distribuciones. Pueden utilizarse otras pruebas, como la t de Student y la F de Spearman Pearson para comparar las medidas y desviaciones estandar obtenidas a partir de muestras con sus valores esperados, si la distribucion se conoce o si se considera conocida.

4.1.6 Otros estadisticos

Algunas veces, una respuesta observada es una funcion de multiples variables. Con la consideracion de que existe independencia entre esas variables es posible separar los efectos, dado un numero suficientemente grande de observaciones. Para esta tarea se utilizan dos tipos de metodos de analisis: regresion multiple y analisis de varianza. La regresion determina una recta que describe una relacion lineal entre las variables individuales y la variable dependiente resultante. Muchas de las relaciones en sistemas de computacion no son lineales. El analisis de varianza relaciona causas y efectos mediante agrupamientos discretos y se ha empleado para analizar el desempeno de los sistemas de operacion por pagina, con base en las variales, tamano de memoria, tamano del programa, secuencia de carga y algoritmos de manejo por paginas. Tanto la regresion como el analisis de varianza consideran relaciones lineales de la combinacion de las variables independientes con la variable dependiente.

4.2 SIMULACION.

Cuando no se dispone de ninguna fuente de datos que pueda utilizarse para predecir el comportamiento de un sistema nuevo, puede construirse un modelo a escala para generar datos predictivos. Tal modelo de simulacion se basara en el proceso y en el equipo que se vayan a utilizar en la realidad, y se alimentara con una secuencia de descripciones de los calculos deseados. La salida de una simulacion no estara constituida por los resultados del calculo, sino por mediciones recolectadas durante la simulacion.

En la figura 4.1 se muestran los componentes principales de una simulacion discreta, manejada por eventos. La entrada a la simulacion es una linea de espera de eventos, que contiene anotaciones de eventos que se crean externamente al sistema simulado. La salida es un registro de eventos con sus tiempos, que pueden analizarse aun mas.

Las anotaciones externas de eventos pueden generarse mediante un programa. Este utilizara la distribucion esperada de solicitudes para plantear los tipos y sus tiempos. Otra fuente de eventos externos puede ser un archivo proveniente de un sistema ya existente, o un programa sintetico que genere, durante su ejecucion, solicitudes de eventos. Las bitacoras o los programas tienen la ventaja de que proporcionan un estandar para probar el sistema.

Proceso de simulacion. Una simulacion tiene su propio mecanismo de control de tiempo. Para determinar cuanto durara el calculo simulado en la realidad, la simulacion utiliza un reloj variable para registrar el tiempo simulado. El reloj se incrementa en el siguiente evento que se va a simular.

Cuando se simula un archivo de acceso directo, resultan adecuados los siguientes pasos :

1) Se calcula una direccion de bloque para un registro con la llave deseada utilizando la transformacion de llave a direccion que se este considerando.

2) Se determina la posicion del cilindro de entrada.

3) Se calcula el tiempo para la localizacion a partir de la posicion actual.

4) Se determina el tiempo necesario de localizacion.

5) Se genera un tiempo de espera aleatorio.

6) Se calcula el tiempo de transferencia de bloques a partir de la razon de transferencia y del tamano de bloque.

7) Se utiliza una funcion aleatoria para determinar si existe una colision, en cuyo caso se regresa al paso 2 para calcular los tiempos de incremento; de otra manera.

8) Se calcula el tiempo de conclusion.

9) El estado de la unidad de disco utilizada se iguala a "ocupado".

Los eventos internos se unen con los externos y la simulacion continua hasta que se acaba la lista de eventos.

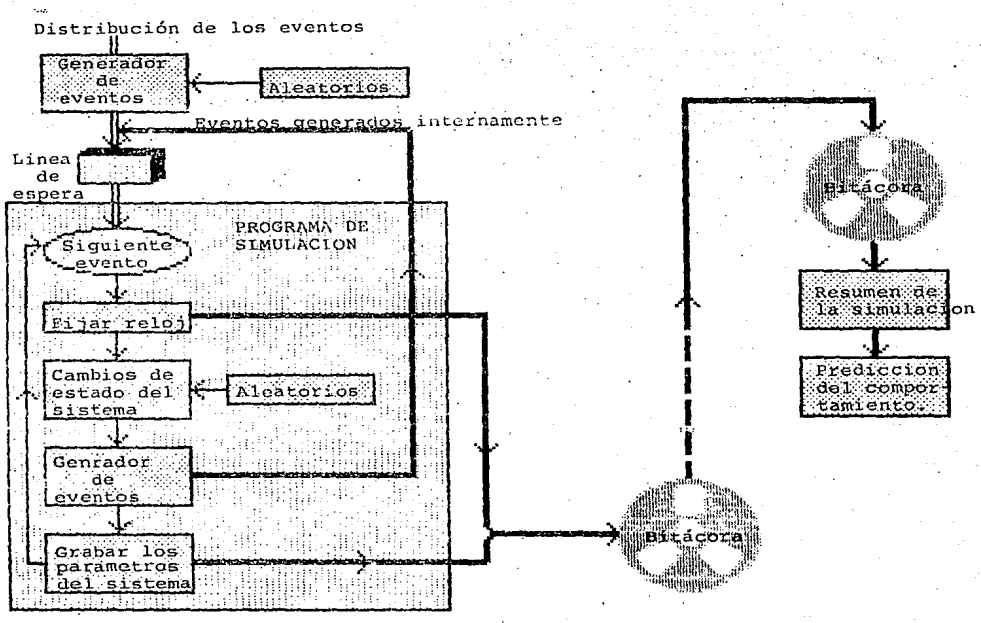


Figura 4.1 Componentes de una simulación

Cuando un dispositivo se encuentra ocupado, las solicitudes se colocan en una linea de espera. Una senal de que el "dispositivo esta desocupado", proveniente de la linea interna de espera, indica a la simulacion que revise la linea de espera del dispositivo, seleccione una solicitud para procesarla y efectue la simulacion.

Un problema de las simulaciones manejadas por listas externas de eventos es que no existe retroalimentacion del sistema al generador. Esto esta bien si el proceso que solicita servicios es realmente independiente del sistema.

Estas solicitudes no independientes se manejan de mejor manera mediante un modelo interno de la simulacion. Cuando se utilizan eventos generados aleatoriamente, son necesarias multiples y largas pruebas segmentadas para verificar la estabilidad de los resultados de la simulacion.

Los parametros de interes se procesan en forma incrementada o se escriben en una bitacora para su condensacion posterior. Los parametros de interes comunmente son el tiempo total para procesar todos los eventos, el tiempo necesario para procesar eventos individuales y las razones de actividad de los dispositivos empleados. Los datos obtenidos pueden presentarse en forma conveniente como histograma o describirse mediante distribuciones. Entonces, es posible hacer predicciones como "el tiempo de respuesta para una consulta sera menor de 10 segundos en un 90% de las veces."

Como escribir una simulacion.- El empleo de tecnicas de simulacion para investigar el comportamiento de sistemas de archivo se simplifica debido a la gran diferencia de las velocidades del procesador y del archivo. A menudo es posible obtener resultados adecuados mediante una simulacion que se concentre en los retrasos de archivo y que sea un modelo de computo solo en forma rudimentaria.

Se dispone de cierto numero de lenguajes de simulacion que se han utilizado ampliamente para la simulacion de sistemas. Los lenguajes de simulacion de que se dispone en la actualidad, utilizan tecnicas de procesamiento por lotes (en batch). Cuando una simulacion es grande, el tiempo de computadora necesario puede volverse un elemento significativo; por lo tanto, es necesario planear cuidadosamente las simulaciones, de manera que proporcionen un beneficio adecuado conforme al esfuerzo invertido.

4.3 LINEAS DE ESPERA Y TECNICAS DE MANEJO CRONOLOGICO

Cuando un calculo solicita servicio a un dispositivo que esta ocupado, el calculo se retrasara. Ya que el calculo no debe abandonarse, la solicitud se anotara en una linea de espera del dispositivo.

Lineas de espera y procesos.- Un calculo puede iniciar multiples procesos que iran en forma independiente a solicitar servicios al sistema de archivo. Es imposible que el calculo total continue mas alla de ciertos puntos hasta que los procesos iniciados se hayan concluido.

Cada solicitud de servicio en una linea de espera se debe a un proceso individual. La suma de todas las longitudes de las lineas de espera en el sistema esta limitada por el numero de procesos posibles en dicho sistema. Este numero puede ser bastante grande.

Los procesos, a menos que esten restringidos en forma especifica, operan asincronicamente. Esto significa que no es conveniente generar muchos procesos individuales para que escriban una secuencia de bloques a un archivo: es posible que los bloques no se escriban en el orden deseado; sin embargo, si cada uno de los bloques tienen su propia direccion, un buen sistema de manejo de lineas de espera sera capaz de escribir los bloques al archivo, en un orden que reduzca los retrasos de hardware. Cada proceso de escritura o lectura estara asociado por una o mas areas de memoria que contendran los datos. El numero de areas asociadas posiblemente limite el numero de procesos activos y el grado de operaciones posibles en paralelo.

4.3.1 Manejo de lineas de espera

Las actividades que forman parte del manejo de lineas de espera para archivos se presentan en la figura 4.2. Habra cuando menos una linea de espera por cada dispositivo de archivo y puede haber tambien algunas lineas de espera para otros tipos de dispositivo, todas manejadas cronologicamente por el mismo programa de lineas de espera.

La razon mas comun para que el sistema operativo llame al programa es que un dispositivo haya sido liberado por el proceso que lo estaba ocupando. A menudo esta liberacion se inicia por una senal de hardware que indica que se ha concluido la transferencia real de los datos. Esto ira seguido por un procedimiento de verificacion, despues de lo cual el proceso ya no necesita al dispositivo. Ahora este dispositivo puede asignarse a otro proceso.

Ocasionalmente, debido a una falta de equipo o de algun programa, algun proceso no libera o tal vez ni siquiera comienza a utilizar el dispositivo. Si se especifica un limite de tiempo o tiempo de parada, es posible cancelar el proceso cuando este falle o exceda el tiempo permitido, y asigne el dispositivo a otro proceso. En los sistemas en que no se especifican limites de tiempo, todos los calculos que soliciten un dispositivo se detendran y podran cesar toda la actividad del sistema.

En los sistemas interactivos las lineas de espera para unos cuantos dispositivos a menudo son largas, mientras que otras se encuentran vacias. La vigilancia de las lineas de espera resulta importante cuando los sistemas deben funcionar en su mejor

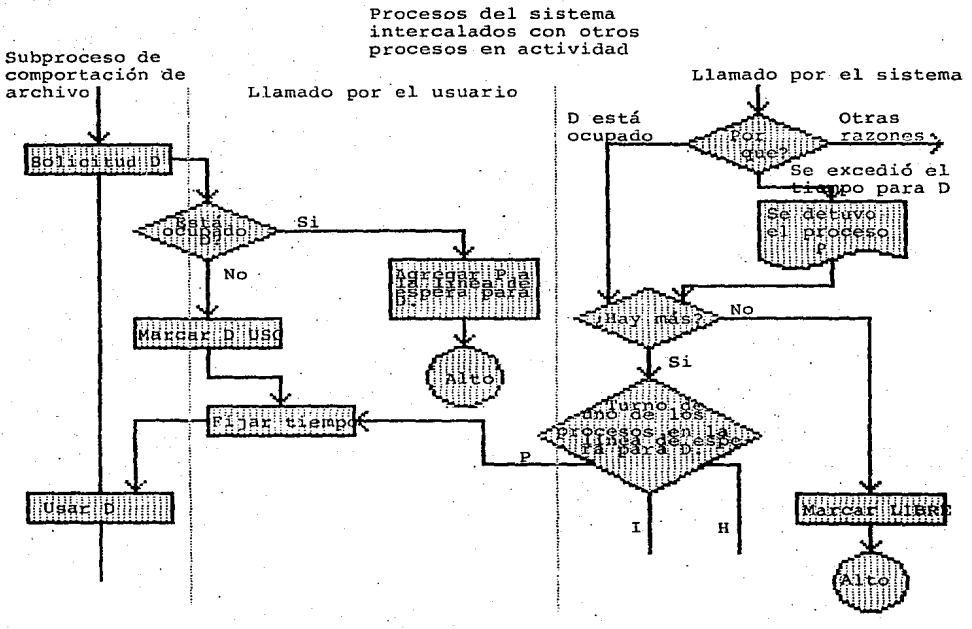


Figura 4.2 Manejo de lineas de espera por un dispositivo.

desempeno. Es posible crear lineas de espera en forma dinamica cuando se agrega un nuevo recurso al sistema.

4.4 LA INVESTIGACION DE OPERACIONES EN EL DISENO DE BASES DE DATOS

La investigacion de operaciones es un area de las matematicas orientada a los problemas de manejo cronologico, operativos y de desarrollo de instalaciones.
Las herramientas de la investigacion de operaciones incluyen:

Teoria de lineas de espera
Teoria de inventarios
Programacion lineal y entera
Teoria de decisiones

Actualmente, cada uno de estos campos constituye por si mismo una disciplina.

4.4.1 Distribuciones de lineas de espera y su aplicacion

Debido a la importancia que tienen las lineas de espera en las comunicaciones, se mencionan a continuacion los parametros necesarios para el tratamiento matematico de las lineas de espera:

- Descripcion de la fuente de solicitudes de servicios
- Numero de solicitudes
- Distribucion de la tasa de llegada
- Descripcion de la linea de espera para un servicio
- Capacidad de la linea de espera
- Politica de manejo cronologico
- Descripcion del servicio
- Numero de servidores
- Distribucion del tiempo de servicio

Las distribuciones de los procesos de llegada o de servicio que a continuacion se muestran

M : exponencial
D : constante o determinado de alguna otra forma
E (m): Erlang
I : general e independiente
G : general

resultan de interes para los problemas que se presentan en el analisis de archivos, principalmente los modelos de lineas de espera del tipo M/G/I y E/G/I.

En las lineas de espera los eventos se describen mediante su razon promedio de llegada. Una alternativa a la razon de llegada es la distribucion de los tiempos entre llegadas. En el caso en que la distribucion de llegadas no sea ni aleatoria ni constante, es preferible describirlas mediante distribuciones de tipo Erlang.

4.4.2 Aplicacion de la teoria de inventarios

Segun la teoria de inventarios clasica existe un conjunto de bienes que se consumen gradualmente. Antes de que el inventario se agote es necesario establecer una orden para que haya reabastecimiento. Hay un costo por realizar una orden, almacenar el inventario y agotar los bienes. Se dispone de una gran variedad de modelos para resolver los problemas de inventario.
Una base de datos tiende a ser menos eficiente con el transcurso del tiempo, hasta que se reorganice, de manera que su comportamiento presente una funcion de costo semejante a la que se muestra en la figura 4.3.

En los archivos directos el costo esta relacionado con el grado de manejo por conglomerados y por la densidad de archivo.
Si se dispone de estadisticas acerca de la actividad de registro, es posible utilizar la reorganizacion para mejorar la eficiencia del archivo. Aun cuando el metodo y su efecto dependen de su organizacion, los costos de uso seran menores si se vuelven a cargar primero los registros de mayor actividad. Por ejemplo, en un archivo directo serian aquellos registros de mayor actividad.

Se considerara solo el caso de un archivo estatico, debido a que el principio del analisis permanece igual. El costo de reorganizacion es equivalente al costo de establecer una orden, pudiendo ignorarse el tiempo necesario entre la realizacion de la orden y la realizacion del servicio.

4.5 ASIGNACION DEL ALMACENAMIENTO

Un sistema de computacion debe poder day apoyo a diversos tipos de archivos. Una funcion del sistema operativo consiste en asignar el espacio de almacenamiento a los diferentes archivos. Existe una interaccion entre la estructura del archivo y la politica de asignacion: los sistemas de archivo que no utilizan indices o apuntadores para localizar los bloques requieren de espacio continuo; otros archivos pueden recibir el espacio segun sean necesarios.
El sistema inicialmente establece el area necesaria para su operacion y despues define las areas disponibles para los usuarios. La primera tarea del sistema consiste en crear un directorio de todas las unidades de almacenamiento disponibles. Durante la operacion el sistema tendra que identificar los paquetes removibles de disco, las cintas, y otros elementos que se encuentren montados en ese momento. El directorio de los dispositivos de almacenamiento vuelve a crearse siempre que se inicialice el sistema, y las anotaciones de dicho directorio se actualizan al recibir una senal de "listo" proveniente de un dispositivo en el cual se haya montado un paquete de disco o algo semejante. Cuando se hace esto, los usuarios pueden regresar y demandar su parte de los recursos. Debe observarse que la asignacion de espacio es una funcion que se realiza mas elementalmente que como opera el sistema de archivo.

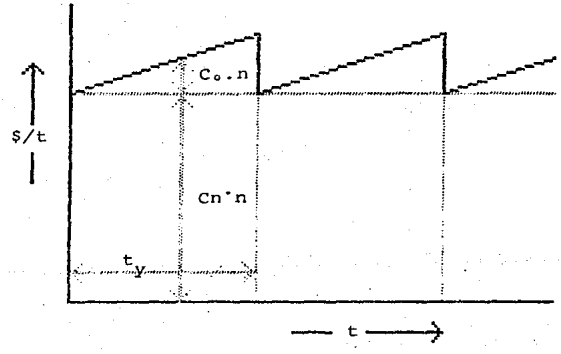


Figura 4.3 Función de costo de un archivo que se está actualizando.

Cuando varios sistemas comparten un recurso de computacion, el sistema operativo tendra que usar mas de una estrategia de asignacion de espacio. El manejo de espacio resulta esencial para conservar en equilibrio la operacion del sistema.

4.5.1 Porciones, tamano comparado con numero

El espacio se asignara a los usuarios, como respuesta a sus solicitudes. La forma, el numero y el tamano de las porciones del espacio de almacenamiento se dan a los usuarios de manera equilibrada entre las necesidades, eficiencia de los archivos aislados y la eficiencia general del sistema.
Para evitar el desperdicio de espacio de almacenamiento las porciones seran de diferentes tamanos.
En los sistemas multiusuarios no es muy benefico proporcionar espacio contiguo a los usuarios,de manera que la asignacion puede basarse en bloques o en trenes de bloques. Algunos esquemas de archivo podran utilizar el espacio extra asignado y solicitaran menos espacio la siguiente vez.
La asignacion mediante porciones variables deja mas espacio desocupado, pero es mas dificil de utilizar que la asignacion por bloque o tren. La eficiencia de la asignacion por bloque mejora con bloques o trenes pequenos. Cuando se asignan bloques o multiplos de bloques, las tablas de asignacion de almacenamiento seran mayores. Debido a que es muy comun que el almacenamiento secundario se asigne durante largos periodos de tiempo, las tablas de control de almacenamiento residen en el disco.

Manejo de porciones asignadas.- Cuando sea necesario asignar porciones al archivo debera equilibrarse el desperdicio de espacio y la falta de contiguidad. Para lograr esta, un enfoque consistiria en especificar previamente el tamano posible de archivo y asignar grandes porciones que puedan permanecer sin uso. Muchos otros sistemas abandonan la contiguidad y asignan bloques conforme resulte necesario. Esta estrategia tambien requiere de grandes tablas para encontrar las porciones de un archivo y; ocasionalmente, puede afectar en forma importante en el desempeno de un archivo directo.

Un esquema para evitar el problema de la asignacion previa debido al desperdicio o a la falta de contiguidad consiste en asignar porciones mayores cada vez conforme el archivo crezca. Si la siguiente porcion se duplica, el tamano de la tabla de asignacion se reducira logaritmicamente. En cualquier momento, menos de la mitad del espacio asignado permanece sin utilizarse.

Asignacion del espacio desocupado.- La tabla de control de almacenamiento tambien contiene, en formas explicita o implicita, la informacion referente a aquellas areas de almacenamiento secundario que aun no se han asignado y que, por lo tanto estan disponibles.

Para asignar el espacio desocupado se recupera una seccion de la tabla de control de almacenamiento del disco para determinar el espacio para un usuario.

Antes de dar cualquier porcion a un usuario es conveniente escribir la tabla de control, para indicar que la porcion ya no esta desocupada. De no hacerse esto de inmediato, una falla del sistema podria provocar la perdida de esta informacion.
Para evitar esto es posible conservar una bitacora que senale las porciones desocupadas en un area de trabajo de la memoria secundaria. Para conservar segura la asignacion de almacenamiento, el calculo deberia tener esta secuencia:

1) Obtener las porciones desocupadas de la tabla de control de almacenamiento para asignarlas a un usuario.
2) Marcar estas porciones en la bitacora como 'en uso'.
3) Volver a grabar la bitacora en el disco.
4) Asignar porciones del area de trabajo a los usuarios conforme se reciban las solicitudes, identificando las porciones asignadas.
5) Se regresa al paso 1.

Despues de una falla habra anotaciones para porciones marcadas "perteneciente al usuario X", anotaciones para porciones marcadas "desocupado" y anotaciones para porciones marcadas "en uso". Las ultimas porciones no se volveran a asignar. Si las porciones asignadas se identifican con su propietario, fecha y tiempo de la operacion mas reciente de escritura, se pueden verificar las anotaciones cuando el usuario las recupera de nuevo, y corregir eventualmente la tabla de control de almacenamiento. Es posible utilizar otras tecnicas, dependiendo del protocolo del sistema del archivo. Estas tareas de mantenimiento y conservacion pueden diferirse y combinarse con tareas de reorganizacion de archivo, ya que la tabla de control de almacenamiento permanece utilizable.

4.5.2 Tablas de control de almacenamiento

La estructura de la tabla de control de almacenamiento se ve afectada por el procedimiento de asignacion al que tiene que apoyar. Existen tres metodos de uso comun para definir el espacio disponible de almacenamiento y su asignacion:

a) Tabla de contenido.- Un archivo mantenido por el sistema en cada unidad, puede utilizarse para describir las asignaciones de espacio utilizando uno o varios registros por archivo. Un archivo puede tener una descripcion extensiva. Comunmente se incluyen elementos tales como la identificacion del propietario, el nombre del archivo, la fecha de creacion, la fecha de uso mas reciente, la fecha de actualizacion mas reciente, y por cada porcion, la posicion y el tamano. Normalmente esta tecnica se utiliza en sistemas que asignan grandes porciones. Esta tabla de contenido se conserva en cada disco del sistema y se encuentra a partir de una posicion fija en cada unidad.

   Las porciones desocupadas podrian asignarse buscando en la tabla y encontrando el espacio no asignado, pero posiblemente esto resulte costoso. A fin de evitar este costo se conserva

en la tabla de contenido un archivo ficticio con propietario X y nombre ARCHIVO. Ya que el numero de porciones de espacio desocupado puede ser mayor que el limite de porciones por archivo unico, las porciones desocupadas pueden encadenarse unas con otras y encontrase a partir de una o varias anotaciones.

b) Encadenamiento de porciones.- Las porciones desocupadas pueden encadenarse unas con otras. Tambien es posible aplicar este metodo cuando las porciones son mas pequenas, por ejemplo en la asignacion basada en bloques. Tiene un sobrepaso de espacio despreciable, ya que utiliza el espacio desocupado para almacenamiento de los datos de control. En una anotacion adicional se conservan pocas descripciones como encabezado. Cuando se necesita el espacio, se recupera uno por uno de los encabezados de los archivos para determinar la siguiente porcion libre en la cadena. Despues se quita la porcion de la cadena y esta ultima se vuelve a escribir, evitando que los datos de control de almacenamiento desocupado sean vulnerables a fallas del sistema.

Es posible conservar multiples cadenas para agrupar cilindros o sectores, con el fin de mejorar la eficiencia. Si se estan asignando porciones de longitud variable, puede haber multiples cadenas para implantar un algoritmo de mejor ajuste. Los archivos obtendran porciones de la cadena que sean adecuadas para el tamano de las porciones requeridas, y se conservaran las porciones grandes hasta que sean necesarias.

El manejo encadenado de almacenamiento desocupado no lleva control de los datos asignados al archivo del usuario. Es conveniente que el sistema operativo conozca todas las asignaciones realizadas. Cuando no existe tabla de contenido, esto puede resolverse mediante una extension del metodo de encadenamiento. Las porciones asignadas se encadenan unas con otras para tal archivo, utilizando un encabezado semejante al empleado para el manejo de almacenamiento desocupado. El exceso relativo de espacio para hacer esto depende del tamano de las porciones asignadas.

c) Tablas de bits.- Un tercer enfoque para el manejo del espacio desocupado consiste en elaborar una tabla de bits. Este metodo utiliza un arreglo que contiene un bit por cada porcion del sistema, y solo se utiliza cuando todos los tamanos de porcion son iguales. Cada valor "0" indica una porcion desocupada correspondiente y un "1" indica una porcion en uso. La lista de porciones asignadas a un archivo especifico se conserva como parte del mecanismo del archivo. Una tabla de bits tiene la ventaja de que grandes secciones de esta pueden conservarse en la memoria y que el costo de asignacion y eliminacion de asignacion es minimo. Esta ventaja es de cierta importancia en los medios altamente dinamicos. Por otra parte, las tablas son bastante vulnerables a los errores, por lo que resultan practicas solo

cuando el sistema de archivo lleva control de su asignacion, de modo que puedan ser utilizadas por el sistema operativo. En este caso es posible reconstruir la tabla de bits mediante todas las tablas individuales de los archivos.

Los sistemas de archivo tambien pueden utilizar tablas de bits para localizar sus porciones miembro. El uso de tablas de bits es especialmente efectivo cuando los datos de asignacion de porciones no tienen que conservarse en orden seriado. Las tablas de bits para un archivo especifico tienden a ser muy dispersas, pero puede incrementarse la eficiencia mediante una tecnica de empacado. El empacado de una tabla dispersa de bits se logra indexando grupos del arreglo de bits. Un grupo que tenga solo anotaciones cero se indexa mediante un cero y en realidad no se representa. El indice puede agruparse e indexarse recursivamente.

Tambien se han utilizado arreglos de bits cuando varios archivos comparten porciones o registros. Debido a que no se requiere de espacio dentro de la porcion, no existe limite en el numero de propietarios que pueda tener dicha porcion.
Es posible que puedan existir algunos objetivos en conflicto en el diseno de un algoritmo de manejo del espacio desocupado. De nuevo, el conflicto principal esta en la velocidad comparada con el espacio. Para aumentar el desempeno del archivo, resulta conveniente proporcionar grandes porciones y asignar, ademas, nuevas porciones cercanas a las predecesoras. Para disminuir el costo del manejo del espacio tambien puede resultar conveniente asignar grandes porciones, lo cual reduce la frecuencia de asignacion, asi como el tamano de las tablas y listas necesarias. Para utilizar bien el espacio de almacenamiento resulta conveniente asignar porciones variables o pequenas.

CAPITULO 5

Análisis del Texto e Indexación Automática

5.1 INTRODUCCION

Dentro del ambiente de procesamiento de documentos uno de los procedimientos mas importantes y dificiles de llevar a cabo es el analisis de la informacion, que consiste en asignarle a cada documento un conjunto de identificadores que permitan representar el contenido del documento.

5.2 INDEXACION MANUAL Y AUTOMATICA

En teoria cuando se tiene una coleccion de documentos pequena el hacer un analisis del contenido es redundante, siendo mas conveniente explorar el texto completo de todos los articulos cada vez que se requiera informacion. Esto en la practica consume demasiado tiempo y resulta muy costoso; por lo que se acostumbra caracterizar a cada articulo asignandole una descripcion corta o perfil; de manera que se pueda obtener el articulo deseado a partir de las caracteristicas de su contenido.
Las operaciones de analisis en el medio bibliotecario estandar se conocen como catalogacion, clasificacion, indexacion y elaboracion de resumenes.

El perfil del documento se divide en dos partes:

1) Informacion objetiva relacionada con los datos externos al contenido del texto como: nombre del autor, editor, fecha, lugar de publicacion, etc. Conocido en las bibliotecas convencionales como titulo/autor o catalogacion descriptiva.

2) Informacion que describe el contenido del documento, conocido en las bibliotecas convencionales como: catalogacion tematica.

La descripcion del contenido del articulo consiste de:

- Un numero elegido de una lista sistematica organizada jerarquicamente.

- Encabezamientos por materia que pueden ser representados por palabras y frases.

Cada encabezamiento por materia que se asigna a un articulo puede introducirse en una tarjeta de catalogo por separado. La coleccion resultante constituye un catalogo de biblioteca.

En una biblioteca moderna los perfiles de busqueda se introducen en una base de datos. En este caso a las operaciones de analisis de contenido de textos se les conoce como "indexacion", cuando la asignacion de identificadores del contenido se hace con ayuda de una computadora, la operacion se convierte en una "indexacion automatica".

Podria pensarse que la catalogacion descriptiva no presenta ningun problema; sin embargo, ha sido necesario elaborar estandares de catalogacion para definir que campos son aceptables y que restricciones pueden presentar. (Vease formatos de comunicacion).

La asignacion de identificadores de contenido se hace para cumplir con:

1) Localizar los articulos que tengan topicos interesantes para el usuario.
2) Relacionar unos articulos con otros.
3) Predecir la importancia de los articulos que cumplan con los requerimientos de informacion especifica a traves del uso de terminos indexados con un enfoque bien definido y con sentido.

La diferencia entre terminos indexados controlados o no controlados radica en que el vocabulario de indexacion no controlado que en teoria puede incluir toda la variedad del lenguaje natural permite que haya ambiguedad y errores. Mientras que el lenguaje controlado es un lenguaje limitado que con frecuencia se avoca a los terminos disponibles para la identificacion del contenido y estos terminos son rigidamente controlados.

Esto permite controlar el vocabulario y la eliminacion de sinonimos, haciendo referencia a un termino unico para cada clase de sinonimo e identificando los terminos relacionados semanticamente. El uso de terminos controlados garantiza la recuperacion de los articulos senalados apropiadamente cuando se conocen los terminos correctos de la busqueda, aunque esto implica la necesidad de tener intermediarios entrenados para que formulen las instrucciones de pregunta (query).

Tambien es necesario hacer la diferencia entre el uso de terminos simples para caracterizar el contenido de un documento, opuesto al uso de terminos en contexto donde los indicadores de relacion pueden conectar varios identificadores, y las unidades basicas pueden consistir de entradas compuestas y frases.

En el modo de termino simple, los identificadores de contenido, conocidos como terminos indice, palabras-llave o descriptores se representan por palabras individuales. Cada documento esta caracterizado por una coleccion de terminos individuales.

Cuando se formulan requerimientos de busqueda, los terminos se pueden combinar o coordinar formando descripciones de topicos. A este proceso se le conoce como "post-coordinacion".

Cuando los terminos compuestos (frases, sustantivos, adjetivos, preposiciones y/o indicadores de relacion) se utilizan para propositos de indexacion, el proceso se llama "pre-coordinacion".

Por ejemplo, cuando se indexa un titulo bajo: "Recuperacion automatica de informacion" se trata de una pre-coordinacion, pero si se hace la indexacion de los terminos por separado y despues se hace una busqueda de "recuperacion" e "informacion", esto representa una post-coordinacion.

Cuando la indexacion es manual es preferible utilizar lenguajes de indexacion controlados utilizando terminos compuestos pre-coordinados. En cambio cuando los sistemas son indexados automaticamente es preferible utilizar la post-coordinacion, debido a que se puede hacer una combinacion de terminos simples en el momento de la busqueda.

Sin importar que tipo de indexacion se utilice, debe tomarse en cuenta la exhaustividad y especificidad de los productos de la indexacion. Donde:

- La exhaustividad es el grado en que los conceptos y nociones que se incluyen en un documento son reconocidos en las descripciones de indice.

Entre mas exhaustiva sea la indexacion mayor ser el numero de documentos pertinentes que se pueden recuperar.

- La especificidad es el nivel generico de los terminos indice utilizados para caracterizar el contenido del documento.

Si se tiene un vocabulario de indexacion especifico, al hacer una busqueda se pueden rechazar muchos documentos que no son pertinentes.

A estas dos caracteristicas se les conoce como indexacion profunda y superficial.

La indexacion profunda implica un alto grado de exhaustividad y especificidad y por lo tanto un buen funcionamiento de recuperacion. La indexacion superficial se produce al utilizar unos cuantos terminos generales para caracterizar a cada documento.

5.3 EXTRACCION AUTOMATICA DE TERMINOS Y PONDERACION.

5.3.1 Consideraciones generales

A continuacion se presentan metodos que permiten extraer terminos del contenido de los documentos, hacer resumenes de los documentos y ponderar los terminos dependiendo de su importancia.

Si todas las palabras tuvieran una frecuencia de ocurrencia igual, seria imposible distinguirlas utilizando un criterio cuantitativo.

Sin embargo, se ha observado que las palabras tienen una frecuencia de ocurrencia distinta en un texto de lenguaje natural. Esto se debe a que es mas facil repetir algunas palabras en vez de utilizar otras nuevas y diferentes. Tambien se observa que las palabras que se utilizan con mas frecuencia son palabras cortas que son faciles de inventar y cuyo esfuerzo de uso es minimo, como se muestra en la tabla 5.A. Por lo tanto es posible distinguir clases de palabras dependiendo de su frecuencia de ocurrencia.

A partir de las caracteristicas de frecuencia de palabras idividuales en textos de documentos es posible derivar factores significativos de palabras.

Una propuesta teorica basica se fundamenta en las siguientes consideraciones generales:

1) Dada una coleccion de n documentos, calcule para cada documento la frecuencia de cada termino unico en ese documento.

2) Determine la frecuencia de la coleccion total para cada palabra, sumando las frecuencias de cada termino unico a traves de los n documentos.

3) Acomode las palabras en orden decreciente de frecuencia. Designe un valor de umbral superior y elimine todas las palabras que tengan un valor de frecuencia que sobrepase este umbral. Esto elimina las palabras con una funcion de frecuencia mayor, como se muestra en la tabla 5.A.

4) De la misma forma elimine las palabras de baja frecuencia. Esto elimina los terminos que ocurren rara vez en la coleccion y cuya presencia no afecta significativamente la recuperacion.

5) Las palabras con una frecuencia media permanecen y se utilizan para asignarlas a los documentos como terminos indice.

El poder de resolucion es la habilidad de los terminos indice para identificar articulos pertinentes y distinguirlos del material no pertinente; por lo tanto, la eliminacion de terminos de alta y baja frecuencia se debe a que estos terminos no son buenos identificadores de contenido.

Pero en la practica:

1) La eliminacion de todas las palabras de alta frecuencia pueden producir perdidas en la funcion de recoleccion, debido a que el uso de palabras de alta frecuencia para la identificacion del contenido es efectiva en la recuperacion de un gran numero de articulos importantes.

RANGO (R)	TERMINO	FRECUENCIA (F)	R.(F/1,000,000)
1	the	69,971	0.070
2	of	36,411	0.073
3	and	28,852	0.086
4	to	26,149	0.104
5	a	23,237	0.116
6	in	21,341	0.128
7	that	10,595	0.074
8	is	10,099	0.081
9	was	9,816	0.088
10	he	9,543	0.095

TABLA 5.A Ilustracion de la frecuencia de ocurrencia de algunas de las palabras mas utilizadas en el idioma ingles.

2) La eliminacion de terminos de baja frecuencia puede producir perdidas en la precision.

3) Elegir en forma conveniente los umbrales para distinguir los terminos de frecuencia media.

4) Medir la frecuencia para la identificacion de identificadores de contenido, debido a que un termino indice debe cumplir con:

- El termino indice debe estar relacionado con el contenido de la informacion del documento, de manera que proporcione la recuperacion del articulo cuando se requiera (funcion de recoleccion).

- Un buen termino indice debe distinguir los documentos a los cuales esta asignado, para prevenir la recuperacion indiscriminada de todos los articulos, se quieran o no (funcion de precision); por lo tanto un termino como "computadora" no debe constituir un termino razonable para asignarlo a una coleccion de documentos del area de computacion, no importa la frecuencia que tenga, dado que este termino ocurrira en cada articulo de la coleccion y por lo tanto no sirve para distinguir un articulo de otro.

A partir de estas consideraciones basicas se han derivado varias funciones de ponderacion de terminos como:

- La funcion de frecuencia inversa de documentos.
- La razon senal a ruido
- Los valores de discriminacion de terminos.

5.3.2 Ponderacion de frecuencia inversa de documentos

La ponderacion de frecuencia inversa de documentos consiste en asumir que la importancia de terminos es proporcional a la frecuencia de ocurrencia de cada termino k en cada documento i ($FREC_{ik}$) e inversamente proporcional al numero total de documentos a los cuales esta asignado cada termino; es decir, es el numero de documentos en los cuales aparece el termino k.

Una forma de medir la frecuencia de documentos inversa es:

$$\log_2 \frac{n}{FREC\_DOC_k} + 1 = \log_2(n) - \log_2(FREC\_DOC_k) + 1$$

donde n es el numero de documentos de la coleccion.

Este metodo hace enfasis en los terminos con menor frecuencia de documentos.

Una expresion compuesta que mide la importancia o ponderacion del termino k en el documento i aumenta cuando la frecuencia del termino en el documento FREC aumenta, pero disminuye cuando la frecuencia del documento (FREC_DOC) aumenta. Una posible funcion de ponderacion es:

$$ponderacion_{ik}=FREC_{ik}[\log_2(n)-\log_2(FREC\_DOC_k)+1]$$

Esta funcion asigna un alto grado de importancia a los terminos que ocurren en unos cuantos documentos de la coleccion.

5.3.3 Razon senal a ruido

Las bases de la teoria de informacion son aplicables a la construccion de un medidor de la importancia de terminos. Por ejemplo, el contenido de informacion de un mensaje, o sea un termino puede medirse como una funcion inversa de la probabilidad de ocurrencia de las palabras en un texto dado. El contenido de informacion de una palabra se mide como:

$$INFORMACION=-\log_2 P$$

Donde P es la probabilidad de ocurrencia de la palabra.

Por ejemplo:
- Si la palabra BIOLOGIA ocurre una vez en cada 10,000 palabras.
- Su probabilidad de ocurrencia es 0.0001
- Y su informacion es: INFORMACION=-log (0.0001)=13.278

El valor de la informacion del termino, puede verse como una reduccion de la incertidumbre, cuando los terminos se asignan a identificadores de contenido, debido a que cuando se conoce un termino se reduce la incertidumbre acerca del contenido del documento.

Cuando un documento esta caracterizado por t identificadores o terminos, cada uno de ellos ocurre con una probabilidad P_k, el promedio o informacion esperada (es decir, la reduccion del promedio de incertidumbre acerca del documento) que se obtiene de uno de los terminos esta dada por la formula de Shannon:

$$INFORMACION\ PROMEDIO=-\ P_k \log P_k$$

Por ejemplo, si se espera que los terminos BIOLOGIA, MATEMATICAS, INGLES e HISTORIA ocurran con una probabilidad de 0.5, 0.2, 0.2 y 0.1 respectivamente, se tiene que:

$$INFORMACION\ PROMEDIO=-[(0.5 \log_2 0.5)+(0.2 \log_2 0.2)+(0.2 \log_2 0.2)+(0.1 \log_2 0.1)]=1.3$$

La informacion promedio se maximiza cuando cada una de las probabilidades de ocurrencia de t terminos diferentes es 1/t.

Por ejemplo si la probabilidad de ocurrencia de los terminos BIOLOGIA, MATEMATICAS, INGLES e HISTORIA fuera de 1/4, entonces el valor de la informacion promedio seria 2 en vez de 1.3 como en el ejemplo anterior.

El ruido $RUIDO_k$ de un termino con indice K para una coleccion de n documentos se puede definir a partir de una analogia con la medicion de la informacion de Shannon.

$$RUIDO_k = \frac{FREC_{ik}}{FREC\_TOT_k} \log_2 \frac{FREC\_TOT_k}{FREC_{ik}}$$

Esta medicion del ruido varia inversamente con la "concentracion" de un termino en el documento de la coleccion. Es decir, el ruido se maximiza cuando se tiene una distribucion uniforme perfecta, donde un termino ocurre un numero identico de veces en cada documento de la coleccion.

Por ejemplo, si el termino k ocurre exactamente una vez en cada documento (todas las $FREC_k$=1)

$$RUIDO_k = \frac{1}{n} \log_2 \frac{n}{1} = \log_2 n$$

Por otra parte, en una distribucion concentrada perfectamente, donde un termino aparece en un solo documento con frecuencia $FREC\_TOT_2$, el ruido es cero.

$$RUIDO_k = \frac{FREC\_TOT_k}{FREC\_TOT_k} \log_2 \frac{FREC\_TOT_k}{FREC\_TOT_k} = 1 \log_2 1 = 0$$

Existe una relacion bien definida entre el ruido y la especificidad de terminos, debido a que los terminos generales o no especificos tienden a distribuirse mas uniformemente a traves de los documentos de la coleccion, y por lo tanto se tiene mayor ruidk11Ho. Se puede utilizar una funcion inversa del ruido como una funcion posible del valor de termino. A esta funcion se le conoce como senal del termino k y se define como:

$$SENAL_k = \log_2 (FREC\_TOT_k) - RUIDO_k$$

En principio, es posible clasificar las palabras indices extraidas de los documentos de una coleccion en orden decreciente del valor de la senal. Este ordenamiento favorece la distincion entre 1 o 2 documentos especificos (aquel que tenga el termino con mayor senal) del resto de la coleccion. Ademas la importancia, o ponderacion del termino k en el documento i puede ser calculado como una funcion compuesta tomando en cuenta la $FREC_{ik}$ y la $SENAL_k$.

A continuacion se presenta una expresion analoga a la funcion de ponderacion de terminos.

$PONDERACION_{ik}=FREC_{ik}SENAL_k$

En la practica se ve que el valor de la senal no produce un funcionamiento optimo en la recuperacion.

5.3.4 Valor de discriminacion de terminos

El valor de discriminación de términos mide el grado con el cual el uso de un término ayuda a distinguir unos documentos de otros. Si se tiene una colección de documentos donde D_i y D_j representan dos documentos. Una medida de semejanza SEMEJANZA (D_i, D_j) puede utilizarse para representar la semejanza entre documentos. Los valores que puede generar la medición de semejanza son:

0	- Para los documentos que no tienen semejanza.
1	- Para documentos idénticos.
Valores intermedios	- Para los casos de semejanza parcial.

El promedio de semejanza de la colección se puede calcular como:

$$\text{SEMEJANZA PROMEDIO}=\text{CONSTANTE} \sum \text{SEMEJANZA}(D_i,D_j)$$

Donde la CONSTANTE puede ser $\frac{1}{n(n-1)}$

La densidad de espacio puede calcularse construyendo un promedio artificial D como el centroide; en el cual los términos exhiben caracteristicas de frecuencia promedio. La frecuencia promedio del termino k se define como:

$$(\text{FREC PROMEDIO})_k=\frac{1}{n}\sum \text{FREC}_{ik}$$

y la densidad se calcula como la suma de las semejanzas de cada documento con el centroide

$$\text{SEMEJ\_PROM}=\text{CONSTANTE} \sum \text{SEMEJANZA}(D,D_i)$$

Ahora si consideramos la colección de documentos original eliminando el término k de todos lo elementos y representamos la densidad de espacio $(\text{SEMEJ\_PROM})_k$. Si el término k ha sido un término general de alta frecuencia con una distribución moderada de frecuencia uniforme; es decir, que ha aparecido en la descripción de la mayoría de los documentos. Al eliminarse reduce el promedio de semejanza. Cuando al término k se le asigna un peso alto en algunos documentos, pero en otros no; en el momento de eliminarlo, puede aumentar el promedio de semejanza entre documentos.

El valor de discriminación para el término k $VALORDISC_k$ puede calcularse como:

$VALOR\_DISC_k=(SEMEJ\_PROM)_k-SEMEJ\_PROM$

Una vez que se ha calculado el valor de discriminación $VALORDISC_k$ para todos los términos k, es conveniente acomodar los términos en orden decreciente del valor de discriminación $VALORDISC_k$. En la tabla 5.B se muestra para colecciones de documentos que pertenecen a diferentes áreas.

En cada caso se presentan los cinco mejores discriminadores, aquellos cuya eliminacion comprimira el espacio de documentos al máximo. También se presentan los cinco peores discriminadores (palabras comunes que tienen valores de discriminación muy pobres). Como puede observarse los términos de la parte superior son altamente específicos, mientras que los términos de la parte inferior de la tabla son más generales.

Para propósitos de experimentación los términos índice se podrán clasifiacar en tres categorías dependiendo de sus valores de discriminación en:

1) Los discriminadores buenos con un $VALORDISC_k$ positivo, cuya introducción para propósitos de indexación reduce la densidad de espacio.

2) Los discriminadores indifentes con un $VALORDISC_k$ cercano a cero, cuya eliminación o introducción no altera la semejanza entre documentos.

3) Los discriminadores pobres cuya utilización tiende a que los documentos sean más semejantes, los cuales producen un $VALORDISC_k$ negativo.

En la tabla 5.C se muestran las distribuciones de frecuencia para los tres términos típicos.

Como puede verse en la columna de la derecha, cada término ocurre varias veces en diferentes documentos; por lo que el término resulta ser un discriminador pobre.

El termino de discriminación indiferente tiene una frecuencia de documentos baja. Por lo que su asignación deja al documento más o menos sin cambio.

De los datos resultantes de la tabla 5.C y de comparaciones de las características de frecuencia de términos con sus valores de discriminación confirman que los mejores terminos índice son aquellas palabras que tienen una frecuencia media.

Cranfield	MED	Time
1. Panel	1. Marrow	1. Buddhist
2. Flutter	2. Amyloidosis	2. Diem
3. Jet	3. Lymphostasis	3. Lao
4. Cone	4. Hepatitis	4. Arab
5. Separate	5. Hela	5. Viet

a) Los mejores discriminadores

Cranfield	MED	Time
2642. Equate	4717. Clinic	7560. Work
2643. Theo	4718. Children	7561. Lead
2644. Bound	4719. Act	7562. Red
2645. Effect	4720. High	7563. Minister
2646. Solution	4721. Develop	7564. Nation

b) Los peores discriminadores

TABLA 5.B Los mejores y peores discriminadores para las colecciones Cranfield con 424 documentos en aerodinamica; MED con 450 documentos en medicina y Time con 425 documentos en asuntos mundiales.

Numero de ocurrencias del termino k dentro de losdocumentos	Frec. Baja termino cero $VALOR\_DISC_k$	Frec. Media termino + $VALOR\_DISC_k$	Frec. Alta termino - $VALOR\_DISC_k$
1	10	26	221
2	3	13	75
3	3	8	19
4		4	15
5		2	3
6		2	4
7			
8		2	
910			
1115		2	
1620		2	
21+			
frec. de terminos total $FREC\_TOT_k$	25	188	527
Frec. de documentos total	16	61	337

TABLA 5.C Caracteristicas de distribucion de un termino tipico en cada una de las tres categorias de discriminacion, dentro de una coleccion de 450 documentos.

5.4 PROCESO DE INDEXACION AUTOMATICA SIMPLE

La asignación automática de términos índice a los documentos de una colección sigue el siguiente proceso:

1) Empieza con la identificación de las palabras individuales que constituyen los documentos. Pero debe considerarse el tipo de documento que se va a utilizar. Existen los llamados sistemas recuperadores de texto completo, en donde para el proceso de indexación se toma el texto completo de los documentos. Este tipo de sistemas es conveniente cuando se trata de áreas especializadas como medicina o leyes, donde el vocabulario puede ser tan especializado y a donde la presencia de un término en particular tiene connotaciones específicas. Sin embargo, el almacenamiento del texto completo de los documentos en la computadora resulta muy caro. En la práctica basta con analizar extractos del documento, como los títulos y resúmenes.

Experimentalmente se ha observado que el uso de resúmenes adicionales a los títulos proporciona grandes ventajas en la eficacia del recuperador, y que los documentos de texto completo producen muy poca mejoría en áreas específicas.

2) Eliminar las palabras de alta frecuencia, que son discriminadores pobres que no sirven para identificar el contenido del texto, con lo que se logra una compresión del 40 al 50% en las palabras del texto. Los discriminadores pobres pueden agruparse en un diccionario llamado "diccionario negativo" o "antidiccionario" (stop words). A continuación se muestra una lista de este tipo.

A	EL	POR	UNA
ABAJO	ENTRE	SOLO	UNAS
ANTES	HASTA	TAMBIEN	UNOS
ARRIBA	JUNTO	TODOS	Y
DESPUES	PARA	UN	YA

3) Identificación de los términos índice que son buenos descriptores y su asignación a los documentos de la colección. Pero primero resulta conveniente eliminar todos los sufijos y probablemente algunos prefijos; reduciendo las palabras originales a palabras raíz. Esto reduce gran variedad de formas diferentes de palabras como análisis, analista, analizar, analizado, analizando, etc., a la palabra anali. Esta última tendrá una mayor frecuencia de ocurrencia en los textos de los documentos que sus variaciones.

Existen varios algoritmos para eliminar las terminaciones de palabras. La mayoría de estos algoritmos se basan en una lista de sufijos seguida por la eliminación de los sufijos más grandes que igualen cualquier entrada de la lista de sufijos.

A continuación se muestra una lista de sufijos:

abilidad	acidad	dad
able	aco	io
áceas	ajo	ion
áceo	ate	ioso
áceos	ativo	mente
acia	asmo	metro

Cuando se utiliza un algoritmo para eliminar sufijos es importante considerar las excepciones. Por lo tanto la lista de sufijos debe contener a las excepciones.

4) Reconocer las raíces equivalentes que ocurren en los textos y escoger aquellas que se utilicen como términos índice. Para determinar el grado de utilidad de las palabras raíz se pueden utilizar las técnicas de frecuencia.

Los términos (palabras raíz) que tienen valores término altos se pueden asignar a los documentos de la colección con o sin una ponderación. Cuando el modo de indexación es binario, la ocurrencia de un término en un documento será asignada con un peso de 1 sin importar la frecuencia de ocurrencia actual. En un sistema de indexación ponderada, un peso del términno puede utilizarse para reflejar la importancia del término mediante las funciones de ponderación, produciéndose un vector de documento para cada documento D_i.

$D_i = < d_{i1}, d_{i2}, \ldots, d_{it}$

donde cada d_{ij} es el peso asignado al j-ésimo identificador del documento D_i.

Si por ejemplo se tienen los términos (ratón, gato, perro), cuyas ponderaciones son $D_1 = < 0, 3, 5 >$

significa que el documento 1 está identificado por el término RATON con un peso de 0, GATO con un peso de 3 y PERRO con un peso de 5.

El tamano del vector (t) corresponde al número de términos diferentes asignados a la colección completa, y la ponderación de cero corresponde a su vez, a los términos no asignados a un vector de documento dado.

En principio, los términos cuyos factores de importancia no son lo suficientemente altos como para asignarlos a los documentos, pueden ser eliminados del vocabulario. En la figura 5.1 se muestra un sistema de indexación prototipo basado en varios métodos de eliminación.

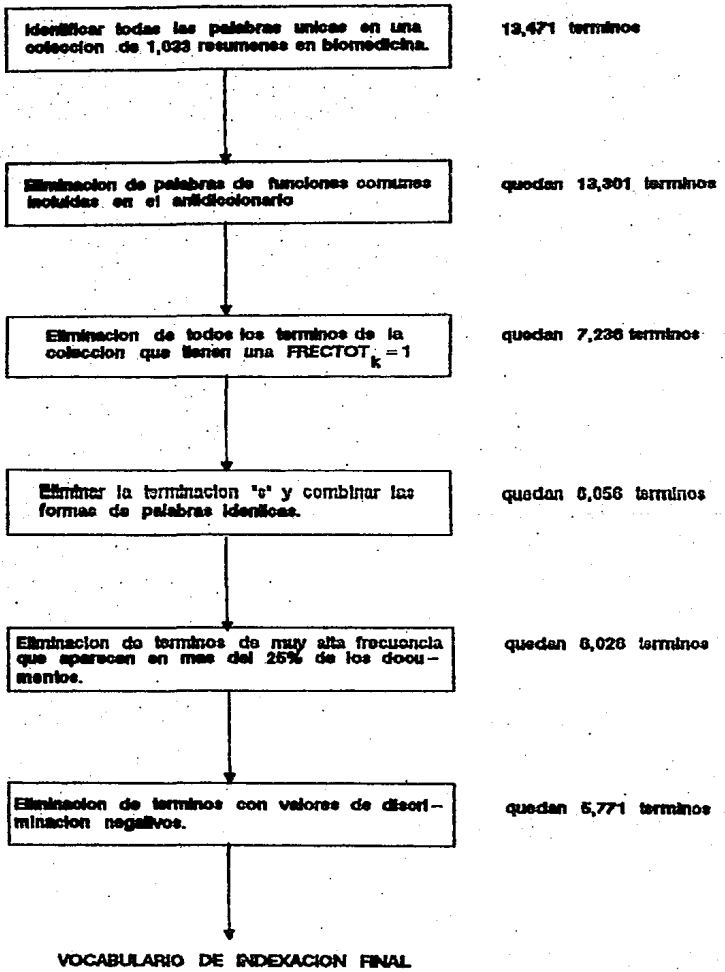


figura 5.1 Algoritmo tipico para la eliminacion de terminos

Los métodos de eliminación de términos deben usarse con precaución para no eliminar términos generales de alta frecuencia que pueden producir pérdidas en el factor de recolección, mientras que la eliminación de algunos términos de baja frecuencia reducen la exhaustividad de la indexación y pueden reducir los factores de recolección y precisión. Es preferible que en vez de eliminar los discriminadores pobres se mejores estos términos para convertirlos en términos con mejores propiedades de discriminación. Esto puede lograrse de diferentes formas usando el contexto y asociaciones de términos.

5.5 ASOCIACION AUTOMATICA DE TERMINOS Y USO DE CONTEXTO

Cuando se extraen palabras del texto de un documento no siempre funcionan efectivamente como términos índice, ya que los terminos de alta frecuencia aparecen en la mayoria de los documentos de una coleccion y los terminos de baja frecuencia rara vez se presentan. Por lo que se requiere de transformar dichos terminos en otros que sean capaces de reflejar el contenido del documento. El lenguaje natural cuenta con una variedad de dispositivos para cambiar la especificidad y el enfoque de terminos individuales. Por ejemplo el termino "computadora" tiene un significado mas amplio que el de "microcomputadora", o la frase "sustancia quimica" tiene una interpretacion mas especifica que los terminos "sustancia" o "quimica" por separado. Para lo cual existen una gran variedad de diccionarios para situaciones de indexacion convencional que permiten al indexador manual escoger los terminos mas generales, mas especificos o terminos relacionados. El proceso explicado anteriormente para eliminar los sufijos es un transformador de terminos especificos (palabras de texto completo) a terminos generales (raices de las palabras).

El mejoramiento de la utilidad de los terminos indice mediante propiedades de discriminacion consiste en utilizar asociaciones entre los terminos para refinar o ampliar su interpretacion.

Muchos tipos de asociacion de terminos pueden incorporarse en sistemas de indexacion automaticos. El primero y mas obvio consiste en imitar el proceso de indexacion manual utilizando un tesauro de terminos.

Actualmente se cuenta con ayudas automaticas que han simplificado la construccion de los tesauros. Por lo que ahora, dada una coleccion de documentos, es fácil producir concordancias automaticamente que exhiban las ocurrencias de todos los terminos en el contexto en el que ocurren, ordenadas alfabeticamente para facilitar el acceso. Por ejemplo, todas las ocurrencias del termino "computadora" seran encontradas en la letra "c", junto con informacion contextual para cada ocurrencia del termino. Esto permite determinar la ubicacion de cada termino dentro de cada clase de tesauro, reuniendo en una clase comun los terminos que ocurren en un conjunto de documentos dado en el mismo contexto.

Un arreglo alfabetico de terminos construido en forma automatica a partir de un conjunto de documentos puede funcionar como una clase de tesauro, y ha sido muy usado en la practica para obtener el acceso a las colecciones de documentos.

Normalmente, los terminos incluidos en dicha lista son las palabras que se presentan en los titulos de documentos. Los productos resultantes se conocen como indices KWIC (KeyWords In Context). Alternativamente se pueden obtener arreglos de terminos afines que se conocen como indices KWAC y KWOC (KeyWord And Context, KeyWord Out of Context). En la tabla 5.D se presentan ejemplos de los indices KWIC y KWAC.

Cuando se cuenta con ayudas como la de los indices KWIC y se aplican los principios para la construccion de tesauros, la tarea de construir la clasificacion de terminos se simplifica, quedando unicamente en forma manual la toma de decisiones para el proceso de agrupacion de terminos.

5.5.1 Uso del tesauro

Un tesauro es una clasificacion de terminos en clases de semejanza (sinonimos o terminos afines) que se utilizan en un area especifica dentro de categorias conocidas como clases de tesauro. Esta clasificacion de terminos puede ser de gran utilidad en el procesamiento del lenguaje por las razones siguientes:

- Un tesauro puede proporcionar un cierto grado de normalizacion del lenguaje, reemplazando el vocabulario de entrada no controlado por las categorias del tesauro controlado.

- Un tesauro se utiliza con frecuencia para generalizar el vocabulario de entrada, anadiendo los identificadores de clase del tesauro mas general. De esta manera se mejora el funciona miento del recuperador.

- Normalmente no hay relacion entre las clases de tesauros. Sin embargo, se pueden construir tesauros multinivel en donde las clases de tesauros individuales se agrupan en conjuntos todavia mas grandes, y asi sucesivamente. Alternativamente se puede tener una relacion jerarquica en las clases de tesauro; es decir, una estructura en donde las clases de nivel bajo (que posiblemente contienen terminos del vocabulario mas especifico) pueden incluirse en clases de nivel mas alto, definido por terminos del vocabulario mas general. Esto permite expandir el vocabulario utilizado en un contexto dado, de lo especifico a lo general o vice-versa. Este tipo de arreglo ha sido utilizado en aplicaciones bibliotecarias durante muchos anos.

TITULO: "SISTEMAS RECUPERADORES DE INFORMACION"

INDICES KWIC

ERADORES DE INFORMACION/SISTEMAS RECUP
ON/SISTEMAS RECUPERADORES DE INFORMACI
INFORMACION/ SISTEMAS RECUPERADORES DE

INDICES KWAC

SISTEMAS
SISTEMAS RECUPERADORES DE INFORMACION 4,222

RECUPERADORES
SISTEMAS RECUPERADORES DE INFORMACION 4,222

INFORMACION
SISTEMAS RECUPERADORES DE INFORMACION 4,222

TABLA 5.D Entradas producidas por los indices KWIC y KWAC

5.5.2 Construccion de tesauros

Sin importar que proceso se utilice en la construccion de un tesauro (construccion manual, semi-manual, o automatica), se presentan los siguientes problemas:

1) Se debe decidir que terminos se van a incluir en el tesauro.

2) Los terminos que se van a incluir deben agruparse apropiadamente (suitably).

Con respecto a la decision de que terminos incluir debe considerarse que el proposito principal de una clasificacion de terminos es mejorar el funcionamiento, y considerando que el modelo de valores de discriminacion indica que los terminos mas importantes son aquellos que tienen una frecuencia de documentos media, seguida por aquellos de baja frecuencia de documentos y por valores con discriminacion cercanos a cero; por lo tanto, se concluye que un tesauro debe incluir una clasificacion de terminos de baja frecuencia de ocurrencia en clases de mayor frecuencia.

Cuando el tesauro se construye en forma manual, la relacion entre los terminos de una clase dada generalmente es linguistica, implicando identidad o semejanza en el significado. Cuando la construccion del tesauro es automatica, la relacion es menos formal y puede estar limitada simplemente a las caracteristicas de co-ocurrencia entre terminos en los documentos de una coleccion.

Los principios para la construccion de un tesauro son:

a) Un tesauro solo debe incluir aquellos terminos que sean de interes para la identificacion del tema en una area dada (por ejemplo un termino como "mano" debe utilizarse en un tesauro de medicina, pero no debe incluirse si su frecuencia de ocurrencia es causada por la expresion "echar mano de " o "mano de obra").

b) Los terminos ambiguos deben codificarse solo para aquellos sentidos que sean importantes en la coleccion del documento que se esta considerando. Por ejemplo cuando se habla del clavo para clavar, o del clavo, especie para cocinar. O cuando se habla de claro de claridad, o de la expresion claro.

c) Para asegurar que la probabilidad de producir una comparacion entre preguntas y documentos sea aproximadamente igual para todas las clases de tesauro, entonces cada clase de tesauro debe incluir terminos de aproximadamente igual frecuencia; ademas, la frecuencia total de ocurrencias debe ser aproximadamente la misma para cada clase.

Si estas caracteristicas de frecuencia se violan grandemente - por ejemplo, si un termino de alta frecuencia como "computadora" se introduce en la misma clase que un termino mas especifico como "minicomputadora" - Entonces las preguntas sobre temas especificos produciran respuestas generales de manera que la precision de la busqueda se reduce.

d) El tamano total del tesauro puede estar relacionado con el tamano de la coleccion del documento utilizado para aplicar la estructura del nivel de acceso, esto implica que el numero de terminos (entradas al tesauro) utilizados para cada documento debe ser aproximadamente de un treintavo del tamano del documento.

e) Siempre que sea posible se deben eliminar los terminos con valores de discriminacion negativas, aun cuando la restriccion del tamano no deba seguir inmediatamente a la eliminacion de todos los no discriminadores de alta frecuencia, lo ultimo es relegado a las clases de tesauro de su propiedad porque su clasificacion junto con terminos de baja frecuencia produce salidas de baja precision.

En el procesamiento de documentos se han utilizado muchos diccionarios de terminos. Dentro de los mas conocidos estan los diccionarios palabra-tema que se obtienen tomando el vocabulario de entrada y excluyendo los sufijos o prefijos. Donde para cada palabra-tema que contiene todas las formas de la palabra completa que pertenecen a un tema se crea una clase de diccionarios.

Actualmente se cuenta con muchas facilidades para construir un tesauro. De esta manera, dada una coleccion de documentos es facil producir automaticamente concordancias (indice KWIC) que muestran todos los terminos del contexto en orden alfabetico. El cual es semejante a un indice que hace posible mantener para cada termino en el vocabulario varios significados que se pueden aplicar en un documento en particular. Ademas se puede producir una clase para cada termino de acuerdo a su valor. Por ejemplo en orden decreciente del valor de discriminacion. Esto identifica a aquellos terminos que aparecen como importantes y permite la reduccion del vocabulario a un tamano deseado. Finalmente se pueden obtener caracteristicas de frecuencia, tales como la distribucion de frecuencia de cada termino a traves de los documentos de una coleccion.

Para definir el estado de la clasificacion inicial se tiene que:

a) Clase de termino puede definirse como el conjunto de terminos asignados a un documento particular o conjunto de documentos; esto genera un numero de clases de terminos iniciales iguales al numero de documentos utilizados como conjuntos de inicio.

b) Clase de termino tambien se define como los terminos contenidos en el conjunto de documentos recuperados en respuesta a ciertas preguntas del usuario, aqui el numero de clases de termino iniciales es igual al numero de preguntas iniciales del usuario.

5.5.3 Evaluacion y mantenimiento de tesauros

A traves de los anos se han acumulado muchos resultados para evaluar la eficacia de los tesauros producidos en forma manual o automatica para la recuperacion de informacion.

No siempre se ha podido demostrar la superioridad de los tesauros construidos automaticamente sobre los construidos en forma manual; sin embargo, es evidente que la normalizacion del vocabulario incorporado en los sistemas de clasificacion de terminos en la mayoria de los recuperadores son beneficos.

El nivel de funcionamiento correspondiente al uso de un tesauro producido en forma manual o automatica puede ser aproximado utilizando otros dispositivos mas baratos de implementar. Por ejemplo, el uso de palabras tema ponderadas seguidas por la eliminacion de los terminos con valores de discriminacion pobres pueden producir un funcionamiento equivalente al que se obtiene con un tesauro.

El tesauro tiene una funcion diferente a la del sistema de terminos ponderados, porque el primero actua principalmente como un dispositivo mejorador y el ultimo mejora la precision.

Una desventaja inherente al uso del tesauro es la necesidad de mantenerlo. Presentandose los siguientes problemas:

El tesauro puede requerir una renovacion como resultado de una interaccion entre el usuario y el sistema, debido a que se hacen nuevas preguntas para las cuales el tesauro actual puede resultar inadecuado, o porque aparecen nuevos intereses, o una nueva poblacion de usuarios, o porque se introducen nuevos terminos al vocabulario.

Existe un sistema que mantiene al tesauro en caso de que la coleccion aumente. Cuando se introducen documentos nuevos a una coleccion dada, el tesauro se puede actualizar utilizando cualquiera de las 4 estrategias siguientes:

a) El tesauro original puede quedar sin cambio y utilizarse para la coleccion expandida.

b) Los terminos nuevos pueden colocarse en las categorias de tesauro existentes.

c) Los terminos nuevos pueden colocarse por separado en clases nuevas.

d) El tesauro puede reestructurarse completamente, generando una clasificacion de terminos del vocabulario actualizado.

La cuarta estrategia no es viable, debido a que resulta muy costosa. Cuando un tesauro aumenta al doble de tamano, no se puede tomar una eleccion clara entre los procedimientos de actualizacion (b) y (c), donde la coleccion actualizada utiliza el tesauro original y los tesauros nuevos.
Cuando se utiliza el tesauro original (procedimiento (a), el resultado indica una perdida del 4% en el funcionamiento.

Finalmente el procedimiento para mantener un tesauro debe basarse en los resultados obtenidos de pruebas que se han seguido sobre varios periodos de tiempo.

5.5.4 Construccion de frases termino

El factor de recoleccion de una busqueda puede mejorarse, generalizando los terminos utilizados en las especificaciones tanto de las preguntas como de los documentos y anadiendo nuevos terminos asociados. A su vez el factor de precision puede mejorarse utilizando terminos especificos o combinaciones de terminos.

Para generar combinaciones de terminos o frases, se utilizan dos o mas terminos $TERM_K$ y $TERM_H$ y se reemplazan por una frase ($FRASE_{KH}$). Por ejemplo, los terminos computadora y programa pueden reemplazarse por "programa de computadora" o "programacion de computadoras".

Dado que la frase termino tiene una interpretacion mas especifica que los componentes individuales de la frase, la frecuencia de ocurrencia de la $FRASE_{KH}$ en la coleccion del documento sera menor que la de los terminos $TERM_K$ o $TERM_H$. Por lo que, los mejores terminos frase deben contener terminos cuya frecuencia de ocurrencia como union (frase) en la coleccion sea mayor que la esperada, dada la frecuencia de los terminos individuales. El termino de cohesion de las parejas de terminos se puede definir como:

$$COHESION_{KH} = \frac{TAMANO.FRECUENCIA\ PAR_{KH}}{FRECTOT_K.FRECTOT_H}$$

donde:

$FRECPAR_{KH}$.- Es la frecuencia par total de los terminos $TERM_K$ y $TERM_H$ en la coleccion.

$FRECTOT_K$ y $FRECTOT_H$.- Representan las frecuencias de los terminos individuales en la coleccion.

TAMANO.-Representa un factor relacionado con el tamano del vocabulario de indexacion.

Con la formula anterior es posible escoger los pares de terminos que tengan un factor de cohesion alto.

Para escoger los terminos que pueden funcionar como frase se requiere de seleccionar un contexto amplio en donde se pueda asegurar que los dos terminos co-ocurren siempre que se incluyen en un documento comun. Se puede obtener una mayor precision cuando se restringe que los dos terminos se presenten en la misma oracion, o en la misma oracion separados por K palabras, o en la misma oracion en posiciones adyascentes, o en la misma oracion en posiciones adyascentes preestablecidas. Pero el proceso de deteccion de frases que cumplan con las restricciones es mas caro. Ademas el numero de frases que cumplan con dichas restricciones sera menor.

Los procedimientos estadisticos para el reconocimiento de identificadores frase en muchas ocasiones han resultado inciertos debido a que identifica frases significativas que sintacticamente son incorrectas.

Sin embargo, es importante determinar la posibilidad de utilizar tecnicas linguisticas mas sofisticadas para controlar el proceso de indexacion automatica.

Las aproximaciones utilizando analisis sintactico y/o semantico no han tenido mucho exito, esto se debe en gran parte a la tecnologia inadecuada y al excesivo costo de los procedimientos linguisticos.

A continuacion se describe un proceso de indexacion automatico simple capaz de producir resultados con un alto grado de recuperacion.

1) Las cadenas de palabras que ocurren en los titulos y resumenes de documentos se aislan y se ponderan los terminos mediante los valores de frecuencia inversa de los documentos o de discriminacion.

2) Se identifican tres tipos de terminos.

1- Los que se encuentran en un rango de frecuencia medio, con valores de discriminacion positivos, que se utilizan como terminos indice y sin necesidad de transformarlos.

2- Los terminos generales de alta frecuencia con valores de discriminacion negativos son descartados o incorporados en frases con caracteristicas de baja frecuencia.

3- Los terminos especificos de baja frecuencia con valores de discriminacion cercanos a cero son generalizados mediante las categorias de tesauro. Entonces los identificadores de clase de tesauros se utilizan como terminos indice para la representacion de contenido. En forma esquematica seria como se muestra en la figura 5.2

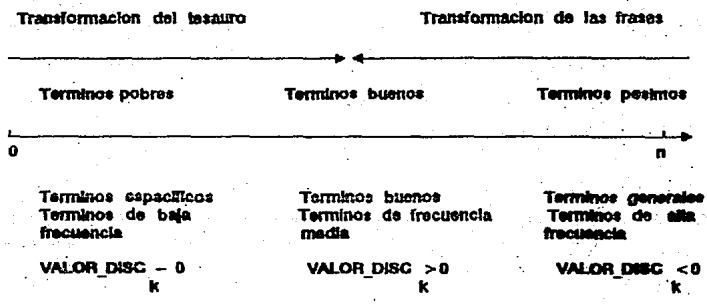


Figura 5.2 **Caracterizacion de terminos en el espectro de frecuencia**

Este proceso automatico producira un gran numero de identificadores de contenido por cada articulo a diferencia del sistema manual en donde se manejan solo unos cuantos.

5.5.5 Extraccion automatica de oraciones

Idealmente se pretende que de un documento en lenguaje natural se construya un buen resumen que informe sobre el contenido del documento original o que le indique al usuario la importancia del documento completo. En realidad, la mayoria de los procesos de extraccion toman del documento original un pequeno conjunto de oraciones que se cree son importantes para representar el contenido.

Los metodos de extraccion utilizados hasta el momento se basan en calculos de la importancia de las palabras y oraciones, que es similar al calculo de ponderacion de terminos en la indexacion automatica.

Los criterios para seleccionar terminos importantes son:

Posicional.- Considera el lugar donde un termino en particular se localiza dentro del documento (titulo, resumen, etc.).

Semantico.- Involucra la relacion entre la palabra u oracion con respecto a otras palabras.

Pragmatico.- Como en el caso de sistemas que consideran altamente importantes a los nombres propios.

Ademas de los criterios anteriores es posible utilizar las ponderaciones estadisticas basadas en las caracteristicas de frecuencia y distribucion del termino.

Cuando se tiene una indicacion significativa de un termino es posible definir la importancia de una frase en funcion de la ponderacion individual de los terminos y de la distancia que exista entre los componentes importantes de una frase mediante:

$$\text{PONDERACION-FRASE} = \frac{1}{2^{\text{Distancia}}} \quad (\text{PESO}_i * \text{PESO}_j)$$

donde:
PESO_i y PESO_j.- Son las ponderaciones correspondientes a los terminos "i" y "j" respectivamente.

Distancia.- Es el numero de palabras entre los terminos.

Una oracion significativa puede ser aquella que contiene un gran numero de grupos de palabras significativas. En la figura 5.3 se muestra un esquema del proceso de extraccion de oraciones.

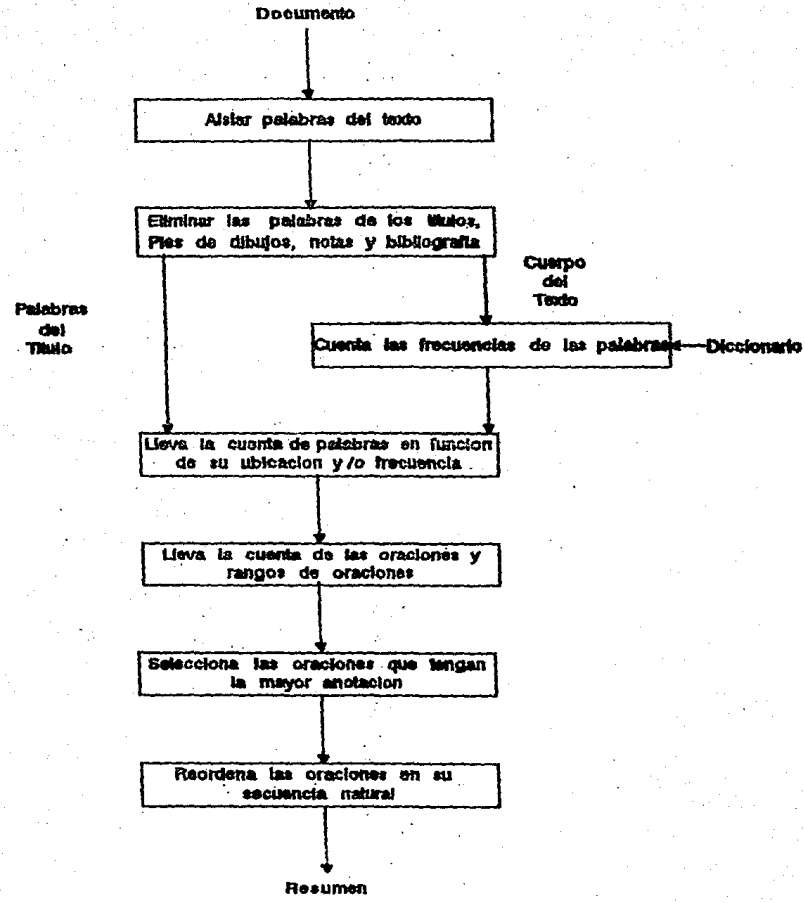


Figura 6.3 Sistema típico para la extraccion de oraciones

Debido a que las caracteristicas de frecuencia puras no son totalmente confiables para los procesos de indexacion o extraccion, existen gran variedad de criterios adicionales para mejorar los resultados en el proceso de extraccion.

Para determinar los criterios de inferencia textual y coherencia sintactica es posible tomar en cuenta los contextos de las palabras y oraciones.

La inferencia contextual utiliza la posicion de las palabras o frases dentro del contexto de un documento como complemento de otro criterio para decidir sobre el rechazo o aceptacion de una oracion. Por lo que cuando las oraciones ocurren bajo ciertos encabezados pueden ser importantes, o cuando las oraciones se encuentran entre signos de interrogacion, normalmente son rechazadas.

El metodo mas utilizado se basa en la presencia de indicadores positivos o negativos del valor de la oracion. Donde la presencia de frases como "Este documento", "La presente investigacion", son consideradas como palabras que van a introducir un informe que debe incluirse en el resumen. Mientras que, las opiniones y referencias a figuras y tablas deben ser eliminadas, las cuales pueden identificarse mediante palabras como "yo creo", "es obvio que", etc. Este metodo requiere de un diccionario que contenga las palabras junto con sus indicaciones del valor sintactico y/o semantico. Resulta mas conveniente incluir las palabras que se van a rechazar debido a que el numero de palabras diferentes es menor aunque su frecuencia de ocurrencia es alta.

No importa que tan sofisticado sea el proceso de extraccion, las oraciones resultantes no son completamente coherentes. Aun cuando las oraciones realmente traten de los topicos apropiados, el flujo de ideas se ve interrumpido entre una oracion y la subsecuente debido a la ausencia de palabras que unan las ideas. Por lo que, se requiere del criterio de coherencia para mitigar en cierto grado las deficiencias inherentes al proceso automatico de extraccion. Para ello se permite que el diccionario contenga palabras o frases que funcionen como enlaces entre oraciones, como ejemplo se listan las siguientes ("presentadas anteriormente", "ellas", "acerca", "afirmo anteriormente", etc) y su presencia en un extracto dado puede utilizarse como una guia para incluir en el resumen las oraciones a las que hace referencia. Cuando los mismos terminos importantes aparecen en oraciones adyascentes, puede presuponerse que existe una relacion entre ambas oraciones, las cuales pueden incluirse o excluirse.

En la realidad no se puede tener un resumen que haya sido generado automaticamente, el cual tenga un estilo bello, y esto se debe a las dificultades linguisticas.

El proceso automatico de hacer resumenes se ha desarrollado y utilizado en menor proporcion que el proceso de indexacion automatico. Los resumenes deben escribirse en un contexto en lenguaje natural y deben obedecer las restricciones normales del estilo. Por otro lado el conjunto de terminos indice no son un obstaculo para las reglas estilisticas.

Actualmente es posible obtener resumenes legibles sin excesiva dificultad, aun cuando no se prevee la perfeccion de estos en un futuro cercano.

CAPITULO 6

Evaluacion de Recuperadores

6.1 INTRODUCCION

El valor de los sistemas recuperadores de informacion dependen de la habilidad de identificar en forma rapida y precisa informacion util, de la facilidad de rechazar articulos ajenos y de la versatilidad de los metodos de busqueda.

Existen dos tipos de pruebas para evaluar un sistema y son: la eficacia y la eficiencia.

La eficacia de un sistema de informacion es la habilidad de proveer al usuario con los servicios de informacion que necesita. Algunas de las componentes de este tipo de evaluacion son:

1) El sistema debe ser capaz de recuperar gran parte de informacion relevante contenida en los archivos y rechazar informacion ajena.
2) El esfuerzo del usuario.
3) Tiempo.
4) Costo.

Los incisos 2), 3) y 4) deben minimizarse en la recuperacion.

Generalmente estas caracteristicas se miden con los valores de recoleccion y precision sobre los resultados de la busqueda, siendo estos dos ultimos componentes de la evaluacion de la eficiencia de la busqueda.

La eficiencia es una medida del costo o tiempo necesario para ejecutar un conjunto de tareas dado.

La viabilidad de un sistema depende de la calidad y del costo de las operaciones; por lo tanto, un proceso de evaluacion completo requiere de la eficiencia y de la eficacia.

Existen muchas razones para evaluar un sistema recuperador de informacion, algunas de las cuales podrian ser:

1) Comparar un sistema con otro.
2) Como cambia la ejecucion del sistema, si se cambia algun componente.
3) Evaluar nuevos componentes del sistema.

Para evaluar un sistema se requiere:

1) Una descripcion detallada del sistema y de sus componentes o un modelo del sistema que va a analizarse.
2) Conjunto de hipotesis a probarse o un prototipo particular contra el cual se va a medir el modelo.
3) Conjunto de criterios que reflejan los objetivos de ejecucion del sistema y una medicion que permita cuantificar los criterios.
4) Metodos para obtener y evaluar datos.

Al final la satisfaccion del usuario dependera de muchos factores como:

1) La facilidad con que el sistema recupera la informacion requerida.
2) Costo del sistema.
3) Esfuerzo requerido por el usuario para hacer una busqueda.
4) Factores humanos como lo serian el diseno de la consola y ubicacion fisica del equipo de busqueda.

6.2 EVALUACION DE LA EFICACIA DEL RECUPERADOR

Para poder analizar como un sistema de informacion es afectado por el medio ambiente y sus operaciones es necesario considerar sus componentes y el medio ambiente. Las componentes son: adquisiciones y politicas de entrada, forma fisica de entrada, organizacion de archivos de busqueda, lenguaje y operacion de indizacion, representacion de los articulos de informacion, analisis de preguntas busquedas y forma de presentacion de la salida.

6.2.1 ADQUISICIONES Y POLITICAS DE ENTRADA

Los parametros relacionados con las politicas de entrada incluyen los retrasos de tiempo que se tienen al introducir documentos en la coleccion, el retraso de tiempo entre recibir un articulo dado y almacenarlo en el archivo, y la cobertura de la coleccion; esto es la cantidad de articulos incluidos actualmente en el archivo.

6.2.2 FORMA FISICA DE ENTRADA

Incluye formato y longitud del documento, titulo, resumen o texto completo. Estos factores afectan directamente a la indizacion, a la busqueda y a la economia del sistema.

6.2.3 ORGANIZACION DE ARCHIVOS DE BUSQUEDA

La organizacion de los archivos de busqueda afectan al proceso de busqueda, al tiempo de respuesta, al esfuerzo de los operadores del sistema, y quiza a la eficacia.

6.2.4 LENGUAJE DE INDEXACION

Consiste de un conjunto de terminos y reglas que existen para asignar estos terminos a los documentos y busquedas requeridas. Existen dos parametros importantes para el lenguaje de indizacion y son: exhaustividad y especificidad. Donde:

- Un lenguaje de indexacion exhaustivo contiene los terminos que cubren todos los temas de la coleccion.
- Un lenguaje de indexacion especifico nunca utiliza un mismo termino para varios temas, sino que el termino es especifico para cada tema.

Durante el proceso de indexacion se escogen los terminos adecuados del lenguaje de indizacion para representar el contenido del documento y se asignan a los articulos de informacion, de acuerdo con las reglas de indizacion establecidas.

La ejecucion de los sistemas recuperadores se mide con los valores de recoleccion y precision, donde:

- El factor de recoleccion mide la habilidad del sistema para recuperar documentos utiles.

- El factor de precision mide la habilidad de rechazar material inutil.

Para obtener una alta recoleccion es util una indexacion exhaustiva junto con un lenguaje de indizacion que provea una variedad de aproximaciones para cubrir el tema dado.

Para asegurar una alta prefcision debe utilizarse un lenguaje de indexacion altamente especifico y los terminos deben tener indicaciones de contenido adicional, como ponderacion de terminos e indicaciones de relacion con otros terminos.

Considerandc que la indexacion se ejecuta en forma manual por personas entrenadas, las variables que afectan a la operacion de indexacion estan relacionadas no solo con la exhaustividad y especifidad de terminos, sino que tambien con la "consistencia tematica", la influencia de la experiencia para indexar en la ejecucion y la exactitud de los terminos asignados.

El analisis de preguntas y la operacion de busqueda es dificil de caracterizar.
La seleccion de terminos del lenguaje de indizacion para hacer la busqueda, formulacion de instrucciones booleanas y la comparacion de requerimientos "contra" la informacion almacenada son tareas dificiles son tareas dificiles.

La operacion de analisis para solicitar un documento es igual a la que se utiliza para hacer una busqueda, debido a que las nociones de exhaustividad y especificidad se utilizan igual en preguntas y documentos. En la practica el proceso de consulta es un poco diferente a la indizacion de documentos, debido a que el usuario se involucra directamente con el primer proceso y no con el segundo. En muchos sistemas la operacion de analisis de requerimientos y consulta se delega a personas entrenadas que utilizan una forma apropiada para introducir los requerimientos del usuario. Caso contrario al de la "introduccion de documentos", en el cual el usuario no interviene para nada.

1 EVALUACION DE LA EFICACIA DEL RECUPERADOR

Los sistemas de informacion pueden evaluarse desde diferentes puntos de vista, ya sea desde el punto de vista del usuario o desde el punto de vista de los operadores y administradores del sistema.
Dentro de los criterios de evaluacion que conciernen a la poblacion de usuarios existen 6 que resultan criticos y son:

1) La RECOLECCION.- Es la habilidad que tiene el sistema para presentar articulos pertinentes.

2) La PRECISION.- Es la habilidad de presentar solo aquellos articulos que sean pertinentes.

3) El ESFUERZO intelectual o fisico que requieren los usuarios para formular preguntas, para elaborar estrategias de busqueda y para visualizar los resultados.

4) El TIEMPO que transcurre desde que el usuario hace una peticion hasta que obtiene resultados.

5) La FORMA en que se presentan los resultados de la busqueda y la influencia que estos tendran en el usuario para la utilizacion del material recuperado.

6) La COBERTURA de la coleccion.- Es el numero de articulos incluidos en el sistema.

En la tabla 6.A se muestran los criterios anteriores junto con los parametros que utiliza cada uno de ellos.

- Los criterios de RECOLECCION y PRECISION son los mas dificiles de medir tanto en concepto como en la practica. Por esta razon seran tratados al final.

- El ESFUERZO del usuario puede medirse en terminos del tiempo requerido para:

    - Formular preguntas
    - Elaborar estrategias de busqueda
    - Interactuar con el sistema
    - Examinar los resultados del sistema

- El TIEMPO de respuesta se mide directamente.

- La FORMA de presentar los resultados puede medirse considerando la facilidad y flexibilidad que tiene el sistema para presentar resultados.

CRITERIOS DEL USUARIO	PARAMETROS RELACIONADOS
Recoleccion y Precision	Exhaustividad de Indexacion (Entre mas exhaustiva, mejor es la recoleccion). Especificidad del lenguaje de indexacion (Entre mas especifico, mejor es la precision). Provisiones en el lenguaje de indexacion para mejorar la recoleccion (reconocimiento de sinonimos, reconocimiento de relaciones entre terminos, etc.). Provisiones en lenguaje de indexacion para mejorar la precision (uso de ponderacion de terminos, uso de frases termino). Habilidad de la poblacion de usuarios para formularrequerimientos debusqueda. Habilidad para planear estrategias de busqueda adecuadas.
Tiempo de respuesta	Tipo de dispositivo de almacenamiento y organizacion del almacenamiento. Tipo de pregunta. Ubicacion del centro de informacion. Proporcion de llegada de preguntas del usuario. Tamano de la coleccion.
Esfuerzo del usuario	Caracteristicas del dispositivo de acceso al sistema. Ubicacion de los dispositivos de acceso y almacenamiento. Disponibilidad del sistema para proveer de ayuda o ayudas disponibles desde el sistema en busquedas no delegadas. Cantidad de material recuperado. Tipo de interaccion con el sistema. Facilidad en la formulacion de requerimientos de busqueda.
Forma de presentacion	Tipo de acceso y dispositivo de despliegue. Tamano de archivo de informacion almacenada. Tipo de salida (titulo, resumen, texto completo).
Cobertura de la coleccion	Tipo de dispositivo de entrada, tipo y tamano del dispositivo de almacenamiento. Facilidad del analisis del contenido (La cobertura puede ser mas amplia cuando el analisisdelcontenidoes simple). Demanda del servicio (la demanda se incrementa con una cobertura mayor).

Tabla 6.A MEDICION DE LOS CRITERIOS ANTERIORES

- La COBERTURA de la coleccion puede presentar problemas cuando se desconoce el numero de articulos de interes que se tienen por tema. Sin embargo, existe una forma facil de hacer una estimacion del numero total de articulos que se tienen en la base de datos y esto es consultando los indices publicados y los volumenes de referencia.

- RECOLECCION y PRECISION

El primer problema que se tiene es la interpretacion de la pertinencia, existiendo dos puntos de vista: el objetivo y el subjetivo.

- El punto de vista objetivo considera que la pertinencia es la correspondencia que existe entre un requerimiento del usuario y un articulo. Esto puede medirse por el grado de cobertura del articulo con respecto a los requerimientos del usuario.

- El punto de vista subjetivo considera tanto el contenido del documento, como el grado de conocimiento que el usuario tiene sobre el tema durante la busqueda. Por lo tanto, la pertinencia depende de la utilidad que cada articulo proporciona al usuario. Entendiendose como un documento pertinente, aquel que trata el tema y con el cual el usuario no esta familiarizado.

Con la pertinencia se podria medir la eficacia del recuperador. El problema seria como establecer los limites entre varios grados de pertinencia y como evaluar la pertinencia.

Por esta razon los observadores han definido a la pertinencia en terminos probabilisticos. En este caso, la pertinencia es una funcion de la probabilidad que analiza la semejanza entre la pregunta del usuario y el vocabulario de los documentos.

6.3 MEDICION DE LOS CRITERIOS DE RECOLECCION Y PRECISION

La recoleccion es la proporcion de material pertinente recuperado.

La precision es la proporcion del material recuperado que es pertinente.

Dependiendo del tipo de necesidades de informacion que tienen los usuarios, algunos prefieren que el sistema tenga un alto grado de recoleccion, y otros que el sistema tenga un alto grado de precision; sin embargo, un buen sistema es quel que presenta un alto grado de recoleccion y precision.
Si se hiciera un corte en la coleccion de documentos, para distinguir los articulos recuperados de los no recuperados, se tendria la figura 6.1

y si ademas separamos los articulos pertinentes de los que no lo son, la recoleccion y la precision estandares se prodrian definir como:

$$R = \frac{\text{Numero de articulos pertinentes recuperados}}{\text{Numero total de articulos pertinentes en la coleccion}}$$

$$P = \frac{\text{Numero de articulos pertinentes recuperados}}{\text{Total de articulos recuperados}}$$

Dado que cada pregunta produce un conjunto desordenado de documentos que son pertinentes o no, es posible obtener una precision simple o una recoleccion simple. Cuando se comparan los factores de recoleccion y precision de dos busquedas i y j en donde la $Recoleccion_i <= Recoleccion_j$ y la $Precision_i <= Precision_j$, se puede observar que los resultados de la busqueda j son mejores que los de la busqueda i. Pero este analisis se complica cuando aparecen casos en donde la $Recoleccion_i > Recoleccion_j$ y la $Precision_i < Precision_j$; y por lo tanto, la decision de cual busqueda es superior, depende del enfoque del usuario.

Es por esto que el usuario debe determinar si prefiere un alto grado de recoleccion, o de precision, y evaluar la importancia de las diferencias entre ambos valores.

En un sistema recuperador de informacion tipico, el grado de recoleccion aumenta cuando el numero de documentos recuperados aumenta; mientras que la precision decrece. De manera que los usuarios que quieran obtener un alto grado de recoleccion deben hacer preguntas generales, mientras que para obtener un alto grado de precision se tienen que hacer busquedas especificas.

Para medir el factor de recoleccion se necesita conocer el numero total de documentos pertinentes en la coleccion con respecto a cada pregunta. Un analisis de este tipo resulta posible cuando el tamano de la coleccion es pequeno, volviendose muy complicado cuando el tamano de la coleccion es grande. Por esta razon para obtener figuras de recoleccion confiables resulta necesario hacer una estimacion del numero total de documentos pertinentes en la coleccion, lo cual puede hacerse mediante tecnicas de muestreo. Otra forma de medir el factor de recoleccion seria procesar una pregunta dada utilizando diferentes estrategias de busqueda y diferentes metodos de recuperacion, asumiendo que todos los documentos pertinentes van a ser recuperados. Despues se combinan los resultados de las busquedas, obteniendose un listado con los documentos pertinentes. Esta lista se obtiene evaluando la pertinencia del listado anterior.

Para calcular los factores de Recoleccion-Precision de una pregunta que tiene un total de 5 documentos pertinentes dentro de una coleccion de 200, se ha elaborado la tabla 6.B que contiene los rangos de articulos importantes en orden decreciente con respecto a la semejanza entre la pregunta y el documento; asi como los factores de recoleccion y precision para una pregunta dada. De manera que si se recuperan 6 documentos de los cuales 4 son pertinentes se tiene que la recoleccion es de 4/5=0.8 y la precision es de 4/6=0.67

RECOLECCION-PRECISION DESPUES DE RECUPERAR N DOCUMENTOS

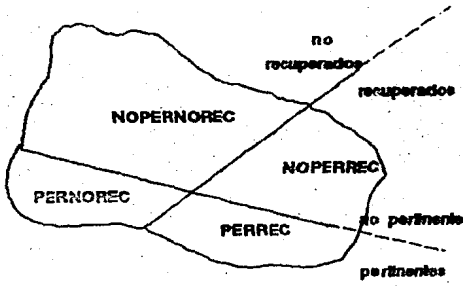
n	Numero de documento (X= pertinente)		Recoleccion	Precision
1	588	X	0.2	1.0
2	589	X	0.4	1.0
3	576		0.4	0.67
4	590	X	0.6	0.75
5	986		0.6	0.60
6	592	X	0.8	0.67
7	984		0.8	0.57
8	988		0.8	0.50
9	578		0.8	0.44
10	985		0.8	0.40
11	103		0.8	0.36
12	591		0.8	0.33
13	772	X	1.0	0.38
14	990		1.0	0.36

Tabla 6.B Resultados de recoleccion y precision de una pregunta dada

La grafica de los valores de recoleccion y precision se tiene en la figura 6.2.

Como puede verse, el valor de recoleccion 0.4 tiene dos valores de precision; sucediendo lo mismo cuando para la precision de 1.0 se tienen dos valores de recoleccion 0.2 y 0.4. Para solucionar este problema se toma el valor de recoleccion mas alto y se traza una linea hacia la izquierda de cada punto pico de precision, quedando la figura 6.3.

Como puede verse en la grafica, para un valor de precison unico existe un valor de recoleccion. Por ejemplo para un valor de recoleccion 0.4 la precison es de 1.0; sin embargo, para un valor de recoleccion de 0.41 la precision es de 0.75.



PERREC : Numero de articulos pertinentes recuperados
NOPERREC : Numero de articulos no pertinentes recuperados
PERNOREC : Numero de articulos pertinentes no recuperados
NOPERNOREC: Numero de articulos no pertinentes no recuperados

FIGURA 6.1 Particion de la coleccion

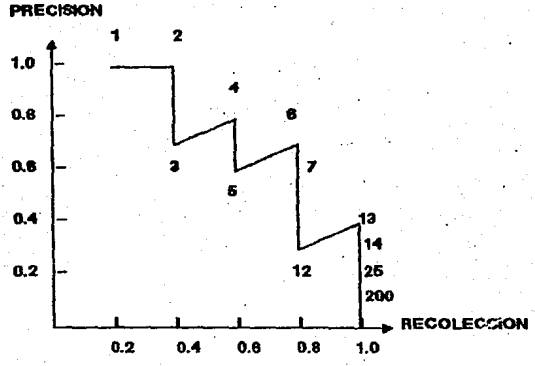


FIGURA 6.2 Grafica de la precision contra la recoleccion para los valores de la tabla 6 – B

El promedio del grado de recoleccion orientado al usuario refleja el funcionamiento que el usuario promedio puede esperar del sistema. Este promedio puede obtenerse a partir de:

$$RECOLECCION_i = \frac{RECPER_i}{RECPER_i + NORECPER_i}$$

$$PRECISION_i = \frac{RECPER_i}{RECPER_i + RECNOPER_i}$$

Esto hace posible que el calculo de la recoleccion y precision permita graficar un curva continua, como se muestra en la figura 6.4.

La parte izquierda de la grafica que es la mas angosta, muestra que las preguntas formuladas fueron especificas, obteniendose unos cuantos documentos& k l 2 H p o r l o t a n t o, la precision alta, mientras que el grado de recoleccion es bajo. Lo contrario se observa en el lado derecho de la curva, que representa lo extenso; es decir, la formulacion de preguntas fue general y por lo tanto se recuperaron un gran numero de documentos, teniendose un alto grado de recoleccion y un bajo grado de precision.

El promedio del nivel de recoleccion orientado al usuario refleja el funcionamiento del sistema esperado por el usuario promedio. Esto puede obtenerse a partir de las siguientes ecuaciones:

$$RECOLECCION_{NR} = \frac{1}{NUM} \frac{RECPER_i}{RECPER_i + NORECPER_i} \quad (1)$$

$$PRECISION_{NR} = \frac{1}{NUM} \frac{RECPER_i}{RECPER_i + RECNOPER_i} \quad (2)$$

El promedio del nivel de documentos orientado al sistema, se obtiene a partir del numero total de articulos pertinentes recuperados por el sistema sobre el numero de preguntas; asi como el numero total de articulos no pertinentes recuperados. El promedio del nivel de documentos se define como:

$$RECOLECCION_{ND} = \frac{RECPER_i}{(RECPER_i + NORECPER_i)} \quad (3)$$

$$PRECISION_{ND} = \frac{RECPER_i}{(RECPER_i + RECNOPER_i)} \quad (4)$$

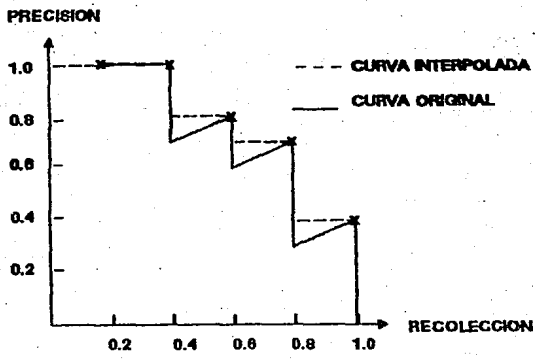


FIGURA 6.3 Curva interpolada de recoleccion – precision para los datos de la tabla 6 – B

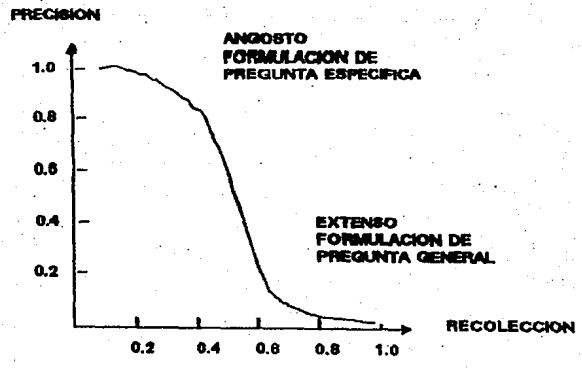


FIGURA 6.4 Grafica tipica de la recoleccion – precision promedio

Los promedios de las ecuaciones (1) y (2) dan igual importancia a cada pregunta; mientra que, en las formulas (3) y (4) los promedios dependen mas dlas preguntas con mas documentos pertinentes que de aquellos con pocos articulos pertinentes.
En la recuperacion de informacion la eleccion del metodo para obtener el promedio depende de, si es mas importante presentar el promedio de los resultados del usuario, (ecs. 1 y 2) o reflejar que sucede al promedio de documentos pertinentes (ecs. 3 y 4), si la funcion de la pregunta no depende del numero de documentos pertinentes, los dos promedios proporcionan resultados semejantes.

Para evaluar el funcionamiento de los sistemas recuperadores de informacion, se pueden utilizar las curvas de recoleccion-precision. Esto se haria calculando los valores de recoleccion y precision de de dos o mas sistemas, o con el mismo sistema operando bajo diferentes condiciones, y se procede a sobreponer las graficas producidas por los sistemas. De esta manera se puede determinar que sistema es superior y por cuanto. Generalmente la curva mas cercana a la esquina superior derecha de la grafica representa el mejor funcionamiento, debido a que es donde se maximizan los valores de recoleccion y precision.

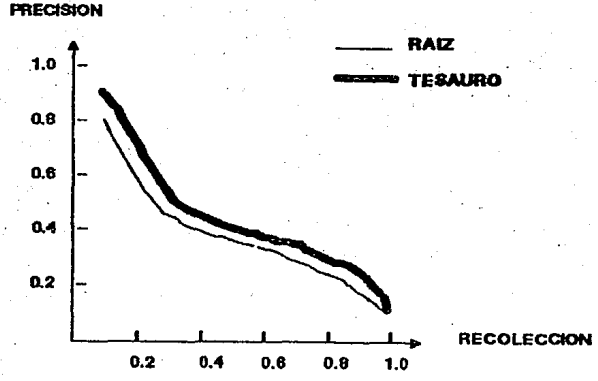
A continuacion se presenta un ejemplo que compara el funcionamiento de dos sistemas de indexacion diferentes para una coleccion de documentos en ciencias bibliotecarias promediando sobre 35 preguntas del usuario.

El proceso de indexacion por raiz extrae las palabras del resumen del documento y las utiliza como terminos indice para representar el contenido del documento.
El proceso de indexacion con tesauro reemplaza las palabras por conceptos que son extraidos de un tesauro, representando clases de terminos relacionados o sinonimos de las raices originales.

Como puede observarse en la figura 6.5, el promedio de precision del tesauro es mejor que el de raiz de un 4 a un 22%. Aunque es dificil juzgar el significado de las diferencias entre dos curvas a traves de los porcentajes de mejoria, es posible construir estadisticas que indiquen si una diferencia dada entre dos promedios es significativa.

RECOLECCION	PROMEDIO DE LA PRECISION		% DE MEJORIA
	RAIZ	TESAURO	
0.1	0.7953	0.8788	10.4
0.2	0.6350	0.7567	19.2
0.3	0.5283	0.6464	22.4
0.4	0.4603	0.5577	21.2
0.5	0.4051	0.4912	21.3
0.6	0.3699	0.4470	20.8
0.7	0.3393	0.3993	15.1
0.8	0.2998	0.3287	9.7
0.9	0.2568	0.2726	6.2
1.0	0.2018	0.2093	3.7

a)



b)

Figura 6.5 Resultados del promedio de la recoleccion – precision para dos metodos de indexacion
a) Valores de recoleccion – precision
b) Grafica de los valores de recoleccion – precision

2 MEDICION DE LA EFICACIA DEL RECUPERADOR

Es conveniente hacer notar que las mediciones de recoleccion y precision normalmente estan ligadas a una coleccion de documentos y a un conjunto de preguntas dado; es decir, cuando se varian las politicas de indexacion, o el lenguaje de indexacion, o la metodologia de busqueda, es importante analizar como estos cambios pueden afectar el funcionamiento del sistema.

Los valores de recoleccion y precision deben utilizarse con cuidado cuando se compara el funcionamiento de dos sistemas completamente diferentes, basados en colecciones de documentos diferentes, conjuntos de preguntas diferentes y diferente poblacion de usuarios, debido a que dichos valores pueden deteriorarse.

Cuando se miden los valores de recoleccion-precision es importante ver el efecto que ocasionan los diferentes tipos de pregunta en la evaluacion de la salida. Existen varios tipos de pregunta, dentro de los cuales se encuentran los encabezamientos de materia cortos, en los cuales el tema se expresa como un pequeno conjunto de caracteres descriptivos del titulo, donde una oracion sencilla o titulo describe apropiadamente el area del tema. (por ejemplo: control termico, estudios de turbulencia, inversion termica), y de texto completo donde se utiliza un parrafo completo para formular una busqueda. Se pueden generar diferentes tipos de preguntas en sistemas donde la formulacion de la busqueda es delegada por el usuario a intermediarios entrenados en busquedas.

Con frecuencia, las busquedas por encabezamientos de materia proporcionan temas generales, mientras que con busquedas de texto completo son mas especificas; sin embargo, no siempre es verdad que la longitud de la busqueda este directamente relacionada con la especificidad. De cualquier manera el sistema se tiene que probar utilizando una mezcla de busquedas reales que reflejen los diferentes tipos que surgen en situaciones funcionales.

Cuando se calculan la precision y la recoleccion se tienen que asignar grados de pertinencia a los documentos de una coleccion dada, asi como elegir que rangos de documento se van a tener a la salida. Cuando se toman en cuenta las dos consideraciones anteriores, se tiene que varios documentos pueden ser la respuesta adecuada para una busqueda y son colocados en orden consecutivo a la salida. Sin embargo, como el orden afecta a la evaluacion de los valores de precision y recoleccion, existen tecnicas que permiten compensar el orden arbitrario de los documentos semejantes. Una de estas tecnicas consiste en asignar a cada uno de los documentos un grado de pertinencia igual al grado promedio de este conjunto de documentos. Pero en la practica resulta mas dificil manejar muchos grados de pertinencia, que simplemente decidir entre los documentos pertinentes, parcialmente pertinentes y no pertinentes, lo cual vuelve al sistema muy impreciso.

Asi como se tiene un valor que refleja el funcionamiento de los articulos pertinentes (valor de recoleccion); es necesario tener una forma de medir el funcionamiento de los documentos no pertinentes, y a este se le conoce como rechazo y se define:

$$\text{RECHAZO} = \frac{\text{RECNOPER}}{\text{RECNOPER} + \text{NORECNOPER}}$$

$$\text{RECHAZO} = \frac{\text{Numero de articulos no pertinentes recuperados}}{\text{Numero total de articulos no pertinentes en la coleccion}}$$

Expresado en terminos probabilisticos, la recoleccion y el rechazo son:

La recoleccion es la probabilidad de recuperar un documento pertinente.

El rechazo es la probabilidad de recuperar un documento no pertinente.

Por lo tanto, un sistema recuperador de informacion eficaz es aquel que presenta un valor de recoleccion maximo y un valor de rechazo minimo.

Los factores de recoleccion, precision y rechazo de un recuperador que se encuentra en un medio ambiente normal son dependientes de un factor de generalidad, definido como el numero promedio de articulos pertinentes por pregunta incluidos en la coleccion. A continuacion se presenta una tabla que muestra las mediciones tipicas para la evaluacion de recuperadores:

SIMBOLO	CRITERIO DE EVALUACION	FORMULA	EXPLICACION
R	Recoleccion	$\frac{\text{RECPER}}{\text{RECPER} + \text{NORECPER}}$	Proporcion de articulos pertinentes que son recuperados.
P	Precision	$\frac{\text{RECPER}}{\text{RECPER} + \text{RECNOPER}}$	Proporcion de articulos recuperados que son pertinentes.
F	Rechazo	$\frac{\text{RECNOPER}}{\text{RECNOPER} + \text{NORECNOPER}}$	Proporcion de articulos no pertinentes que son recuperados.
G	Generalidad	$\frac{\text{RECPER} + \text{NORECPER}}{\text{TOTAL}}$	Proporcion de articulos pertinentes por pregunta.

TOTAL=RECPER+RECNOPER+NORECPER+NORECNOPER

Como puede observarse, cualquiera de los valores R, P, F y G pueden obtenerse a partir de 3 de ellos. Por ejemplo, la precision puede obtenerse a partir de la recoleccion, el rechazo y la generalidad como:

$$P = \frac{R.G}{(R.G) + F(1-G)}$$

Es posible utilizar cualquiera de los dos pares, recoleccion-precision y recoleccion-rechazo para evaluar la eficacia. Ambos pueden responder a diferentes necesidades en situaciones de recuperacion.

Se dice que la pareja recoleccion-precision esta orientada al usuario, debido a que busca optimizar la recuperacion de articulos pertinentes; mientras que la pareja recoleccion-rechazo esta enfocada al sistema, debido a que es un indicador de que tan bueno es el sistema para rechazar los documentos no pertinentes en funcion del tamano de la coleccion.

Todas las mediciones anteriores se han basado en juicios de pertinencia objetiva; es decir, que son independientes del conocimiento anterior que el usuario puede tener sobre el tema. A continuacion se presentan otras formas de medir la eficacia del sistema a partir de la pertinencia subjetiva:

1) La proporcion de novedad.- Es la proporcion de articulos pertinentes recuperados y juzgados por usuarios que no han tenido un conocimiento anterior al de la recepcion del resultado de la busqueda.

2) La proporcion de cobertura.- Es la proporcion de articulos pertinentes recuperados del total pertinente conocido por los usuarios antes de la busqueda.

3) La recoleccion explorada.- Es la pertinencia total examinada por los usuarios despues de una busqueda, dividida entre la pertinencia total de los usuarios que habrian querido revisarla.

Algunos observadores rechazan el uso de la tabla de contingencia (que se muestra a continuacion) para construir parametros que reflejen la eficacia del recuperador, y prefieren utilizar las propiedades de la medicion ideal de la eficacia que se postulan a continuacion:

1) La medicion debe ser capaz de reflejar la eficacia del recuperador de manera independiente de otros criterios, como seria el costo.
2) La medicion debe ser independiente del numero de documentos recuperados en una busqueda en particular.
3) La medicion debe expresarse como un simple numero (en vez de 2, como en el caso de la recoleccion-precision) que pueda ser puesto en escala para obtener valores absolutos o relativos.

TABLA DE CONTINGENCIA

	PERTINENTES	NO PERTINENTES	
Recuperados	RECPER	RECNOPER	RECPER + RECNOPER
No recuperados	NORECPER	NORECNOPER	NORECPER + NORECNOPER
	RECPER + NORECPER	RECNOPER + NORECNOPER	RECPER + RECNOPER + NORECPER+NORECNOPER

La mas conocida de estas mediciones de valores unicos es la medicion E de Swets, y para medirla se toman dos poblaciones POB_1 y POB_2 asociadas con los documentos pertinentes y no pertinentes con respecto a una pregunta. Se utiliza el parametro r para representar algunas caracteristicas medibles, como seria la semejanza pregunta-documento de cada documento. A partir de las funciones de densidad de probabilidad $FUNC_1(r)$ Y $FUNC_2(r)$, de las medias $MEDIA_1$ y $MEDIA_2$, y de las varianzas VAR_1 y VAR_2, se puede analizar el comportamiento de la caracteristica r en las dos poblaciones.
La funcion $FUNC_1(r)$ representa la probabilidad de que un documento de la POB_1 tenga el valor r.
En la figura 6.6 se muestra una grafica tipica de esta funcion de densidad de probabilidad.

Si se elige un valor r=c, este es en realidad un corte, donde para cualquier articulo que cumpla con FUNC(r)>=C es recuperado.

Debido a que las funciones $FUNC_1(r)$ y $FUNC_2(r)$ estan asociadas con las poblaciones de documentos pertinentes y no pertinentes respectivamente, las areas bajo las curvas de densidad a la derecha de r=c representan respectivamente la proporcion de documentos pertinentes y no pertinentes para los cuales FUNC(r)>=C . Siendo la primera medicion la recoleccion, y la segunda el rechazo.

Si se grafican los procentajes de las poblaciones POB_1 y POB_2 a la derecha de C, variando C, se obtiene una curva de las caracteristicas de operacion, como la que se presenta en la figura 6.7.

Cuando dos poblaciones son semejantes a la caracteristica r, se obtiene como caracteristica de operacion una recta que cruza a la grafica en diagonal. Cuando las dos poblaciones son totalmente diferentes a la caracteristica r se tiene una curva como la mostrada en la grafica anterior.

Se han graficado las caracteristicas de operacion para un gran numero de sistemas recuperadores de informacion y se ha llegado a concluir que las funciones de densidad de probabilidad de la recoleccion y el rechazo con respecto al parametro r son normales.

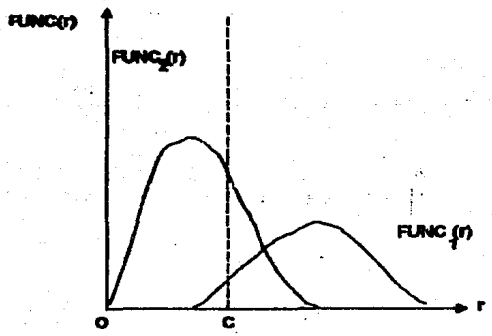


Figura 6.6 Funciones de densidad de probabilidad de las poblaciones POB1 y POB2

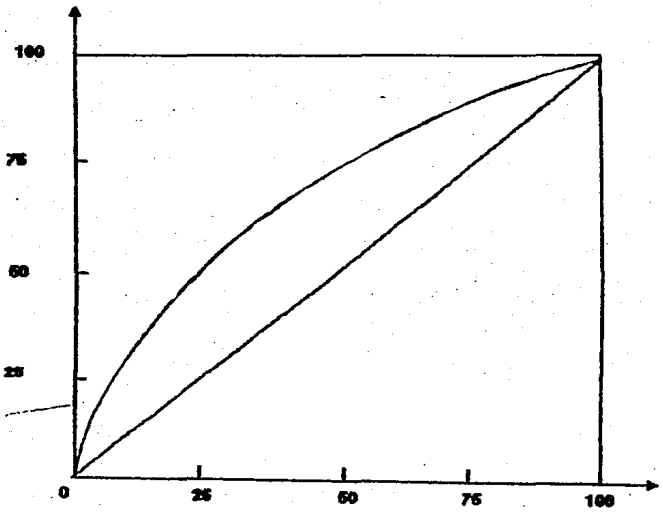


Figura 6.7 Curva de las caracteristicas de operacion

El funcionamiento de la recoleccion-rechazo puede representarse analizando la posicion de la linea. Por ejemplo, si se tienen dos lineas caracteristicas de operacion tipicas, llamadas A y B. La medicion E de Swets se define como:

E= 2 . DIST

Donde DIST es la distancia entre O y la curva caracteristica de operacion, a lo largo de la linea O a R (Vea la figura 6.8). Cuando el angulo de la curva caracteristica de operacion es el mismo que el de la diagonal como en el caso de la linea A; es decir, la pendiente de la curva es igual a 1, entonces el valor de E representa la eficacia del funcionamiento. Cuando el angulo de la curva caracteristica de operacion no es el mismo que el de la diagonal como en la linea B, entonces es necesario presentar la pendiente de la curva caracteristica de operacion; asi como el valor de E.

La ventaja principal de las mediciones E de Swets, PENDIENTE es que ambas se derivan de una teoria estadistica muy conocida y aceptada.

La desventaja es de que a diferencia de la recoleccion y precision, las mediciones (E de Swets, PENDIENTE) no son faciles de leer por la poblacion de usuarios.

Es importante considerar que el valor E de Swets no puede obtenerse a partir de un recuperador sencillo que distingue el umbral entre los los articulos recuperados y no recuperados, que es lo que normalmente se obtiene; sin embargo, a partir de estos datos es posible calcular la recoleccion-precision.

Se han desarrollado varias mediciones globales basadas en teorias establecidas, especialmente en teorias probabilisticas y teorias de informacion, pero en la practica no han recibido consideracion alguna.

Estas mediciones generalmente combinan la recoleccion y precision en una expresion simple. A continuacion se presenta una funcion basada en consideraciones de teoria de medicion, que utiliza el parametro , el cual hace posible asociar grados de importancia a los componentes de recoleccion y precision.

$$E = 1 - \frac{1}{(1/\text{PRECISION}) + (1- \quad)(1/\text{RECOLECCION})}$$

Existen ademas mediciones valuadas simples que tambien utilizan las diferencias entre los rangos reales de los articulos pertinentes recuperados, y los rangos ideales, donde todos los articulos pertinentes son recuperados antes de cualquier articulo no pertinente, o un sistema aleatorio donde los articulos pertinentes son esparcidos de manera aleatoria de entre los no pertinentes. Dentro de este tipo de mediciones se encuentran "La longitud de busqueda esperada" y la proporcion deslizante.

La longitud de busqueda es el promedio de los articulos no pertinentes que tienen que ser explorados por el usuario antes de alcanzar el numero deseado de articulos pertinentes.

La longitud de busqueda esperada (ESP) para la pregunta (PREGUNTA) sera:

$$ESP(PREGUNTA) = PREVNOPER + \frac{NOPER.NUM}{NOPER+1}$$

donde:

PREVNOPER.- Es el numero de documentos no pertinentes en todos los conjuntos precediendo a aquel donde la busqueda termina.

Considerando que hay articulos pertinentes PER en el conjunto final y que estos son puestos en intervalos iguales entre los documentos no pertinentes NOPER en ese conjunto, entonces PER+1 subsecuencias de documentos no pertinentes seran creados conteniendo (NOPER/PER)+1 documentos no pertinentes cada una.

NUM.- Es el n-simo articulo pertinente del ultimo conjunto en donde se satisface el requerimiento.

La longitud de busqueda aleatoria esperada se obtiene esparciendo aleatoriamente todos los documentos pertinentes TODOSPER de una pregunta (PREGUNTA) a traves de los articulos ajenos AJENOS, y esto se define como:

$$AESP(PREGUNTA) = \frac{DESEADOS.AJENOS}{TODOSPER + 1}$$

donde:
DESEADOS.- Es el numero total de articulos pertinentes deseado.

A partir de las mediciones obtenidas ESP y AESP se puede obtener una medicion util.

$$REDUCCION\ ESP(PREGUNTA) = \frac{AESP(PREGUNTA) - ESP(PREGUNTA)}{AESP(PREGUNTA)}$$

$$= 1 - \frac{PREVNOPER + (NOPER * NUM)/(PER+1)}{(DESEADOS * AJENOS)/(TODOSPER - 1)}$$

La medicion de la proporcion deslizante esta basada en la comparacion entre la salida de un sistema recuperador real y la salida de un sistema ideal, en donde los articulos son clasificados en orden decreciente de pertinencia. Este modelo es mas complicado que el anterior debido a que permite asignarle un peso de pertinencia numerica a los documentos, reemplazando asi la asignacion de pertinencia binaria (pertinentes y no pertinentes).

La medicion de la proporcion deslizante DESLIZANTE(NUM) para un corte, puede definirse como:

$$\text{DESLIZANTE(NUM)} = \frac{\text{DPESO REAL(NUM)}}{\text{DPESO IDEAL (NUM)}}$$

donde DPESO REAL(NUM) y DPESO IDEAL(NUM) son la suma de los pesos de la pertinencia de todos los articulos recuperados hasta el rango NUM en los sistemas actuales e ideales respectivamente. A continuacion se presenta una tabla que contiene los calculos del DESLIZANTE(NUM) para 5 documentos que tienen rangos diferentes.

Rango de recuperacion	1	2	3	4	5
Numero de documento	3	4	5	1	2
Pesos de la pertinencia $PESO_i^{real}$	10	0	8	5	2
DPESO REAL(NUM)= $PESO_i^{real}$	10	10	18	23	25
Numero de documento	3	5	1	2	4
Pesos de la pertinencia $PESO_i^{ideal}$	10	8	5	2	0
DPESO IDEAL(NUM)= $PESO_i^{ideal}$	10	18	23	25	25
$\text{DESLIZANTE(NUM)} = \frac{\text{DPESO REAL(NUM)}}{\text{DPESO IDEAL(NUM)}}$	1	0.55	0.78	0.92	1

El valor de DESLIZANTE(NUM) muestra la habilidad deH s i s t ema actual para aproximar la capacidad de recuperacion a la de un sistema ideal.

En el limite, como NUM se aproxima al numero total de documentos de la coleccion

PESO REAL(NUM) llega a ser igual a PESO IDEAL(NUM)

===>DESLIZANTE(NUM)=1

La proporcion deslizante puede utilizarse en sistemas de recuperacion binaria, donde a los articulos pertinentes se les asigna el valor de 1 y a los no pertinentes 0. En este caso, la proporcion aproxima la recoleccion normalizada y la longitud de busqueda esperada.

Considere que tiene 3 articulos en el conjunto 1, y 5 articulos en cada uno de los conjuntos 2 y 3, como se muestra en la figura 6.9 . El conjunto 1 se recupera antes que el 2 y este a su vez

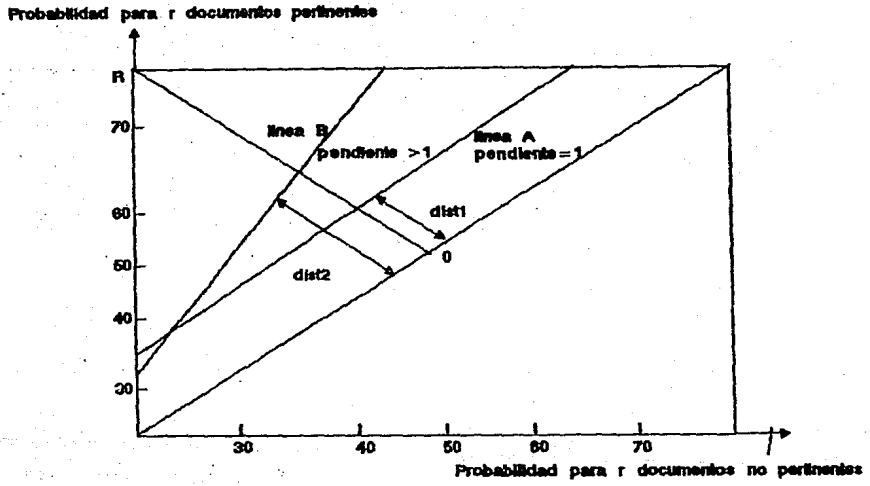
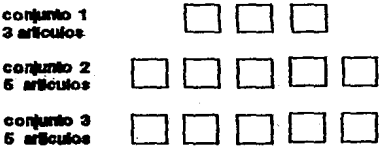


Figura 6.8 Caracteristicas de operacion en escalas de probabilidad normales



a)

Numero de articulos requeridos	Numero de conjuntos a buscar	Longitud de la busqueda	Longitud promedio de la busqueda
1	1	0,1,o 2	1/3*0 + 1/3*1 + 1/3*2 = 1
6	3	3,4,5,o 6	4/10*3 + 3/10*4 + 2/10*5 + 1/10*6 = 4

b)

Figura 6.9 Longitud de busqueda promedio
a) Salida ordenada parcialmente
b) Calculo de la longitud de busqueda promedio

antes que el 3. Cada uno de estos conjuntos contiene documentos que no estan clasificados en orden; por lo tanto, durante la busqueda se tendran que examinar 4 documentos no pertinentes en promedio, para encontrar 6 documentos pertinentes.

En este caso, si solo hay un articulo pertinente en el conjunto 1 y 4 en el conjunto 2, resulta necesario buscar en el tercer conjunto para obtener un articulo pertinente y con este tener un total de 6. (figura 6.9 b)

MEDICION DE LA UTILIDAD

Hasta este momento no se habia considerado el costo de recuperar la informacion. Considerando el costo y/o los parametros de valor es posible planear estrategias de evaluacion de recuperadores basada en una extension de la tabla de contingencia estandar, como la que se muestra a continuacion.

	PERTINENTES	NO PERTINENTES	
Recuperados	V_1(RECPER)	C_1(RECNOPER)	RECPER+RECNOPER
No recuperados	C_2(NORECPER)	V_2(NORECNOPER)	NORECPER+NORECNOPER
	RECPER+NORECPER	RECNOPER+NORECNOPER	N

Ademas de los 4 parametros ya conocidos RECPER, RECNOPER, NORECPER y NORECNOPER.

$VALOR_1$.- Es el valor que se le asigno a cada articulo pertinente recuperado.
$VALOR_2$.- Es el valor asignado a cada articulo no pertinente rechazado.

A su vez $COSTO_1$ y $COSTO_2$ estan asociados con los articulos no pertinentes recuperados y con los articulos pertinentes faltantes.

Si pudiera expresarse mediante una variabe la medicion de semejanza entre un documento y una pregunta, se tendria:

VAR=FUNC(PREGUNTA,DOC)==>La utilidad de un documento pertinente dado (CONJDOC) con respecto a una pregunta en el umbral de recuperacion VAR=UMBRAL, puede expresarse como:

(Ecuacion 5)

$$UTIL(CONJDOC,PREGUNTA,UMBRAL)=VALOR_1.RECPER - COSTO_1*RECNOPER - COSTO_2*NORECPER + VALOR_2*NORECNOPER$$

o

(Ecuacion 6)

UTIL(CONJDOC,PREGUNTA,UMBRAL)=
$VALOR_1$ * N Prob{DOC es pertinente y VAR >= UMBRAL}
$-COSTO_1$ * N Prob{DOC no es pertinente y VAR >= UMBRAL}
$-COSTO_2$ * N Prob{DOC es pertinente y VAR < UMBRAL}
$+VALOR_2$ * N Prob{DOC no es pertinente y VAR < UMBRAL}

Donde N es el numero total de documentos en el sistema.
La expresion (6) puede transformarse utilizando las funciones de densidad $FUNC_1$(VAR) y $FUNC_2$(VAR) (fig. 5-7), donde el area bajo las curvas de densidad a la derecha de un umbral dado representan la probabilidad de que la variable VAR tenga un valor mayor al umbral, dado que los documentos son pertinentes o no respectivamente. Por sustitucion de integrales en la expresion (6) se obtiene un umbral de recuperacion util para el cual la utilidad del sistema es positiva.

6.4 EVALUACION DEL COSTO Y EFICIENCIA DEL SISTEMA

El analisis de la eficiencia no se ha desarrollado tanto como el analisis de la eficacia, debido a que resulta dificil aproximar el costo en funcion del tiempo, esfuerzo y gastos; asi como la imposibilidad en la mayoria de los casos de asegurar el valor del mejoramiento de los servicios de informacion prestados y los beneficios derivables de esta. Ademas cuando se analiza el costo de un recuperador de informacion, no se puede establecer una comparacion con otras evaluaciones de costo, debido a que la evaluacion de la eficacia involucra un gran numero de factores intangibles.

Sin embargo, es necesario considerar el analisis de costo, y evaluar la eficiencia potencial de un sistema de informacion antes de instalarlo.

Es importante hacer una distincion entre el analisis costo-eficacia y el analisis costo-beneficio.

El analisis costo-eficacia.- Consiste en encontrar los medios mas baratos para llevar a cabo un conjunto de operaciones, u obtener el valor maximo de gastos dado.

El analisis costo-beneficio.- Consiste en hacer una comparacion entre los costos de operaciones individuales y los beneficios que se derivan de ellos.

Los costos de un sistema se pueden dividir en: Costos del desarrollo inicial; que incluyen diseno, prueba y evaluacion.

Costos de operacion.- Estos son variables y dependen de las tareas ejecutadas, el personal utilizado y la cantidad de equipo requerido.

Costos fijos como la renta, impuestos, etc.

Los beneficios que se pueden obtener de un sistema de informacion pueden estar relacionados con los costos reducidos o con el incremento de la productividad.

En un sistema recuperador de informacion en donde se conocen el volumen de operaciones como numero, tamano, y costo de los documentos; asi como, el numero promedio de preguntas, se presentan por una parte las alternativas basicas y los procesos internos relacionados con la entrada, y las operaciones de indexacion de documentos; y por la otra, la busqueda de informacion y transacciones de salida.

6.4.1 ANALISIS DEL COSTO

El analisis del costo de un sistema de informacion puede llevarse a cabo basicamente por dos aproximaciones distintas. Una consiste en analizar cuidadosamente cada uno de los pasos que incluye un proceso y efectuar mediciones directas de las diferentes cantidades que entran en el costo para un medio ambiente de operacion dado. La otra consiste en generar un modelo abstracto del sistema en estudio, e indagar la eficiencia del sistema, llevando a cabo estudios de simulacion. En ambos casos se deben tomar en cuenta todos los costos, incluyendo los costos de desarrollo, costos de operacion y costos fijos.

Un analisis tipico de eficiencia basado en las mediciones iniciales de costo, tiempo y volumen de operaciones, empieza con una descripcion formal del sistema, incluyendo la interrelacion entre procesos, y los parametros basicos como tamanos de achivo, proporciones de entrada y salida; asi como otras caracteristicas.

En bibliotecas y Centros de informacion existen varios modelos funcionales de este tipo, que incluyen adquisicion de informacion, codificacion de datos, indexacion, organizacion de almacenamiento, formulacion de preguntas, busqueda de informacion, operaciones de salida, y en algunos casos calculos del usuario y operaciones de realimentacion que permiten la reformulacion de preguntas.

El modelo de costo-tiempo-volumen es valioso en situaciones donde las especificaciones del sistema son exactas y cuando se tienen los valores de los parametros, situacion dificil de tener en la practica. Pero cuando se llega a tener estos valores, se puede utilizar una simulacion del sistema, permitiendo un analisis teorico de nuevos conceptos e ideas, incluyendo estudios sobre el crecimiento del sistema, sobre error y confiabilidad y evaluaciones comparativas de nuevas configuraciones del sistema.

CAPITULO 7

Administración de Base de Datos

7.1 DEFINICION DE BASE DE DATOS

Una base de datos es un conjunto de datos que conservan relaciones entre ellos para permitir la ubicacion contextual de grupo, ofreciendo conotaciones individuales para facilitar su recuperacion.

Hay controversia en cuanto a la definicion de los conceptos de bases de datos, bancos de datos y la generalizacion de bancos de informacion. A continuacion se describen las definiciones en polemica, que consideramos que en un momento dado reafirman el concepto unico de lo que queremos decir.

1.-
BASE DE DATOS: Se refiere a la coleccion relacionada de informacion bibliografica o referencial.
BANCO DE DATOS: Se refiere a la coleccion de hechos estadisticos o resultados numericos.

2.-
BASE DE DATOS: Se refiere a la coleccion de informacion primaria o aportada por su generador.
BANCO DE DATOS: La relativa a la informacion secundaria o de referencia, ya que es informacion indirecta.

3.-
BASE DE DATOS: Unidad electronica de informacion almacenada en medios magneticos y que es recuperable.
BANCO DE DATOS: Lugar fisico para el deposito de bases de datos.

4.-Unificacion:
BANCO DE INFORMACION: Coleccion de informacion relacionada de cualquier indole para facilitar su recuperacion.

Es importante hacer notar que se esta distinguiendo en los casos 1.- y 2.- la fuente de la informacion, mientras que en el 3.- se confunde con el sistema administrador de bases de datos, que normalmente suponemos que es automatizado, sin recordar que estos conceptos son aplicables tambien a sistemas manuales de clasificacion y archivo.

7.2 MODELOS CONCEPTUALES DE BASES DE DATOS

Existen tres modelos de bases de datos de acuerdo a la relacion que guardan sus archivos entre si. Estos modelos son:

- Relacional
- Jerarquico
- Reticular

En adelante se describiran brevemente cada uno de estos modelos o estructuras logicas de los sistemas administradores de bases de datos.

7.2.1 RELACIONAL

Desde el punto de vista logico de la informacion, un modelo relacional es aquella matriz formada por atributos y ocurrencias o tuples, que en terminos normales de programacion serian los campos y registros de un archivo respectivamente, donde el nombre de la relacion es el nombre del archivo. En la figura 7.1 se muestra la concepcion logica de una estructura relacional.

RELACION

NOMBRE	EDAD	OCUPACION	SUELDO	
JUAN SALGADO	21	COCINERO	45,415	T
MARIA GOMEZ	26	BAILARINA	85,320	U
PABLO ROBLEDO	34	PINTOR	64,370	P
GILBERTO PEREZ	26	PILOTO	93,805	L
LOURDES SALAS	31	VENDEDOR	45,500	E
MIGUEL RAMOS	45	MUSICO	34,201	S
LAURA ESTRADA	29	ESCRITOR	56,725	

A T R I B U T O S

+ Una relacion es una matriz de dimension [atributos,tuples]
+ Un tuple es un renglon o registro.
+ Un atributo es una columna o campo.

Figura 7.1 Estructura Relacional

De una manera formal, una relacion se define de la siguiente manera.

Sea D un conjunto ordenado de conjuntos Di para

i = 1,2,3,...,n-1,n.

Entonces,

$$D = (Di),\ i = 1,2,3,\ldots,n-1,n.$$

Una RELACION en D se define como un sub-conjunto del producto Cartesiano de D. A los conjuntos Di se les llama DOMINIOS de la relacion.

En otras palabras, R es una relacion en D si y solo si:

(1) R es un conjunto;
(2) cada miembro de R es un conjunto ordenado
(Xi : xi es un elemento de Dl para 1<= i <= n)

Cada elemento de R se llama TUPLE (abreviatura de "N-TUPLE", que es un conjunto ordenado de N elementos). El grado de R es N; Al conjunto formado por el i-esimo elemento de cada tuple de R se le llama el i-esimo atributo de R, definido SOBRE el domino Di, por lo consiguiente el dominio Di esta definido BAJO el conjunto de atributos.

7.2.2 JERARQUICA

Un arbol esta compuesto por elementos jerarquicos, llamados nodos. El nivel mas alto del arbol es un nodo llamado raiz. Exceptuando la raiz, todos los nodos tienen una relacion directa con un nodo superior, al cual se le llama nodo padre. Ningun nodo puede tener mas de un padre, pero si puede tener mas de un elemento en un nivel inferior, al cual se le llamada hijo. Finalmente los nodos que no tengan hijos, se conocen como hojas.

Un modelo jerarquico se refiere a una estructura de arbol entre registros. Algunos manejadores de bases de datos, se disenaron para manejar archivos planos o archivos jerarquicos. En ocasiones, esta estructura es conveniente, pero en general presenta serias deficiencias debido a la limitada flexibilidad que ofrece en la definicion de las estructuras y a la dificultad de acceso a la informacion.

7.2.3 RETICULAR

En una estructura de red, encontramos las mismas relaciones que en un arbol; es decir, un nodo padre es aquel que se encuentra en un nivel superior, un hijo es el que se encuentra en un nivel inferior y existe un nodo raiz. Esto es valido pero encontramos la diferencia en que en una estructura reticular podemos tener para cada nodo uno o mas padres, exceptuando al nodo raiz.

Esta ultima consideracion puede causar problemas en la identificacion de hijos y/o padres, el nodo raiz es unico y siempre lo conocemos ya que es el nodo de referencia.

En una estructura de este tipo, es posible definir ciclos (loops), donde un padre es a su vez el hijo de un mismo nodo, o loops, donde un nodo expresa una relacion con el mismo. Estas dos posibilidades hacen estructuras cerradas, que en general no son muy comunes debido a la dificultad del manejo fisico de la informacion y la complejidad de los programas recuperadores. Existen metodos para lograr que una relacion compleja se convierta en una relacion simple, lo cual se logra duplicando nodos, perdiendo asi integridad de la informacion, e incrementando el tamano del archivo. Cuando la duplicidad de informacion es relativamente pequena, se puede considerar aceptable, pero hay que ser muy cauteloso en las repeticiones para no perder el control y hacer obsoleta la informacion por la perdida de integridad.

7.3 DESARROLLO DE LA BASE DE DATOS

El desarrollo de una operacion sustancial de base de datos invariablemente es una tarea que requiere de la cooperacion de diversos individuos. El grupo incluira a un gran numero de posibles usuarios, a especialistas en el analisis de datos, a cientificos de la informacion, a expertos en varios aspectos de la computacion, a especialistas en comunicaciones, al futuro administrador de la base de datos y a representantes de la gerencia, quienes invertiran recursos de la institucion en el esfuerzo.

No es raro que personas sin amplia experiencia puedan participar en los analisis iniciales, ya que pueden aportar experiencias considerables para el manejo de distintos problemas, pero es necesario que sus opiniones se validen en los sitios de implantaciones previas. La investigacion experimental con grandes bases de datos es dificil, debido al gran costo y a las largas escalas de tiempo asociadas con el desarrollo del sistema de base de datos.

Planteamiento de objetivos.- Los componentes determinantes de una operacion de base de datos son los usuarios y los datos. Los usuarios determinan los objetivos de sus aplicaciones, y la semantica de los datos determina la forma en que estos objetivos pueden satisfacerse. El tipo, la actividad y la cantidad de usuarios y de datos deben cuantificarse antes de que pueda realizarse cualquier diseno especifico del sistema. La determinacion de estos parametros es responsabilidad de la gerencia. Se requiere de retroalimentacion documentada a la gerencia cuando las demandas impuestas de servicio son excesivamente costosas.

Con el fin de que el grupo compuesto por distintos individuos se mueva en la misma direccion, se deben plantear un conjunto de objetivos, especificando los niveles inferiores y superiores de los parametros operativos que se lograran. Algunos limites cuantificados de los objetivos de desempeno del sistema para una aplicacion dada pueden ser:

- Tiempo de respuesta: 90% de las consultas de un solo elemento deben durar menos de 1 segundo desde el principio hasta la conclusion de la consulta.
- Respaldo: Debe disponerse de respaldo para todos los datos fuente que se hayan capturado o introducido hace mas de tres horas. Debe conservarse un respaldo de todos las datos que se eliminaron hace menos de tres meses.
- Puntos muertos: Debera presentarse menos de un punto muerto por ano.
- Costo: Una consulta de un solo dato deberia costar menos de 1000 pesos. Los siguientes elementos consultados costaran 1/20 del costo del primer dato.

Los objetivos definidos y comprendidos en este nivel forman la base para la toma de decisiones en comun y debe preceder a la exploracion de las alternativas de sistema. Los objetivos enunciados sin restriccion, como respuesta instantanea, confiabilidad absoluta, proteccion de la privacia, etc., pueden llevar a un esfuerzo desequilibrado en la implantacion del sistema.

Asignacion. - La asignacion mide la extension en que se ha intercambiado la flexibilidad con la eficiencia. Se han identificado elecciones de asignacion en estructuras de archivo, en modelos de bases de datos, en manejo de esquemas, en recuperacion de informacion, en mantenimiento de datos y en la proteccion de la integridad.

Si se requiere de mucha flexibilidad, el sistema tendra asignaciones no muy estrictas en muchos niveles. La estructura del registro sera variable, las rutas de acceso se aumentaran o eliminaran facilmente, el esquema tendra capacidad para aceptar modificaciones y el sistema de procesamiento de consultas podra interpretar modificaciones realizadas en cualquier momento. Si el desempeno es importante, se impondran restricciones en algunos de los niveles, pero en algunos otros debera conservarse la flexibilidad.

El concepto de asignacion puede promover un entendimiento comun entre la gente implicada en el proceso de desarrollo del problema. Un problema general en el diseno de bases de datos es que se requiere conjuntar a personas dedicadas a muchas disciplinas, tanto usuarios como implantadores.

Documentacion.- En una base de datos la documentacion mas importante es su modelo. El modelo de la base de datos determinara los procesos necesarios para la creacion y el mantenimiento de archivos, y para la recuperacion de informacion. El esquema ampliado con observaciones acerca de los vinculos en el mundo real, las restricciones de conexion y las definiciones de los dominios de variable y su representacion se vuelven el deposito formal para la documentacion del modelo de la base de datos, conforme se afina el diseno.

Cuando se han planteado las elecciones del sistema basico es necesario desarrollar especificaciones mas detalladas. Habra muchas interfases entre los componentes del sistema, las cuales requieren de cuidadosa documentacion. Es dificil apreciar las interfases humanas solo mediante documentacion, y pueden resultar convenientes ejemplos u operaciones piloto. Tambien es indispensable tener especial cuidado para las interfases que utilicen unidades de salida visual, si ha de explotarse su potencial total.

La amplitud y el detalle de la documentacion de programas requerida varian de acuerdo con el objetivo y la complejidad del sistema. Si se utilizan lenguajes de alto nivel, a menudo resulta adecuada una descripcion externa de la funcion de un modulo del programa y documentacion detallada de todas las variables utilizadas. Los diagramas de flujo han resultado utiles cuando interactuan multiples procesos.

El diseno de un sistema generalizado de manejo de bases de datos se vuelve mucho mas complicado debido a la ausencia de objetivos del usuario y de modelos especificos de bases dedatos. En la practica se postulan ciertos tipos de usuarios y se construyen modelos artificiales.

Programacion. - Las tecnicas de programacion estructurada tienen perspectivas como auxiliares en la construccion de grandes programas.

El manejo de una seccion critica de codigo requiere de herramientas paa manejar los mecanismos de seguros. El hecho de que los seguros trascienden los niveles estructurales hace que sean propensos a errores. Si estos temas no se consideraran antes de la implantacion, puede desperdiciarse un gran esfuerzo de programacion. Los intercambios entre el control de integridad, desempeno y flexibilidad en la aplicaciones, no estan bien entendidos por la mayoria de los programadores. Los escritos en donde se afirma que se han resuelto estos problemas han enganado a muchos lectores poco criticos.

La separacion de estructuras de datos es otra herramienta que resulta importante en el desarollo y mantenimiento de los sistemas. El empleo de esquemas proporciona esta facilidad para la base de datos, pero algunas facilidades semejantes son tambien utiles para el acceso a otros recursos compartidos del sistema. El empleo de un enfoque descendente en el diseno de programas es factible cuando existen esperanzas sobre las capacidades en los niveles inferiores. La abstraccion de alternativas en el nivel inferior constituye los bloques de construccion de los niveles susperiores; estas abstracciones deben basarse en construcciones realizables.

7.4 MANTENIMIENTO DE UNA BASE DE DATOS

Cuando un sistema esta concluido, depurado adecuadamente y poblado de datos, empieza el verdadero trabajo. Debe recordarse que el verdadero valor de una base de datos no radica en el manejador de la misma, sino en el contenido de informacion, y particularmente en los resultados de las consultas planteadas.

Cuando en las etapas de implantacion de una base de datos, se requiere de modificaciones, es posible evitar algunas perturbaciones serias no considerando los cambios durante la ultima parte del desarrollo. Los cambios se probaran en una copia de la base de datos, de manera que el uso normal no se vea afectado hasta que se verifiquen los cambios.

Posteriormente se presentara la necesidad de cambiar las estructuras internas de los datos. Un caso frecuente es la necesidad de agregar nuevos datos en el sistema, cuando los usuarios se dan cuenta de que no les es posible efectuar todas las operaciones que ellos esperaban.

7.4.1 Afinacion y vigilancia.

El administrador de sistemas debe vigilar constantemente que exista una proporcion adecuada entre el desempeno y el costo. Un administrador estara desarrollando continuamente herramientas para medir la productividad del sistema. Es importante la sensibilidad con la que se entrevista a los usuarios, para tratar de encontrar las causas de las inconformidades planteadas.

Casi todas las mejoras de desempeno aumentan la redundancia y la asignacion del sistema. Generalmente las rutas de acceso son mas faciles de manipular que las replicas de datos reales. Un sistema que permite la creacion de nuevas rutas de acceso puede asignarse rapidamente para mejorar el comportamiento de recuperacion. Mejorar las transacciones que tienen acceso a muchos elementos tiende a ser mas dificil y tal vez haga necesario considerar la redundancia en los datos. Desde luego, el aumento de la redundancia disminuye el desempeno en la actualizacion. El aumento en la duracion y complejidad en las actualizaciones incrementa la probabilidad de puntos muertos. En general, los fragmentos de bases de datos que no aparecen interrelacionados en forma estrecha pueden separarse con ventajas. Esta separacion, si se realiza despues de que los datos han estado juntos largo tiempo, puede ocasionar problemas, ya que podria revelar la existencia de vinculos que no se habian documentado.

La distribucion de las bases de datos es mejor si se apoya en una particion en fragmentos que no esten fuertemente asignados, ya que esto minimizaria los problemas de autonomia e integridad.

Vigilancia.- Las medidas de la utilizacion del sistema que pueden controlarse, son:
Estadisticas de utilizacion de dispositivos.

Estadisticas de utilizacion de archivos.
Estadisticas de utilizacion de registros.
Estadisticas de utilizacion de atributos.

Las medidas pueden obtenerse mediante muestreos continuos. Usualmente estas mediciones se almacenan en una cinta de movimientos o bitacora de movimientos, para un probable analisis automatizado posterior.

7.4.2 Vida util de los datos y sistemas de bases de datos

Los datos y el sistema pierde su valor con el paso del tiempo. Rsulta dificil predecir que se volvera obsoleto primero, por lo que es necesario considerar ambos efectos.

Conservacion de los datos en forma almacenada.- Mediante el paso del tiempo, los datos pierden su oportunidad y valor, llegando el momento en que su valor hara inconveniente continuar almacenandolos en linea. El bajo costo del almacenamiento intenso en cinta fuera de linea hace posible conservar datos antiguos mientras exista alguna probabilidad de que puedan llegar a necesitarse nuevamente. Los discos opticos ofrecen nuevas alternativas para conservar archivadas viejas versiones al minimo costo.

Los procedimientos utilizados durante la creacion de bitacoras de transacciones han producido una amplia capacidad de respaldo, pero el contenido de las bitacoras y el punto de verificacion estan muy relacionados con el estado del sistema cuando el respaldo se genero, de manera que puede resulta dificil emplear estos archivos. Para el almacenamiento archivado a largo plazo resulta mejor generar cintas en formato de salida, las cuales podran leerse cuando sea necesario mediante procedimientos de entrada. Si las anotaciones del esquema para atributos conservados en los archivos se graban tambien en cintas para el archivo, se conservara mucha de la documentacion del archivo.

Es importante leer los archivos del consecutivo historico de movimientos (archivos historicos) inmediatamente despues de su creacion, con el fin de asegurar que la informacion que esta almacenada sea una fiel reproduccion de la que consta en el archivo original, y no tener la desgracia de encontrar errores, tiempo despues, una vez que sea el unico ejemplar existente.

Ciclo de vida del sistema.- El hecho de que una base de datos se vuelva obsoleta a menudo esta asociado con el equipo fisico que se ha vuelto anticuado, pero lo peor es cuando los programas son inadecuados. El costo de mantenimiento del equipo y de los programas tiende a disminuir inicialmnete, conforme se eliminan los errores ocultos, pero comienza a aumentar de nuevo si se esta forzando el empleo de programas para que sean compatibles con nuevos desarrollos.

Los desarrollos externos que requieren adaptacion pueden ser del tipo tecnico u organizativo; aunque es posible realizar algunas adaptaciones, especialmente en sistemas que no esten rigurosamente asignados o acotados. Una adaptacion necesaria costara mas que una nueva implantacion, en la que tambien podria aprovecharse la tecnologia mas avanzada.

Debido a esto se puede hablar del ciclo de vida de un sistema, comenzando con el diseno, desarrollo, implantacion, carga de datos, operacion, mejoria, mantenimiento, y concluyendo con una transferencia de servicios para renovar el sistema. Una vez que se determina el ciclo de vida de un sistema, es posible tomar otras decisiones. Pueden rechazarse inversiones en mejoras al sistema que no produciran beneficios durante el resto del ciclo de vida.

CAPITULO 8

Transportabilidad de la Base de Datos

8.1 FORMATOS DE COMUNICACION

8.1.1 NORMA ISO 2709

La norma ISO 2709 especifica los estandares para la transferencia de informacion en cinta magnetica.

Cada registro de esta norma se compone de tres partes fundamentales, que son :

1. Encabezado
2. Directorio
3. Campos de datos de longitud variable

El encabezado y el directorio son segmentos de control que describen con precision los datos contenidos en el tercer segmento.

El encabezado comienza con una clasificacion cuya extension se compone de 24 caracteres, de los cuales el contenido es el siguiente:

0	al 4	Extension del registro.
5		Posicion del registro.
6		Reservado.
7		Nivel bibliografico.
8		Reservado
9		Reservado
10		Indicador de extension.
11		Identificador de extension de sub-campos
12	al 16	La direccion base de datos.
17	al 19	Para usuarios de los sistemas.
20		Extension del directorio.
21		Extension del caracter en posicion de comenzar en el directorio.
22		Extension de la seccion definida de cada entrada en el directorio
23		Reservado.

Directorio.

El directorio es un tablero que contiene una cantidad variable de entradas de catorce caracteres cada una, terminados por un caracter separador de campo.

Cada entrada del directorio corresponde a un campo especifico en el registro, el cual se divide en cuatro partes:

- Senal.
- Extension del campo de datos.
- Posicion del caracter donde comienza la informacion del dato.
- Implantacion de la seccion definida.

Campo de datos

Un campo de datos se compone a su vez de tres niveles de informacion

- Indicadores.
- Uno o mas subcampos, cada uno de los cuales es precedido por un identificador de subcampo.
- Un separador de campo.

Subcampo.

Un subcampo se compone de un identificador de subcampo seguido por una hilera de datos, que se termina por otro identificador de campo o por un separador de campo.

Separador de campo.

El separador de campo es un caracter que se coloca al final de todo campo de datos, excepto para el ultimo campo en el registro.

Separador del registro.

El separador de registro es aquel que marca la terminacion del campo de datos final en el registro, y constituye el caracter final del registro.

EJEMPLO:

```
00B090000000004330004500005000500000000600320000510100410000711100360004811200280
008420100220011220000130013422100330014722200100018022300210019022400600211225O
0070021722600900022422700050023322800700238229001500245230001000260230001000270
230008008028023100040028820200260029220400020031821600030032021600030032321600030
032621600030032921600030032221600030033521600030038216000300341218001900344212O
00306363239030500366240004003711 1242 S CONSEJO NACIONAL DE CIENCIA Y TECNOLOGIA
 DIRECCION DE SERVICIOS INFORMATICOS SUBDIRECCION DE INFORMACION BIBLIOTECA-HEME
ROTECA MXDFCCTBHC01 CIRCUITO CULTURAL UNIVERSITARIO EDIF B PB CIUDAD UNIVERSITA
RIA 04515 MEXICO COYOACAN D.F. 20-033 1774521 CNCTME 655-63-66 655-74-88 EX 3019
 915 CHAIGNEAU SOTO / MERCEDES 4 PS PD PI BR CR CO CC DF 8:30-15:00 LUN-VIE LC 6
000 VCT
```

CORRESPONDE A:

ENCABEZADO	DIRECTORIO	DATOS
00B09	005 0005 00000	1242 S CONSEJO NACIONAL DE CIENCIA Y TECNOLOGIA DIRECCION DE SERVICIOS INFORMATICOS SUBDIRECCION DE INFORMACION BIBLIOTECA-HEMEROTECA MXDFCCTBHC01 CIRCUITO CULTURAL UNIVERSITARIO EDIF B PB CIUDAD UNIVERSITARIA 04515 MEXICO COYOACAN D.F. 20-033 1774521 CNCTME 655-63-66 655-74-88 EX 3019 915 CHAIGNEAU SOTO / MERCEDES 4 PS PD PI BR CR CO CC DF 8:30-15:00 LUN-VIE LC 6000 VCT
0	006 0002 00005	
00001	101 0041 00907	
0	111 0036 00048	
0	112 0029 00084	
00433	201 0022 00112	
000	200 0013 00134	
4	221 0033 00147	
5	222 0010 00180	
0	223 0021 00190	
0	224 0006 00211	
	225 0007 00217	
	226 0009 00224	
	227 0005 00233	
	228 0007 00238	
	229 0015 00245	
	230 0001 00260	
	... ETC ...	

8.2 CLASIFICACION BIBLIOGRAFICA

8.2.1 Formato comun de comunicaciones (CCF)

El CCF es un formato que se derivo de la necesidad de intercambiar informacion entre bibliotecas y centros de documentacion. Fue en las reuniones de UNESCO, donde se genero una comision que estudiara la posibilidad y mecanismo que permitiera normalizar la informacion, sin que rompiera las estructuras de clasificacion con las que ya se contaba. En 1984 se presento el proyecto CCF en UNESCO.

Formato CCF.

El proposito del CCF es proporcionar un metodo estructurado y detallado para el registro o grabacion de un numero de elementos de datos opcionales y obligatorios entre un registro bibliografico legible por computadora con fines de intercambio entre dos o mas sistemas basados en computadoras. Sin embargo, esto tambien puede ser util dentro de los sistemas bibliograficos no computarizados. El uso de los elementos de los datos involucrados en el CCF en tales sistemas, simplificara la computarizacion de sus actividades en una fecha posterior.

Una de las politicas adoptadas en la creacion del CCF fue que la estructura del nuevo formato seria conforme a la norma internacional ISO 2709.

Se derivo de una comparacion de todos los elementos de los datos en varios formatos de intercambio bibliograficos, los cuales son: UNIMARL, las guias para ISDS, MEKOF. Las especificaciones de intercambio ASIDEC/EUSIIDEC/ICSU/AB/NFAIS; y el formato de comunicacion comun URSS-US.

Para fines de 1982, el grupo dedicado a la creacion del CCF, habia terminado el diseno. La edicion final tuvo lugar durante 1983.

Dentro de un sistema de informacion, es posible que existan diferentes formatos separados pero que finalmente deben ser altamente compatibles entre ellos, a continuacion se mencionan algunos de estos:

a) formato de captura
b) formato de mantenimiento
c) formato de almacenamiento
d) formato para recuperacion
e) formato de salida
f) formato para intercambio

Por lo que resulta obvio tratar de crear un formato comun que contemple todas las necesidades.

El CCF pretende facilitar la comunicacion de informacion bibliografica.

El diseno del CCF sigue tres propositos principales:

1) permitir el intercambio de registros bibliograficos entre grupos de bibliotecas y servicios.
2) permitir a una agencia bibliografica controlar con una sola serie de programas de computadora, registros de programas bibliograficos recibidos tanto de bibliotecas como de servicios.
3) servir como la base de un formato para una agencia generadora de informacion propietaria de un banco de datos bibliograficos

Estos usos han sido adaptados en las siguientes formas:

- Especificacion de pequena cantidad de los elementos de los datos obligatorios que son reconocidos por todos los sectores de la comunidad informativa como escencial, con el fin de identificar un apartado.

- Proporcionando una lista de elementos de los datos obligatorios, suficientemente flexible para adaptar practicas descriptivas variables.

- Proporcionando una serie de elementos opcionales que pueden ser utiles para describir un apartado mas acorde a las practicas de la agencia que crea el registro.

- Permitiendo a la agencia inventora incluir elementos no normalizados que son considerados utiles dentro de su sistema, aun cuando no sean usados por otras agencias.

- Proporcionando un mecanismo para unir registros y segmentos de registros sin la imposicion de la agencia inventora de cualquier practica uniforme, referente al tratamiento de los grupos relacionados de registros, de los elementos de los datos.

ETIQUETAS DEFINIDAS EN EL CCF PARA DESCRIPCION DE DOCUMENTOS

001	001	RECORD IDENTIFIER
002	010	RECORD IDENTIFIER FOR SECONDARY SEGMENTS
003	010 A	Control number
004	011	ALTERNATIVE RECORD CONTROL NUMBER
005	011 A	Alternative control number
006	011 B	Identification of agency in coded form
007	020	SOURCE OF RECORD
008	020 A	Identification of agency in coded form
009	020 B	Name of agency
010	020 L	Language of name of agency
011	021	COMPLETENESS OF RECORD
012	021 A	Level of completeness code
013	022	DATE ENTERED ON FILE
014	022 A	Date
015	030	CHARACTER SETS USED IN RECORD
016	030 A	Alternative Control Set (C1)
017	030 B	Default Graphic Set (G0)
018	030 C	Second Graphic Set (G1)
019	030 D	Third Graphic Set (G2)
020	030 E	Fourth Graphic Set (G3)
021	030 F	Additional Control Set
022	030 G	Additional Graphic Set
023	031	LANGUAGE OF RECORD
024	031 A	Language of the record
025	040	LANGUAGE AND SCRIPT OF ITEM
026	040 A	Language of item
027	040 B	Script of item
028	050	PHYSICAL MEDIUM
029	050 A	Physical medium code
030	060	TYPE OF MATERIAL
031	060 A	Type of material code
032	080	SEGMENT LINKAGE FIELD: GENERAL VERTICAL RELATIONS
033	080 A	Segment relationship code
034	080 B	Segment indicator code
035	080 C	Bibliographic level code
036	081	
037	081 A	Segment relationship code
038	081 B	Segment indicator code
039	081 C	Bibliographic level code
105	240	UNIFORM TITLE
106	240 A	Uniform title
107	240 B	Number of part(s)
108	240 C	Name of part(s)
109	240 D	Form subheading
110	240 E	Language of item (as part of uniform tit
111	240 F	Version
112	240 G	Date of version
113	240 L	Language of uniform title
114	240 Z	Authority number
115	260	EDITION STATEMENT AND ASSOCIATED STATEME
116	260 A	Edition statement
117	260 B	Statement of responsibility associated w
118	260 L	Language of edition statement
119	300	NAME OF PERSON
120	300 A	Entry element
121	300 B	Other name elements
122	300 C	Additional elements to name
123	300 D	Date(s)
124	300 E	Role (coded)
125	300 F	Role (non-coded)
126	300 Z	Authority number
127	310	NAME OF CORPORATE BODY
128	310 A	Entry element
129	310 B	Other parts of name
130	310 C	Qualifier
131	310 D	Address of corporate body
132	310 E	Country of corporate body
133	310 F	Role (coded)
134	310 G	Role (non-coded)
135	310 L	Language of entry element
136	310 S	Script of entry element
137	310 Z	Authority number
138	320	NAME OF MEETING
139	320 A	Entry element
140	320 B	Other parts of name
141	320 C	Qualifier
142	320 E	Country
143	320 G	Location of meeting
144	320 H	Date of meeting (in ISO format)
145	320 I	Date of meeting (in free format)
146	320 J	Number of meeting
147	320 L	Language of entry element
148	320 S	Script of entry element
149	320 Z	Authority number
150	330	AFFILIATION

040	082	SEGMENT LINKAGE FIELD: VERTICAL RELATIONSHIP FROM
041	082 A	Segment relationship code
042	082 B	Segment indicator code
043	082 C	Bibliographic level code
044	083	SEGMENT LINKAGE FIELD: VERTICAL RELATIONSHIP FROM
045	083 A	Segment relationship code
046	083 B	Segment indicator code
047	083 C	Bibliographic level code
048	085	SEGMENT LINKAGE FIELD: HORIZONTAL OR CHRONOLOGICA
049	085 A	Segment relationship code
050	085 B	Segment indicator code
051	085 C	Bibliographic level code
052	086	FIELD TO FIELD LINKAGE
053	086 A	Identification of field linked from
054	086 B	Field relationship code
055	086 C	Identification of field linked to
056	100	INTERNATIONAL STANDARD BOOK NUMBER (ISBN)
057	100 A	ISBN
058	100 B	Invalid ISBN
059	100 C	Qualification
060	101	INTERNATIONAL STANDARD SERIAL NUMBER (ISSN)
061	101 A	ISSN
062	101 B	Invalid ISSN
063	101 C	Cancelled ISSN
064	102	CODEN
065	102 A	Coden
066	110	NATIONAL BIBLIOGRAPHY NUMBER
067	110 A	National bibliography number
068	110 B	National bibliographic agency code
069	111	LEGAL DEPOSIT NUMBER
070	111 A	Legal deposit number
071	111 B	Legal deposit agency
072	120	DOCUMENT IDENTIFICATION NUMBER
073	120 A	Document identification number
074	120 B	Type of number
075	200	TITLE AND ASSOCIATED STATEMENT(S) OF RESPONSIBILI
076	200 A	Title
077	200 B	Statement of responsibility associated with title
078	200 L	Language of title
079	200 S	Script of title
080	201	KEY TITLE
081	201 A	Key title
082	201 B	Abbreviated key title

151	330 A	Entry element
152	330 B	Other parts of the name
153	330 C	Qualifier
154	330 D	Address
155	330 E	Country of affiliation
156	330 L	Language of entry element
157	400	PLACE OF PUBLICATION AND PUBLISHER
158	400 A	Place of publication
159	400 B	Name of publisher
160	400 C	Full address of publisher
161	400 D	Country of publisher
162	410	PLACE OF MANUFACTURE AND NAME OF MANUFAC
163	410 A	Place of publication
164	410 B	Name of publisher
165	410 C	Full address of publisher
166	410 D	Country of publisher
167	420	PLACE OF MANUFACTURE AND NAME OF MANUFAC
168	420 A	Place of manufacture
169	420 B	Name of manufacturer
170	420 C	Full address of manufacturer
171	420 D	Country of distributor
172	440	DATE OF PUBLICATION
173	440 A	Date in formalized form
174	440 B	Date in non-formalized form
175	441	DATE OF LEGAL DEPOSIT
176	441 A	Date of legal deposit
177	450	SERIAL NUMBERING
178	450 A	Serial numbering and date
179	460	PHYSICAL DESCRIPTION
180	460 A	Number of pieces and designation
181	460 B	Other descriptive details
182	460 C	Dimensions
183	460 D	Accompanying material
184	480	SERIES STATEMENT AND ASSOCIATED STATEMEN
185	480 A	Series statement
186	480 B	Statement of responsibility associated w
187	480 C	Part statement
188	480 D	ISSN
189	480 L	Language of title
190	480 S	Script of title
191	490	PART STATEMENT
192	490 A	Volume/part numeration and designation
193	490 B	Pagination defining a part
194	490 C	Other identifying data defining a part
195	500	NOTE

083	201 L	Language of key title
084	201 S	Script of key title
085	210	PARALLEL TITLE AND ASSOCIATED STATEMENT(S) OF RES
086	210 A	Parallel title
087	210 B	Statement of responsibility associated with paral
088	210 L	Language of parallel title
089	210 S	Script of parallel title
090	220	SPINE TITLE
091	220 A	Spine title
092	220 L	Language of spine title
093	221	COVER TITLE
094	221 A	Cover title
095	221 L	Language of cover title
096	222	ADDED TITLE PAGE TITLE
097	222 A	Added title page title
098	222 L	Language of added title page title
099	223	RUNING TITLE
100	223 A	Running title
101	223 L	Language of running title
102	230	OTHER VARIANT TITLE
103	230 A	Other variant title
104	230 L	Language of title
196	500 A	Note
197	510	NOTE ON BIBLIOGRAPHICAL RELATIONSHIP
198	510 A	Note
199	520	SERIAL FREQUENCY NOTE
200	520 A	Frequency
201	520 B	Dates of frequency
202	530	CONTENTS NOTE
203	530 A	Note
204	600	ABSTRACT
205	600 A	Abstract
206	600 L	Language of abstrac
207	610	CLASSIFICATION SCHEME NOTATION
208	610 A	Notation
209	610 B	Identification of classification scheme
210	620	SUBJECT DESCRIPTOR
211	620 A	Subject descriptor
212	620 B	Identification of subject system

CAPITULO 9

Desarrollo de Equipo Especializado

En este capitulo se examinan dispositivos y metodolcgias utiles para el almacenamiento y el acceso a la informacion, relativos a sistemas mas especializados que permiten obtener acceso rapido a la informacion almacenada, incluyendo procesadores paralelos y asociativos asi como los procesadores de proposito especifico para busquedas o "back-end". Tambien se revisan los principales procedimientos actuales para procesar bases de datos textuales.

9.1 MEJORAS AL HARDWARE DE RECUPERACION.

En las ultimas decadas, se habia venido trabajando la idea de un dispositivo de almacenamiento como un aparato que consume poca potencia y que almacena grandes volumenes de informacion, esto ha sido posible gracias al desarrollo de los microprocesadores que generan la siguiente clasificacion de los dispositivos dependiendo de su proposito y grado de utilizacion.

1. Perifericos inteligentes que permiten la seleccion, el procesamiento, la verificacion en el propio equipo, facilitando las operaciones paralelas, evitando transferencias innecesarias a la unidad central de proceso (CPU) de la computadora central.

2. Las funciones de los perifericos inteligentes pueden incrementarse agregando otras posibilidades de procesamiento, creando procesadores de bases de datos en el manejo de los dispositivos "back-end" conectados a la computadora central, esta controla normalmente al procesador de bases de datos y maneja la transferencia de informacion entre maquinas; donde el procesador "back-end" se encarga de propositos especificos tales como la busqueda o algunas operaciones.

3. La red de computadoras especializadas en la busqueda de informacion. La informacion puede accesarse a traves de cualquier computadora de la red a solicitud del procesador de bases de datos.

   Cuando la computadora central recibe una solicitud de informacion, esta la pasa al "back-end" para que le sea respondida, mientras queda libre para hacer mas agil el modo de operacion interactivo dentro del sistema multiusuario bajo el que se desarrolla.

9.2 PROCESADORES PARALELOS.

Existen dos configuraciones tipicas bajo esta modalidad de procesamiento. La primera se refiere a los sistemas donde hay mas de un procesador pero son totalmente independientes entre si; por lo tanto, es posible hacer busquedas independientes sobre la base de datos, este esquema puede observarse en la figura 9.2.

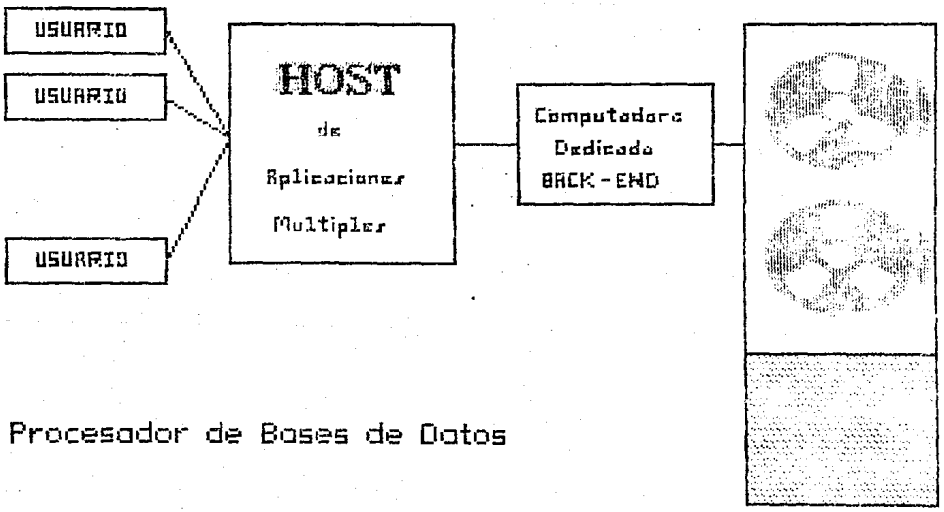
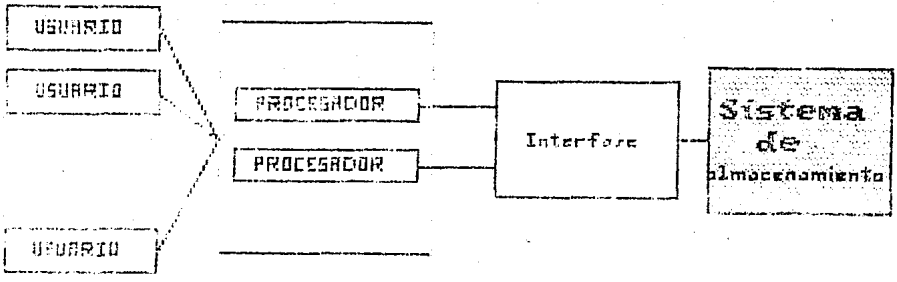

USUARIO
USUARIO
USUARIO
HOST
de
Aplicaciones
Multiples
Computadora
Dedicada
BACK-END
Procesador de Bases de Datos

Figura 9.1



Procesadores paralelos independientes

Figura 9.2

Procesadores paralelos independientes.

Debe ser claro que para una operacion de busqueda, intersectando 3 terminos, el proceso a seguir es el siquiente:

1. Buscar los terminos, haciendo uso del procesamiento paralelo de los procesadores multiples.

2. Intersectar 2 terminos, debe llevarse la operacion a un procesador unico y realizarse secuencialmente. No mejora el tiempo de respuesta de acuerdo al numero de procesadores paralelos en esta configuracion dado que esta operacion requiere de solo uno de estos dispositivos, formando un "cuello de botella" al efectuar las operaciones restrictivas.

3. El resultado del punto 2 se trata nuevamente en forma secuencial con el tercer termino para completar la busqueda.

Hay que senalar que el beneficio de los procesadores paralelos solamente se aprecia en la busqueda paralela de terminos independientes, como en el punto 1, pero debe evaluarse contra el costo extra que representa la multiplicidad de procesadores. Para decidir sobre la ventaja de esta configuracion, se debe asegurar que cada procesador adicional incremente la eficiencia del sistema de informacion.

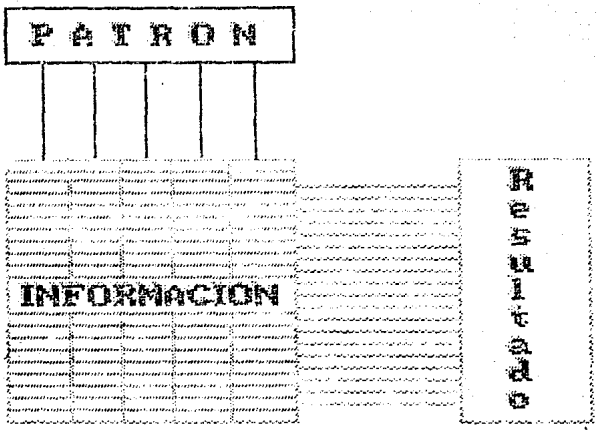
9.3 PROCESADORES ASOCIATIVOS

En la segunda de las configuraciones de procesadores multiples tambien llamada de procesadores asociativos, se basa en la localizacion de ciertos datos de acuerdo con un patron y recibiendo la respuesta en un vector de registros, como se ilustra en la figura 9.3.

Las ventajas del procesamiento asociativo son muchas. No se necesita conocer la localizacion de los resultados parciales, sino hasta que se cumpla el criterio de seleccion. Es muy simple implementar metodos para buscar terminos truncados o con porciones no definidas ("wild cards") del tipo A*A, *AA, AA*. Desafortunadamente la utilizacion de estos procesadores acarrea muchas desventajas, tales como la transferencia de informacion al procesador asociativo, ya que generalmente se realiza secuencialmente y se pierde gran parte de la eficiencia. Segunda, debe especificarse con precision la localizacion del termino a buscar. Tecera, es muy cara y muy escaso el equipo que facilite esta configuracion.

9.4 CALCULOS RAPIDOS USANDO ARREGLOS DE PROCESADORES.

Los arreglos de procesadores se han desarrollado para efectuar calculos logicos y aritmeticos muy rapidamente, y trabajar conjuntamente con una computadora central.
Los arreglos de procesadores se utilizan principalmente como maquinas de punto flotante especializadas y de alta velocidad, trabajando paralelamente con la computadora central. No proveen



CAM
Content Address Memory

PROCESADOR ASOCIATIVO

Figura 9.3

facilidades para el manejo de caracteres ni para operaciones de entrada o salida. La potencialidad de los arreglos de procesadores se basa en las siguientes caracteristicas:

1. Unidades funcionales paralelas: donde en vez de agrupar todas las funciones aritmeticas y logicas tipicas de una unidad central de proceso (CPU) en una sola como en las computadoras convencionales, cada funcion de la unidad central de proceso se separa en unidades funcionales para que puedan operar en paralelo.

2. Unidades funcionales con PIPELINE: que pueden ejecutar ciertas operaciones con mayoir velocidad ya que la descomponen en partes ejecutables en paralelo dentro de la misma unidad funcional, aun cuando el resultado de algunas de ellas les lleve mas tiempo que la ejecucion completa de las otras.

Cuando se acopla un arreglo de procesadores a una computadora central, la computadora se debera encargar del manejo de las operaciones de entrada y salida, operaciones de bases de datos, etc., dejando solamente la ejecucion del programa mientras la computadora central resuelve otras tareas. Hasta que el arreglo de procesadores finaliza la ejecucion encomendada, manda una interrupcion de periferico a la computadora central para que lea el resultado y continue su operacion normal.

9.5 MEMORIA SECUENCIAL DE SEGMENTO DIRECCIONABLE POR CONTENIDO.

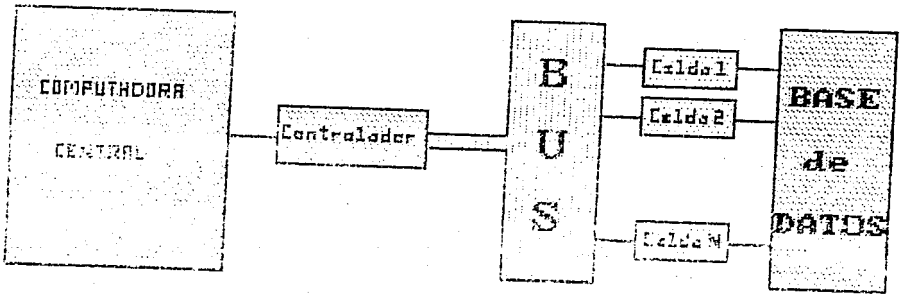
El diseno de este tipo de memorias fue para apoyar las aplicaciones NO numericas a traves de un dispositivo de proposito general. Su funcionamiento se basa en la segmentacion fija de la informacion en distintas unidades de almacenamiento, este tipo de dispositivos, requiere de que el software empleado permita segmentar la informacion de esta manera. En la figura 9.4 se muestra la organizacion tipo CASSM.

Estas memorias son aun un prototipo experimental que se ha manejado hasta la fecha con programas simuladores.

9.6 PROCESADOR ASOCIATIVO RELACIONAL.

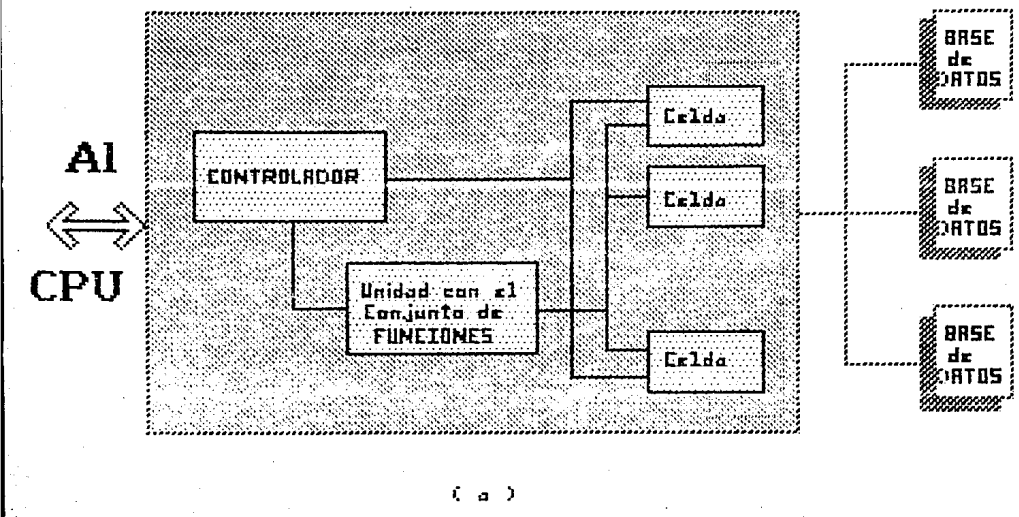
Se diseno con el proposito de procesar la informacion especificada en un modelo relacional de bases de datos, como las memorias secuenciales de segmento direccionable por contenido, estos procesadores tambien son un periferico inteligente, capaz de buscar y manipular los datos en celdas propias o locales en el mismo dispositivo. En la figura 9.5 se muestra un procesador asociativo relacional.

Cada celda contiene un area de memoria para conectar la unidad de proceso a la base de datos, asi como a una unidad aritmetica y logica y una unidad de busqueda y manipulacion de la informacion. Si la base de datos es tan pequena que la memoria local del dispositivo sea suficiente para retener toda la informacion, entonces esta memoria se considera un sistema con procesadores

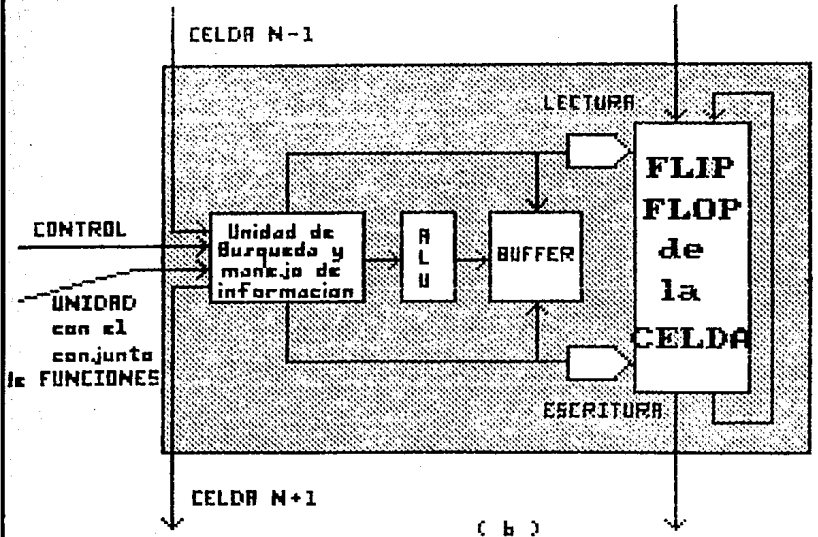


Organizacion de un sistema CASSM

Figura 9.4



(a)



(b)

Figura 9.5 Procesador asociativo relacional a) Organizacion del sistema
b) Organización de la celda

multiples usando busquedas directas a memoria. Pero cuando la base de datos es mayor que la capacidad de almacenamiento de la memoria, esta tendra que accesar repetitivamente en un proceso de busqueda, por lo tanto se considera que es un dispositivo de procesadores multiples con busqueda indirecta. Esta segunda condicion es la mas normal en sistemas de esta naturaleza, por lo tanto siempre estara haciendo busquedas indirectas, con accesos multiples.

9.7 COMPUTADORAS PARA BASES DE DATOS.

Fue disenada para el manejo de grandes bases de datos. Debe conectarse a una computadora central que sea la interface con el usuario. Las operaciones de busqueda se dividen en dos ciclos, en el primero se incluyen las funciones para accesar los datos, el cual es llamado ciclo de datos. En el segundo, se incluyen las funciones asociadas con la estructura de la informacion, por lo que se llama ciclo estructural, a continuacion se muestra una computadora de bases de datos. En la figura 9.6 se muestra la organizacion de una computadora para bases de datos.

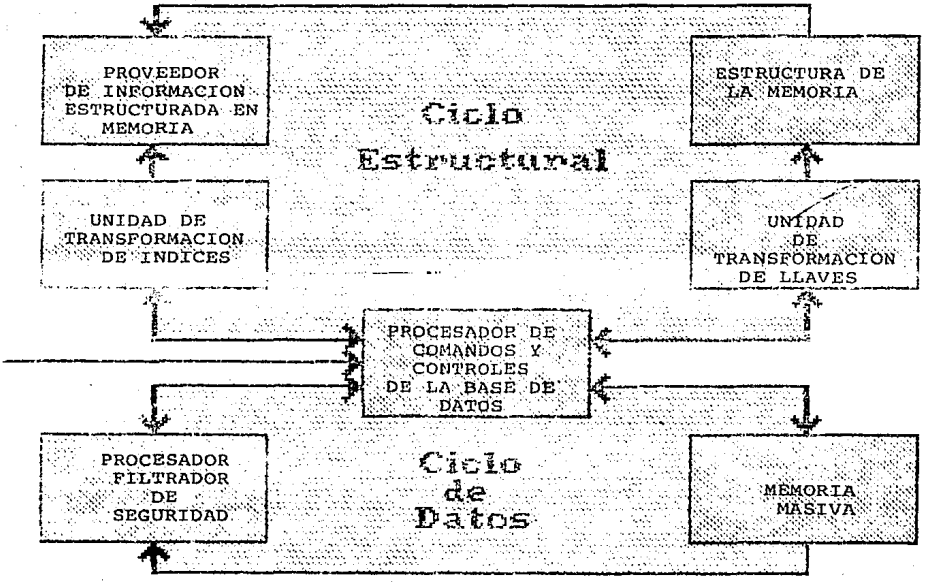
El modelo central de esta computadora es el procesador de control y comandos. Este elemento recibe una solicitud de la computadora central, la cual es traducida a comandos para los diversos componentes funcionales de la computadora de bases de datos y controla el funcionamiento independiente de cada modulo. Cuando se ejecutan los comandos, este procesador envia la informacion solicitada a la computadora central.

El ciclo estructural se compone de cuatro unidades, las cuales traducen una solicitud en un conjunto de localidades fisicas por buscar.

El ciclo de datos tiene dos unidades que accesan y validan la informacion antes de enviarse a la computadora central.

La estructura de la memoria del ciclo estructural, es un archivo invertido a la base de datos.

Las unidades de decodificacion y codificacion de llaves o datos trabajan a manera de PIPELINE.



ORGANIZACION DE UNA COMPUTADORA PARA BASES DE DATOS

Figura 9.6

CAPITULO 10

Tendencia de los Recuperadores de Información

10.1 INTRODUCCION

Es evidente que la tendencia de los sistemas de informacion dependen de las teorias estudiadas actualmente y que estan en fases de experimentacion activa.

Las teorias que han despertado mayor interes son las que se tratan de sistemas con procesamiento de languaje natural, usando representaciones complejas del contenido de informacion, para lograr la manipulacion de diversos materiales incluyendo graficas y simbologia especializada. Las tecnologias y dispositivos nuevos disenados incluyen "chips" de procesamiento especial, microprocesadores dedicados, lectores opticos de caracteres, memorias opticas, y dispositivos de micro-graficacion.

Se espera que los desarrollos teoricos se acoplen a las nuevas tecnologias en un tiempo cercano, logrando la implantacion de sistemas flexibles y suficientemente poderosos que permitan controlar muchas actividades diferentes para el procesamiento de archivos. En un tiempo cercano, los documentos impresos seran remplazados por sistemas "electronicos" (paperless) en donde se procesaran con formatos legibles por computadora y todo el flujo de operaciones se llevara electronicamente.

Es dificil emitir una opinion definitiva sobre los sistemas de informacion del futuro. Sin embargo, es posible estudiar los desarrollos que deben esperarse para formar la base del diseno del sistema manejador de informacion del futuro.

10.2 DESARROLLOS TECNICOS

10.2.1 Captura automatica de documentos.

En los ultimos anos los sistemas de informacion han adquirido mucha popularidad. A medida en que se agragen elementos en la poblacion de usuarios interesados en usar las facilidades de busqueda y recuperacion automatica. Desafortunadamente existen aun muchos tipos de material que no pueden incorporarse en las bases de datos.

En principio, es posible encargar a un capturiasta, perforista o encargado de captura, la conversion de documentos impresos en formas magneticas legibles por computadora. Un capturista puede escribir a un ritmo de 100 palabras por minuto evitando errores, esto genera 192 paginas a doble espacio en una jornada de 8 horas, suponiendo que no hay interrupciones ni correcciones. Una persona experta en captura, puede escribir 50 paginas en un dia, lo cual es una cuota mas razonable.

Afortunadamente se esta estudiando la solucion al problema para la captura de datos. Necesariamente seguira existiendo la escritura en alguna maquina aun para obtener el documento original. Estos originales se estan produciendo con equipo capaz

de retener la informacion tecleada y retransmitirla posteriormente, ya que se encuentran en un formato legible por computadora. En el futuro se espera incrementar el numero de documentos disponibles en este formato, para alimentar los sistemas de recuperacion de informacion.

En la actualidad se emplean las microcomputadoras como procesadores de texto y la posterior incorporacion a las bases de datos. El equipo procesador de texto se compone de:

1. Teclado
2. Unidad de almacenamiento
3. Monitor
4. Impresora
5. Una conexion a una computadora o red de transmision de datos que unifique varias estaciones de trabajo.

Un procesador de texto puede funcionar como estacion originadora o receptora en un sistema de correo electronico.

Con el proposito de resolver los problemas de captura, puede usarse equipo que lea caracteres de texto mediante metodos opticos para convertir la informacion impresa en un formato legible por computadora.

Los lectores opticos actuales, transforman diferentes juegos de caracteres, a una velocidad de 80,000 caracteres por hora, produciendo mas de 500 paginas en un periodo de 8 horas.

El equipo mas sofisticado reconoce correctamente cerca del 90% de los caracteres, y un 70% de palabras contenidas en texto. Si a este proceso se le agraga una revision manual, se puede lograr que la informacion alcance el 100% de precision.

Esta revision posterior se realiza frecuentemente en dos pasos:

1. Procesamiento automatico de reconociemiento de caracteres.
2. Remplazo manual, para corregir los errores del paso anterior.

Obviamente el proceso de postedicion incrementa el costo y decrementa la eficiencia del reconocedor. El proceso automatico de reconocimiento determina el nivel de certidumbre de cada caracter como una funcion del grado de parentezco entre las caracteristicas registradas a la entrada y los patrones almacenados en el sistema reconocedor.

Un sistema reconocedor de caracteres puede complementarse por otro sistema de reconocimiento vocal. Con un sistema como este, los datos pueden dictarse, leerse, o introducirse auxiliados del procesador de texto. Este metodo vocal ha sido el que mas problemas ha tenido para su funcionamiento, aun cuando ya existen aparatos capaces de reconocer ciertas palabras, pero se limitan debido a la pronunciacion, el idioma y principalmente el vocabulario, siendo el mismo problema que en la recuperacion en lenguaje natural.

10.2.2 Almacenamiento optico.

El almacenamiento de informacion es un punto critico de los sistemas recuperadores de informacion, la eficiencia de actualizacion o la localizacion de los documentos dentro de la base de datos depende fuertemente de los dispositivos de almacenamiento.

Es comun que el usuario desee la informacion tal cual se encuentra en la furnte primaria y no una descripcion o referrencia bibliografica, mas aun, llega a pensar en la obtencion de graficas o dibujos, esto genera una gran variedad de tamanos en los archivos, tipos de informacion almacenada, etc..

Una fotografia o cualquier otro medio de almacenamiento optico puede utilizarse, ya que es capaz de guardar informacion digital asi como informacion de video. Los dispositivos de video mas usados son, los videodiscos, hologramas y micrograficacion.

Videodiscos. La importancia de este dispositivo de almacenamiento, reside en que permite almacenar grandes cantidades de informacion a un bajo costo. Los videodiscos son grabados mediante un rayo laser, quemando la superficie del disco. Este dispositivo permite unicamente la lectura del disco una vez grabada la informacion. La capacidad de estos discos es cercana a los 1000 megabytes de informacion en formatos de 12 pulgadas, tambien se encuentran de 5 pulgadas 1/4 con una capacidad de 500 a 600 megabytes de informacion o hasta 3000 imagenes, a este ultimo se le conoce como CD-ROM. Existe tambien una tecnica de grabacion en disco compacto, con la posibilidad de agregar informacion, no borrarla, es conocida como W.O.R.M. (write once read mostly).

Holografia. Es una fotografia que muestra un objeto en tercera dimension. Se logra iluminando un objeto con rayos laser desde dos angulos diferentes. Una caracteristica interesante de la holografia, es que aun recortando una fotografia por la mitad, puede percibirse el objeto completo en cualquiera de las partes, sin embargo, pierde resolucion.

Esto modifica substancialmente la posible perdida de informacion ya que puede regenerarse con tan solo un fragmento de ella.
Un holograma es capaz de almacenar cerca de 20 millones de caracteres de informacion. Este dispositivo como el videodisco, impiden la re-escritura o borrado como tal.

Micro-graficacion. Estos dispositivos, asi como los de microfichas o microfilmacion, son otra solucion al problema de almacenamiento. Estos dispositivos se utilizan para guardar grandes volumenes de informacion, la economia reside en la reduccion de tamanos de los documentos (24 veces normalmente), logrando que la plantilla convencionalmente almacene hasta 98

paginas de informacion digital en formato visual. La desventaja que se encuentra en estos equipos es la necesidad de contar con un aparato que amplifique la imagen, haciendo esto que los usuarios prefieran los documentos originalmente impresos.

10.3. TEORIAS DE INFORMACION Y MODELOS.

10.3.1 Procesamiento en lenguaje natural.

Sin duda en la prediccion de los problemas a resolver en el futuro, se encuentra el entendimiento y procesamiento del lenguaje natural, escrito y hablado. Si los lenguajes de consulta pudieran utilizar una gramatica libre en lenguaje natural y se pudiera analizar automaticamente la informacion, obteniendose resumenes, descriptores, clasificaciones, etc., las principales dificultades en la realizacion de una base de datos practicamente desaparecerian. No habria que preocuparse por el lenguaje controlado, entrenamiento de indizadores expertos, ni por intermediarios que realizan las consultas por el desconocimiento de los terminos validos en un sistema determinado.

Desafortunadamente, la manipulacion libre, no restringida del lenguaje natural no es un prospecto para un futuro cercano. Particularmente, no hay un arreglo final en la forma de procesar los documentos, ni del conocimiento necesario para entender textos e interpretarlo en lenguaje natural.

Los avances al respecto en ciencias de la informacion, se encaminan a utilizar tecnicas especializadas para ofrecer al menos una forma linguistica al analisis actual de los documentos para su incorporacion y recuperacion. Estos metodos, se basan en la obtencion de frases particulares y en la asignacion de descriptores.

La dificultad mas grande en el procesamiento en lenguaje natural, es la flexibilidad y extension de la mayoria de los lenguajes. Existen muchas formas para decir la misma cosa. Por lo tanto existiran muchos terminos que variaran su significado de acuerdo con el contexto en que se emplean, haciendo que se generen varias versiones del significado de lo expresado.

Existen otras tecnicas que se han empleado en sistemas de informacion atacando el problema de la ambiguedad del lenguaje natural. El modelo de procesamiento vectorial, se asignan pesos a los terminos que identifiquen un documento reflejando la importancia de cada termino o el grado de certidumbre en que describe al contenido del documento. Asi que, el grado de afinidad entre dos elementos puede variar dependiendo de la certidumbre o interpretacion del contenido de los descriptores del documento respectivo

10.3.2 Teoria de conjuntos difusos (FUZZY SET).

La idea principal de esta teoria es que los elementos que conforman el conjunto, pertenecen a el en diferentes grados de propiedad. Esto significa que en vez de decidir si pertenece o no al conjunto, se asigna una funcion de membresia que refleja la fuerza con la que se adhiere al conjunto. Esto tiene un interes muy alto en los sistemas de informacion ya que se recuperaran los documentos solicitados con menor interferencia, dado que se establecen criterios de semejanza para la recuperacion de informacion.

En procesamiento de lenguaje, se han hecho varios intentos usando conjuntos difusos para modelar la ambiguedad y ambivalencia.

Por ejemplo, dado un conjunto bien definido de categorias de significados, el significado total de una palabra dada puede expresarse como una combinacion ponderada de las funciones de membresia de la palabra en varias clases de significado. Los calificadores linguisticos y los delimitadores pueden describirse tambien, utilizando medidas difusas de alguna clase.

En sistemas de informacion, la aproximacion de los conjuntos difusos puede usarse para clasificar los documentos en grupos difusos afines y tambien para controlar el proceso de recuperacion.

Considerese un documento DOC y un termino particular A.S. A denota la "clase de concepto" de todos los elementos que se refieran al sujeto denotado por A, entonces la funcion de membresia del documento DOC en el conjunto A puede escribirse como Fa (DOC). En la terminologia usual, esta expresion representa la ponderacion del termino A en el documento DOC.

Sean A,B...,Z un numero dado de clases de conceptos, representando varias areas especificas, es posible identificar cada documento expresando su funcion de membresia con respecto a cada clase de concepto esto es:

D=[Fa(DOC), Fb (DOC),...,Fz(DOC)]...(1)

Como se observa en esta expresion, se esta generando una representacion victorial ponderada sobre cada clase.

La distancia (o similaridad) entre dos documentos o entre el documento y la solicitud de informacion, puede obtenerse como una funcion de las diferencias entre las funciones de membresia en sus correspondientes clases de conceptos.

Especificamente, sean T diferentes clases de conceptos, la distancia difusa entre el documento DOC' y DOC" se obtiene como

$$d(DOC',DOC") = \sum_{x \in T} Fx(DOC') - Fx(DOC")$$

$$d(DOC',DOC") = \left(\sum_{x \in T} (Fx(DOC')-Fx(DOC"))^2\right)^{0.5}$$

En la busqueda de un DOC, se puede lograr la recuperacion dentro de un rango de distancias difusas al punto de interes.

Una caracteristica atractiva de los conjuntos difusos es la capacidad de extender la definicion de la funcion de membresia a una combinacion de terminos. Por lo tanto dadas las funciones de membresia de los terminos A y B a las clases de conceptos, pueden seguirse las siguientes reglas para aplicar las operaciones booleanas a las combinaciones de terminos:

F (A and B) (DOC)=Min (Fa(DOC), Fb(DOC))
F (A or b) (DOC) =Max (Fa(DOC), Fb(DOC))
F (NOT A) (DOC) =1 - Fa(DOC)

La atraccion que tiene el modelo de conjuntos difusos es la compatibilidad con los sistemas convencionales de procesamiento de solicitudes booleanas, asi como la interpretacion de las ponderaciones difusas como indicadores linguisticos de la ambiguedad y ambivalencia de terminos.

CAPITULO 11

Implantacion de un Caso Practico

11.1 ANTECEDENTES

La biblioteca de la Facultad de Ingenieria registra diferents tipos de material documental, dentro de los cuales existen monografias, estadisticas, publicaciones periodicas, tesis, diagramas, proyectos de investigacion, mapas, patentes, conferencias, reportes, etc.

El sistema de informacion que en esta tesis se plantea, esta enfocado a la automatizacion de aquellos documentos que pueden describirse bibliograficamente.

El caso particular que se resolvera como un ejemplo de la metodologia sugerida a lo largo de este trabajo es de las tesis de licenciatura de la facultad de ingenieria.

Actualmente esta informacion se consulta en forma manual a traves de las tarjetas catalograficas por titulo, autor y descriptores.

Los terminos seleccionados como descriptores del contenido de un documento son muy rigidos y esto se debe a que se tiene que pensar detenidamente en que terminos se van a utilizar, y dichos terminos deben encontrarse en el diccionario. Por ejemplo, si se quere utilizar la palabra "analisis" como descriptor, y en el diccionario solo se encuentra la palabra "analizar", esta es la que debe de utilizarse. Ademas cada documento esta limitado a manejar solo unos cuantos descriptores (En la practica no siempre resultan ser los mas importantes), debido al trabajo que representa el tener que darle mantenimiento a tantas tarjetas.

La consulta a la informacion tambien se hace en forma manual y consiste en ir a los tarjeteros y empezar a buscar por titulo o por autor cuando se conoce alguno de ellos. Pero en la mayoria de los casos se desconoce la existencia del documento, por lo que no se tiene ni el titulo ni el autor, sino que se busca por algun descriptor del tema deseado, el cual no necesariamente esta incluido como descriptor en alguna de las tarjetas. El hacer la busqueda en forma manual propicia el no tener bien definida una estrategia de busqueda, sino que esta se limita al interes que puede despertar la descripcion de cada uno de los documentos, provocando con esto la inclusion de material no relacionado con el tema de nuestro interes.

La seguridad de la informacion depende del control y vigilancia que se tenga de los tarjeteros, ya que en caso de faltar una tarjeta se pierde el acceso al documento, el cual ya no es consultado, sino hasta que se le de mantenimiento a los tarjeteros y se detecte su ausencia (En la mayoria de las bibliotecas el mantenimiento se lleva a cabo cada año).

DESVENTAJAS:

- Duplicidad de tarjetas y trabajo
- Tiempo de consulta
- Estrategia de busqueda (and, or, not)
- Ordenamiento alfabetico de las tarjetas
- Rigidez en la catalogacion
- Seguridad de las tarjetas
- Mantenimiento del tarjetero

CONSIDERACIONES

Actualmente existen 1500 tesis registradas en la coordinacion de seminarios y mas de 500 tesis en proceso. Esto implica un crecimiento de cerca de 300 tesis anuales.

Para poder ofrecer un buen servicio se va a analizar la posibilidad de automatizar el sistema de control y consulta de tesis, para lo cual se va a seguir el esquema de un estudio de viabilidad sin entrar a detalle.

11.2 DIAGNOSTICO DE LA SITUACION ACTUAL.

Funciones y Objetivos.

- Disenar la infraestructura del banco de datos utilizando un paquete recuperador de informacion.

- Solucionar la problematica de consultar tesis, por lo que se pretende facilitar y agilizar el servicio de consulta de tesis.

- Desarrollar mecanismos de consulta de tesis.

- Almacenar y actualizar periodicamente la informacion para mantenerla vigente.

- El banco de datos va a proporcionar los medios al alumnado y al personal docente para elegir la tematica de investigacion o localizacion de fuentes de consulta para futuras tesis.

- Promover al usuario de informacion con el fin de desarrollar la investigacion.

- Facilitar el desarrollo de estadisticas relacionadas con los proyectos, temas, areas, asesores o autores de las tesis.

- Permitir la visualizacion de los temas de tesis aceptados y propuestos en la universidad.

- Hacer del conocimiento publico los trabajos de investigacion que se realizan en la universidad.

- Dar a conocer la trayectoria de temas de interes.

- Permitir el analisis cualitativo y cuantitativo de la tematica desarrollada.

- Establecer los vinculos necesarios para el intercambio o adquisicion de conocimientos.

Considerando que se desea ofrecer un servicio a la comunidad universitaria tratada por su complejidad como un servicio publico interactivo, es necesario analizar tanto a la unidad generadora como a la usuaria para obtener una mejor ubicacion dentro del contexto.

Analizando la unidad generadora de informacion, se encuentra que es la coordinacion de Seminarios, que informa de las tesis concluidas, aprobadas en desarrollo, aprobadas y abandonadas de acuerdo a sus reportes de estado de avance, con informacion detallada de cada una de ellas respecto al tema, autor, fecha, de registro, numero de paginas e idioma, siendo este ultimo un campo opcional para cuando se encuentren ejemplares en otro idioma ademas del idioma oficial (español).

Forma de generar la informacion:

Para el registro de una tesis se acude a la coordinacion de seminarios en donde al menos existen 2 caminos para la aceptacion de temas de tesis:

1) Inscribirse en algun seminario propuesto a la facultad por algun profesor donde se este solicitando la asistencia de alumnos interesados. En este caso el profesor indicara el numero de participantes que acepta en su grupo de trabajo.

2) El alumno o alumnos propondra(n) su propio tema de tesis que previamente ha(n) discutido con su asesor. Esta propuesta se entrega en la coordinacion de Seminarios donde se analizara para su aprobacion, modificacion o rechazo.

El registro de tesis consiste en: Dar un titulo de la tesis, temas que trata, objetivos, nombrar a un asesor, asignarle una fecha de inicio y numero de identificacion.

- Para mantener en vigencia el estado de la tesis es necesario presentar reportes del avance autorizados por su asesor, indicando sus posibles alteraciones, y un porcentaje estimado del avance para la culminacion del trabajo.

Dado que los datos son referencias bibliograficas de los estudios realizados en la universidad, la informacion generada tiene valor cientifico y tecnico que cuando pierde su vida util pasa al acervo historico conservando un gran valor estadistico.

La actualizacion de la informacion proporcionada por la coordinacion puede tener la misma frecuencia que la de la entrega de reportes mas los informes de los trabajos concluidos.

La informacion que debera generar la coordinacion de seminarios para la creacion o actualizacion de sistema de informacion de tesis debera llevar un analisis para la homogeneizacion del Vocabulario empleado; asi como para complementar la informacion recabada y requerida como minimo obligatorio. En este punto debera observarse que para las tesis en desarrollo se asignaran descriptores provisionales mientras que para las tesis concluidas la mejor opcion sera auxiliarse de los especialistas que existen en las bibliotecas afines al tema, para su mejor juicio en la asignacion de los descriptores definitivos.

En el caso de la FI, a nivel licenciatura se han recabado los datos de la tabla 11-A donde es notorio que anualmente el numero de alumnos titulados no excede 340 ni mucho menos 500 alumnos si pensamos en incluir a los del area mecanica electrica que no fueron considerados. Esta estadistica refleja los trabajos de tesis concluidos y podra mejorarse conforme se agreguen datos de las tesis en desarrollo. Para un caso practico consideraremos que se elaboran 1500 tesis anuales concluyendo exclusivamente el 30% de estas.

Actualmente se cuenta con un comite para la aprobacion de tesis, el cual podra tomar decisiones mas atinadas cuando sepa con precision cuantas y cuales investigaciones se han desarrollado bajo que temas, y cuales quedan aun por estudiar. Esto con el objeto de lograr una distribucion mas uniforme.

Se pueden distinguir 2 categorias de unidades usuarias: la que compete a labores administrativas para la generacion de estadisticas y registro de las investigaciones, y la que se proporciona como un servicio auxiliar de la biblioteca, en donde se pretende orientar bibliograficamente sobre algun tema en particular tal como se muestra en la figura 11.1.
Los usuarios del area administrativa utilizaran solo aquella informacion estadistica para responder las siguientes preguntas y sus combinaciones:

1) Cuantos alumnos de la carrera "X" estan registrados?
2) Cuantas tesis concluidas existen?
3) Cuantas tratan el tema "Y"?
4) En cuantas tesis ha participado el asesor "Z"?
5) Cuantas tesis de licenciatura estan en desarrollo?
6) Cuantos trabajos de tesis han sido desarrollados por "N" alumnos? (por 1,2,3 alumnos..?)
7) Que tesis esta desarrollando el alumno "A"?
8) Cuantos reportes se han entregado para la tesis "t"?
9) Cuantas tesis se registraron en el ano "x"..?
10) Cuantas tesis se concluyeron en el ano "y"..?
11) El numero de usuarios del area administrativa vá a estar limitado al control interno y a consultas ocasionales de los alumnos. El sistema por cuestiones de seguridad sera utilizado por personal autorizado por el administrador del sistema.

CARRERA	# año carrera	81	82	83	84	85	TOTAL
MECANICA	27	48	47	90	87	33	314
INGENIERIA INDUSTRIAL	28	39	82	69	83	70	303
SISTEMAS ELECTRICOS Y ELECTRONICOS	29	59	77	71	49	67	323
* (no incluye computacion)	OTROS	11	24	80	69	51	235
COMPUTACION	32	2	5	17	27	36	87
TOTAL		159	215	338	295	257	1262

NOTA: * No indicaron el area de Ingenieria Mecanica Electrica en que se encontraban registrados.

TABLA 11.A Numero estimado de alumnos que se han titulado desde 1981 al 11/11/85

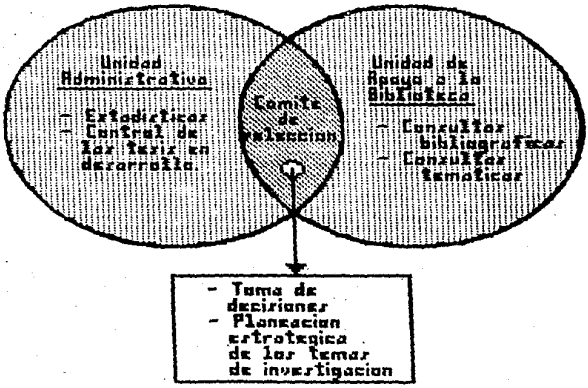


Figura 11.1 Categorias de unidades usuarias

El sistema no hace inferencias deductivas ni hace calculos estadisticos; el sistema solo es capaz de responder preguntas directas siempre y cuando no impliquen calculos. Por ejemplo, el sistema no sera capaz de totalizar el numero de alumnos que estan realizando trabajos de tesis.
Los usuarios del sistema auxiliar de la biblioteca requieren de la informacion bibliografica que les permita optimizar el tiempo para su investigacion. Este sistema debe ser capaz de responder a las siguientes preguntas y sus combinaciones:

1) Que temas se han escrito sobre el tema "t" ?
2) Que caracteristicas tiene la tesis "x" ?
3) Que asesores han tratado un cierto tema ?
4) Que tesis ha escrito el autor "A" ?
5) En que tesis aparece la palabra "P" ?

La interseccion que se muestra en la figura 11.1 es el resultado de la combinacion de estas 2 unidades usuarias de informacion, y seria utilizada por los miembros del comite como una herramienta para la toma de decisiones.

La Facultad de Ingenieria cuenta con diversos equipos de computo dentro de los cuales se encuentran las minicomputadoras y microcomputadoras que se utilizan en las areas administrativas y al servicio de los alumnos.

DETERMINACION DE REQUERIMIENTOS.

Se debe definir el formato de la informacion para poder calcular el espacio de almacenamiento requerido, este se presenta en la siguiente seccion.

La captura de la informacion requerira de aproximadamente 1000 horas/hombre para los primeros 2000 registros.

En promedio los registros contendran 1000 bytes, la utilizacion de apuntadores e indices hace quese multiplique esta cantidad por un factor de 2.5, el cual se ha sacado en base a la experiencia.

Por lo tanto para el manejo de informacion de 5 anos, considerando los datos actuales (sin incrementos) se requeririan de:

$$1{,}500 \times 1{,}000 \times 2.5 \times 5 = 18.75\ M$$

$$\frac{\text{Registro}}{\text{Ano}} \times \frac{\text{Byte}}{\text{Registro}} \times \text{Ano} = \text{Byte}$$

Dado que la biblioteca y la coordinacion de seminarios operan todo el dia, para poder proporcionar un servicio adecuado se requiere de la utilizacion de equipos dedicados, con capacidad de transporte de datos independiente. De los equipos con los que cuenta la FI, los que cuentan con la caracteristica de poderse utilizar como equipos dedicados serian los equipos micro.

La memoria central requerida es de cuando menos 10 registros simultaneos, mas los requerimientos del programa.

El tipo de unidades de almacenamiento debe ser para informacion masiva de alta velocidad, por lo que se recomiendan discos de tecnologia Winchester. Dentro de los equipos micro de la FI que tienen disco duro se encuentran los equipos compatibles con la IBM PC; tales equipos se encuentran en el edificio del DIME, que es otra de las caracteristicas favorables ya que en una fase inicial seria utilizada por la Coordinacion de Seminarios que se encuentra en dicho edificio.

El respaldo de informacion puede hacerse tanto en cintas como en discos flexibles o cartuchos de disco duro.

Para evitar los estados de hibernacion y puntos muertos se recomienda la duplicidad de los archivos en sistemas virtuales multiusuario dado que para cada estacion de trabajo se estaran marcando ocupados los mismos archivos ya que es la misma aplicacion.

El unico programa que se requiere para el logro de nuestros objetivos es un recuperador de informacion.

El recuperador de tesis permitira ademas actualizar las tarjetas catalograficas, para seguir proporcionando la informacion a traves del sistema tradicional de tarjetas. (Este seria un metodo de consulta alternativo.).

Para la homogeneizacion de terminos no es posible utilizar un tesauro de los existentes, ya que estos se encuentran en ingles y el sistema a desarrollar va a contener informacion en espanol; por lo que se tiene que elaborar un diccionario de vocabulario controlado de terminos ingenieriles en espanol con el objeto de uniformar la informacion y para asegurar que en las busquedas se obtengan todos los documentos afines al termino que se solicita.

CONCLUSION

Del analisis anterior se puede concluir que la automatizacion del sistema de tesis es viable por las siguientes razones:

- Se cuenta con el equipo necesario para la implantacion del sistema.
- La universidad puede conseguir el paquete recuperador de informacion en forma gratuita.
- Debido a que la facultad de ingenieria cuenta con programas de servicio social dentro de la coordinacion de seminarios, es posible reclutar personal que haga las labores de captura y actualizacion de la informacion del sistema al menor costo.

- El beneficio que va a obtener la comunidad universitaria es muy alto desde cualquier punto de vista.

- La programacion de los siguientes temas de tesis se plantearan teniendo una vision mas amplia, evitando la duplicidad de esfuerzos y fortaleciendo los trabajos de investigacion de la facultad.

La alternativa de solucion mas viable para efectos de inicio del sistema es la formada por la configuracion de microcomputadoras PC con discos duros independientes de 20 M Bytes,512 K Bytes de memoria, 1 manejador de discos flexibles, impresoras para cada equipo, con el sistema operativo mas comun en este tipo de computadoras es el MS/DOS para el cual se encuentra un desarrollo del 60% del Software mundial para microcomputadoras. El sistema recuperador de informacion mas conveniente para nuestros propositos es MICROISIS por contar con caracteristicas que lo hacen mejor que otros paquetes recuperadores de informacion para microcomputadoras. A continuacion se mencionan algunas de ellas:

- Es un paquete guiado por menus, por lo que resulta muy facil de utilizar.

- La definicion de la base de datos se hace en forma facil y flexible.

- Se puede predefinir la informacion de los campos.

- Permite cinco tipos de indexacion.

- Cuenta con los operadores Booleanos (AND, OR y NOT), con parentesis y truncacion a la derecha, libre o con numero de caracteres.

- Posee un lenguaje muy flexible para elaborar formatos de salida.

- Permite traducir la informacion en formato internacional ISO2709 tanto a la entrada como a la salida.

- Tiene cuatro idiomas para usarse (Espanol, Portugues, Frances e Ingles).

- Se puede modificar la presentacion de los menus.

- El paquete puede funcionar con discos flexibles; es decir, no requiere de disco duro.

- Puede manejar sinonimos.

- Permite la inclusion de un antidiccionario para la indexacion.

- Se puede imprimir el vocabulario de terminos recuperables.

- Permite el ordenamiento de hasta cuatro campos anidados.
- Utiliza archivos con organizacion aleatoria con estructura de BTREE+ para optimizar el tiempo de respuesta en las busquedas.
- Permite la impresion de la informacion en varias columnas.
- Permite el uso de subcampos y campos repetibles.

Este paquete es proporcionado gratuitamente por la UNESCO, mediante la firma de un convenio, a instituciones no lucrativas de ensenanza superior.

11.3 DETERMINACION DEL FORMATO

El instructivo para el llenado del formato para el asentamiento de los datos bibliograficos en el banco de informacion de tesis elaboradas en la Facultad de Ingenieria de la UNAM, esta constituido por once variables y 20 campos distribuidos en forma de renglones, como se indica a continuacion

V A R I A B L E S

#	1	2	3	4	5	6	7	8	9	10	11
C A M P O S	001 022 020 . . . 620 600										

1 Numero del campo. Numero asignado a cada campo de acuerdo con el CCF.

2 Mnemonico. Abreviatura de tres letras extraidas del titulo de cada campo.

3 Titulo del campo y definicion. El nombre se extrae del CCF asi como el tipo de informacion que se va a manejar en el mismo. La definicion se da en conformidad con el tipo de informacion que contiene el dato.

4 Representacion y extension (minima y maxima). Es la extension de letras, digitos o simbolos necesarios para asentar el dato asi como el tipo de dato del que se trata; es decir, numerico, alfabetico o alfanumerico.

5 Obligatoriedad. Si el dato es necesario y prioritario como para no continuar la captura hasta que este se indique, entonces se marca como obligatorio. Si el campo no es obligatorio entonces es un campo optativo.

6 Repetitividad. Es una variable que indica el numero de veces que se va a repetir un campo. Si el campo no es repetible entonces es un campo simple.

7 Indicador de subcampos. Es una lista de los componentes de un dato que se pretende tratar de manera desglozada.

8 Seguridad. Es la especificacion del tipo y los niveles de acceso permitidos a los usuarios del sistema; es decir, la definicion de sus privilegios.

9 Indexado. Nos indica si el campo contendra terminos que deben quedar incluidos en el diccionario de vocabulario controlado.

10 Ejemplo ilustrativo. Estructura y contenido del dato asentado en el campo.

11 Notas. Es una variable que indica observaciones y sugerencias para el codificador, durante el analisis del dato y su representacion en la maquina.

A continuacion se detallan cada uno de los campos que forman parte del formato de codificacion de la la base de datos de tesis de la Facultad de Ingenieria.

Lista de campos incluidos

```
    001 IDENTIFICADOR DEL REGISTRO
    020 FUENTE DEL REGISTRO
    022 FECHA DE CAPTURA
    040 IDIOMA DEL TEXTO
    060 TIPO DE MATERIAL
    200 TITULO
  * 250 GRADO ACADEMICO
    300 NOMBRE DEL AUTOR
 ** 300 NOMBRE DEL ASESOR
    310 INSTITUCION
  * 340 GRADO DEL ASESOR
    400 LUGAR DE PUBLICACION
    440 FECHA DE PUBLICACION
    460 DESCRIPCION FISICA
    600 RESUMEN
    620 DESCRIPTORES
  * 700 NUMERO DE REGISTRO DEL TEXTO
  * 710 FECHA DE INICIO
  * 720 FECHA DE FINALIZACION
  * 730 GRADO DE AVANCE
```

NOTAS:

Los campos marcados con un asterisco no pertenecen al CCF, su unica posible incorporacion seria incluyendolos como una nota en el campo 500. Para esta base de datos, consideramos necesario identificarlos como un campo por separado, por lo tanto se agregaron estos numeros intercalados con los del CCF.

El campo con doble asterisco esta repetido, dado que el campo 300 del CCF indica "Nombre de persona" sin diferenciar al autor del asesor en el caso de la base de datos de tesis de la Facultad de Ingenieria.

CAMPO 001

1: ETIQUETA 001

2: NOMBRE Identificador del registro

3: MNEMONICO REG

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	LONGITUD: MAXIMA	TIPO:
___04	___04	__ ALFABETICO
		_X NUMERICO
		__ ALFANUMERICO

5: OBLIGATORIO SI

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO ACCESO DIRECTO

10: EJEMPLO 001:0001
001:0237

11: NOTAS

CAMPO 020

1: ETIQUETA 020

2: NOMBRE Fuente del registro

3: MNEMONICO FUE

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	MAXIMA	TIPO:
___01	___10	__ ALFABETICO
		__ NUMERICO
		_X ALFANUMERICO

5: OBLIGATORIO SI

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO NO

10: EJEMPLO 020:UNAM-FI

11: NOTAS Deben utilizarse abreviaturas o tablas para su representacion codificada.

CAMPO 022

1: ETIQUETA 022

2: NOMBRE Fecha de captura

3: MNEMONICO FEC

4: REPRESENTACION Y EXTENSION

LONGITUD:		TIPO:
MINIMA	MAXIMA	__ ALFABETICO
___08	___08	_X NUMERICO
		__ ALFANUMERICO

5: OBLIGATORIO SI

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO NO

10: EJEMPLO 022:19861217
022:19860000

11: NOTAS En caso de que no se conozca el mes o el dia, se deben sustituir por ceros.

CAMPO 040

1: ETIQUETA 040

2: NOMBRE Idioma del texto

3: MNEMONICO IDI

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	MAXIMA	TIPO:
___03	___11	__ ALFABETICO
		__ NUMERICO
		_X ALFANUMERICO

5: OBLIGATORIO SI

6: REPETIBLE 3 veces

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO NO

10: EJEMPLO 040:SPA
040:SPA,ENG

11: NOTAS Las repeticiones se deben separar por una coma. Las abreviaturas estan dadas por el CCF.

CAMPO 060

1: ETIQUETA 060

2: NOMBRE Tipo de material

3: MNEMONICO MAT

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	LONGITUD: MAXIMA	TIPO:
___03	___03	__ ALFABETICO
		_X NUMERICO
		__ ALFANUMERICO

5: OBLIGATORIO SI

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO NO

10: EJEMPLO 060:110
060:900

11: NOTAS Los numeros validos segun CCF son:

100 Textual
105 Reporte/reporte tecnico
110 Tesis/disertacion
115 Documento de una reunion
120 Publicacion seriada
125 Periodico
130 Anuario
145 Series irregulares
150 Series monograficas
900 Otro

CAMPO 200

1: ETIQUETA 200

2: NOMBRE Titulo

3: MNEMONICO TIT

4: REPRESENTACION Y EXTENSION

LONGITUD:		TIPO:
MINIMA	MAXIMA	
___01	__200	__ ALFABETICO
		__ NUMERICO
		_X ALFANUMERICO

5: OBLIGATORIO SI

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO POR PALABRAS

10: EJEMPLO 200:Sistemas recuperadores de Informacion bibliografica.
200:Analisis del transporte de informacion entre computadoras mediante cintas magneticas.

11: NOTAS

CAMPO 250

1: ETIQUETA 250

2: NOMBRE Grado academico

3: MNEMONICO GRD

4: REPRESENTACION Y EXTENSION

LONGITUD:		TIPO:
MINIMA	MAXIMA	
___01	___01	_X ALFABETICO
		__ NUMERICO
		__ ALFANUMERICO

5: OBLIGATORIO SI

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO SI

10: EJEMPLO 250:L

11: NOTAS Los datos de este campo se encuentran en forma codificada de acuerdo a la siguiente lista:

L - LICENCIATURA
M - MAESTRIA
D - DOCTORADO
P - POSTDOCTORADO
E - ESPECIALIZACION

CAMPO 300

1: ETIQUETA 300

2: NOMBRE Nombre del autor

3: MNEMONICO AUT

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	LONGITUD: MAXIMA	TIPO:
01	500	___ ALFABETICO
		___ NUMERICO
		_X ALFANUMERICO

5: OBLIGATORIO SI

6: REPETIBLE 10 veces

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO POR PALABRAS

10: EJEMPLO
300:Zugasti, M.
300:Sanchez, A.%Perez, L.

11: NOTAS Las repeticiones se separan por el signo de porcentaje. El nombre debera comenzar por el apellido paterno, una coma, la inicial del nombre, punto.

CAMPO 300+

1: ETIQUETA 300+

2: NOMBRE Nombre del asesor

3: MNEMONICO ASE

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	MAXIMA	TIPO:
___01	__200	__ ALFABETICO
		__ NUMERICO
		_X ALFANUMERICO

5: OBLIGATORIO SI

6: REPETIBLE 4 veces

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO POR PALABRAS

10: EJEMPLO 300:Castro, S.
300:Cordero, L.%Sandoval, D.

11: NOTAS Las repeticiones deben separarse por un signo de porcentaje.

CAMPO 310

1: ETIQUETA	310
2: NOMBRE	Institucion
3: MNEMONICO	INS
4: REPRESENTACION Y EXTENSION	LONGITUD: MINIMA 01 MAXIMA 100 TIPO: ___ ALFABETICO ___ NUMERICO _X ALFANUMERICO
5: OBLIGATORIO	SI
6: REPETIBLE	NO
7: SUBCAMPOS	A,B,C A - UNIVERSIDAD B - FACULTAD O ESCUELA C - CARRERA Y MODULO
8: SEGUDAD	NINGUNA
9: INDEXADO	POR SUBCAMPOS
10: EJEMPLO	310:&aUNAM&bFI&cMECANICA 310:&cCOMPUTACION
11: NOTAS	Los subcampos se indican inmediatamente despues de un simbolo de 'ampersand' (&).

CAMPO 340

1: ETIQUETA 340

2: NOMBRE Grado del asesor

3: MNEMONICO GRA

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	LONGITUD: MAXIMA	TIPO:
___01	___01	_X ALFABETICO
		__ NUMERICO
		__ ALFANUMERICO

5: OBLIGATORIO NO

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO NO

10: EJEMPLO 340:D

11: NOTAS La tabla de datos validos es identica a la del campo 250. En caso de que existan varios asesores, el grado se debera codificar entre parentesis despues del nombre dejando en blanco el campo 340, por ejemplo:
300:GOMEZ, M.(D)%RAMIREZ, A.(L)
340:

CAMPO 400

1: ETIQUETA 400

2: NOMBRE Lugar de publicacion

3: MNEMONICO LUG

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	LONGITUD: MAXIMA	TIPO:
___01	___20	__ ALFABETICO
		__ NUMERICO
		_X ALFANUMERICO

5: OBLIGATORIO NO

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO NO

10: EJEMPLO 400:Mexico

11: NOTAS

CAMPO 440

1: ETIQUETA 440

2: NOMBRE Fecha de publicacion

3: MNEMONICO FPB

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	LONGITUD: MAXIMA	TIPO:
___08	___08	__ ALFABETICO
		_X NUMERICO
		__ ALFANUMERICO

5: OBLIGATORIO NO

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO NO

10: EJEMPLO 440:19840917
440:19870200

11: NOTAS En caso de que no se conozca el mes o el dia, se deben sustituir por ceros.

CAMPO 460

1: ETIQUETA 460

2: NOMBRE Descripcion fisica

3: MNEMONICO FIS

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	LONGITUD: MAXIMA	TIPO:
___01	___40	__ ALFABETICO
		__ NUMERICO
		_X ALFANUMERICO

5: OBLIGATORIO NO

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO NO

10: EJEMPLO 460:276p.
460:120p. [xi]

11: NOTAS Pueden indicarse las dimensiones, numero de paginas, ilustraciones, anexos, indices, etc.

CAMPO 600

1: ETIQUETA 600

2: NOMBRE Resumen

3: MNEMONICO RES

4: REPRESENTACION Y EXTENSION

LONGITUD:		TIPO:
MINIMA	MAXIMA	__ ALFABETICO
___01	_2000	__ NUMERICO
		_X ALFANUMERICO

5: OBLIGATORIO NO

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO POR PALABRAS

10: EJEMPLO 600:Un sistema recuperador de informacion es un sistema que se utiliza para almacenar documentacion que debera procesarse, buscarse, recuperarse y diseminarse para diferentes poblaciones de usuarios. Los sistemas ...

11: NOTAS

CAMPO 620

1: ETIQUETA 620

2: NOMBRE Descriptores

3: MNEMONICO DES

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	LONGITUD: MAXIMA	TIPO:
___01	___40	__ ALFABETICO
		__ NUMERICO
		_X ALFANUMERICO

5: OBLIGATORIO SI

6: REPETIBLE 15 veces

7: SUBCAMPOS NINGUNO

8: SEGURIDAD NINGUNA

9: INDEXADO POR FRASES

10: EJEMPLO 620:America Latina
620:Motores de induccion%flujo magnetico.

11: NOTAS Los descriptores se separan con el signo de porcentaje despues de terminar la frase o la palabra.

CAMPO 700

1: ETIQUETA 700

2: NOMBRE Numero de registro del texto

3: MNEMONICO NUM

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	MAXIMA	TIPO:
01	05	__ ALFABETICO
		_X NUMERICO
		__ ALFANUMERICO

5: OBLIGATORIO NO

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD PARA CONTROL INTERNO

9: INDEXADO ACCESO DIRECTO

10: EJEMPLO 700:8672

11: NOTAS

CAMPO 710

1: ETIQUETA 710

2: NOMBRE Fecha de inicio

3: MNEMONICO INI

4: REPRESENTACION Y EXTENSION

LONGITUD:		TIPO:
MINIMA	MAXIMA	__ ALFABETICO
___08	___08	_X NUMERICO
		__ ALFANUMERICO

5: OBLIGATORIO NO

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD PARA CONTROL INTERNO

9: INDEXADO NO

10: EJEMPLO 710:19850600

11: NOTAS En caso de que no se conozca el mes o el dia, se deben sustituir por ceros.

CAMPO 720

1: ETIQUETA 720

2: NOMBRE Fecha de finalizacion

3: MNEMONICO FIN

4: REPRESENTACION Y EXTENSION

LONGITUD:		TIPO:
MINIMA	MAXIMA	__ ALFABETICO
___08	___08	_X NUMERICO
		__ ALFANUMERICO

5: OBLIGATORIO NO

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD PARA CONTROL INTERNO

9: INDEXADO NO

10: EJEMPLO 720:19870200

11: NOTAS En caso de que no se conozca el mes o el dia, se deben sustituir por ceros.

CAMPO 730

1: ETIQUETA 730

2: NOMBRE Grado de avance

3: MNEMONICO GRV

4: REPRESENTACION Y EXTENSION

LONGITUD: MINIMA	MAXIMA	TIPO:
___01	___03	___ ALFABETICO
		_X NUMERICO
		__ ALFANUMERICO

5: OBLIGATORIO NO

6: REPETIBLE NO

7: SUBCAMPOS NINGUNO

8: SEGURIDAD PARA CONTROL INTERNO

9: INDEXADO NO

10: EJEMPLO 730:80

11: NOTAS Debe especificarse en forma de porcentaje respecto al trabajo total.

HOJA DE CODIFICACION

BASE DE DATOS DE TESIS DE LA F.I.

001 Identificador del registro

022 fecha de captura

020 fuente del registro

060 Tipo de material

040 Idioma del texto

700 Numero de registro

710 Fecha de inicio

720 Fecha de finalizacion

730 Grado de avance

NOMBRE DEL AUTOR	300	
Grado Academico	250	
NOMBRE DEL ASESOR	300+	
Grado del asesor	340	
TITULO	200	
INSTITUCION	310	
Lugar de Publicacion	400	
Fecha de Publicacion	440	
Descripcion Fisica	460	
Descriptores	620	
RESUMEN	600	

```
BUSQUEDA:

     Set  1:BASE$

     P=          4    BASES
     P=          2    BASES DE DATOS
     T=          2  - #3: BASE$
     T=          2  - #1: #3

DESPLIEGUE:

REGISTRO: 00001
REG 1           FEC 19870121          FUE UNAM-FI
MAT 110         IDI SPA               NUM 101
INI 19850600    FIN 19870115          GRV 100

AUT ZUGASTI, M.; SALAZAR, E.
GRD L
ASE CASTRO, S.
GRA L
TIT SISTEMAS RECUPERADORES DE INFORMACION BIBLIOGRAFICA
DES BASES DE DATOS; ARCHIVOS INVERITOS; INTELIGENCIA ARTIFICIAL;
     LENGUAJE NATURAL; LINGUISTICA; TESIS

REGISTRO: 00002
REG 3           FEC 19870121          FUE UNAM-FI
MAT 110         IDI SPA               NUM 2
INI 19860000    FIN 19870000          GRV 80

AUT CASTRO, S.
GRD L
ASE SANDOVAL, D.
GRA L
TIT BASES DE DATOS DISTRIBUIDAS CON BITACORAS REDUNDANTES
DES REDES LOCALES; PROTOCOLO; BASES DE DATOS; COMUNICACION

BUSQUEDA:

     Set  2:(CASTRO * SANDOVAL) + (LEYVA * ZUGASTI) ^ TRANSPORTE

     P=          3    CASTRO
     P=          1    SANDOVAL
     T=          1  - #3: CASTRO * SANDOVAL
     P=          1    LEYVA
     P=          1    ZUGASTI
     T=          0  - #4: LEYVA * ZUGASTI
     P=          2    TRANSPORTE
     T=          0  - #5: #4 ^ TRANSPORTE
     T=          1  - #6: #3 + #5
     T=          1  - #2: #6
```

```
DESPLIEGUE:

REGISTRO: 00003
REG 3           FEC 19870121        FUE UNAM-FI
MAT 110         IDI SPA             NUM 2
INI 19860000    FIN 19870000        GRV 80

AUT CASTRO, S.
GRD L
ASE SANDOVAL, D.
GRA L
TIT BASES DE DATOS DISTRIBUIDAS CON BITACORAS REDUNDANTES
DES REDES LOCALES; PROTOCOLO; BASES DE DATOS; COMUNICACION

BUSQUEDA:

     Set  3: (TOPO$ * BASE$) * DISTRIBUIDA$ ^ CASTRO

      P=          2    TOPOLOGIAS
      T=          2  - #4: TOPO$
      P=          4    BASES
      P=          2    BASES DE DATOS
      T=          2  - #6: BASE$
      T=          1  - #7: #4 * #6
      P=          2    DISTRIBUIDAS
      T=          1  - #8: DISTRIBUIDA$
      T=          1  - #9: #7 * #8
      P=          3    CASTRO
      T=          0  - #10: #9 ^ CASTRO
      T=          0  - #3: #10

NO HAY DESPLIEGUE DE INFORMACION


RESUMEN DE LAS BUSQUEDAS

   Set  Data Base  Hits Query element        Current Data Base name = TESIS
---  ---------- ----- --------------

  001  TESIS      00002 BASE$
  002  TESIS      00001 (CASTRO * SANDOVAL) + (LEYVA * ZUGASTI) ^ TRANSPORTE
003  TESIS      00000 (TOPO$ * BASE$) * DISTRIBUIDA$ ^ CASTRO
```

DATOS DE LA BASE DE DATOS DE TESIS DE LA

FACULTAD DE INGENIERIA

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

TIT ANALISIS DEL TRANSPORTE DE INFORMACION ENTRE
COMPUTADORAS MEDIANTE CINTAS MAGNETICAS
AUT MARTINEZ, E.,LEYVA, V.,,
DES CINTAS MAGNETICAS,PROTOCOLO,TRANSPORTE,COMUNICACIONES,,
RES En la actualidad podemos encontrar una gran cantidad de topologias para interconectar computadoras formando redes locales, pero se han visto serias desventajas para la verdadera comunicacion de computadoras. Esta comunicacion debe permitir la posibilidad de interpretar la informacion de computadoras de diferentes familias; es decir, heterogeneas entre si.

TIT BASES DE DATOS DISTRIBUIDAS CON BITACORAS REDUNDANTES
AUT CASTRO, S.,,
DES REDES LOCALES,PROTOCOLO,BASES DE DATOS,COMUNICACION,,
RES Las bases de datos distribuidas ofrecen mayor confiabilidad en cuanto al contenido de informacion, ya que cada centro dedicado controlara su porcion del sistema logico. Este trabajo describe algunas topologias de redes locales para permitir el acceso remoto de bases de datos, con el mismo concepto de un sistema recuperador de informacion en bitacoras redundantes.

TIT SISTEMAS RECUPERADORES DE INFORMACION BIBLIOGRAFICA
AUT ZUGASTI, M.,SALAZAR, E.,,
DES BASES DE DATOS,ARCHIVOS INVERTIDOS,INTELIGENCIA ARTIFICIAL,LENGUAJE NATURAL,LINGUISTICA,TESIS,,
RES Un sistema recuperador de informacion es un sistema que se utiliza para almacenar documentacion que debera procesarse, buscarse, recuperarse y diseminarse para diferentes poblaciones de usuarios. Los sistemas recuperadores de informacion comparten algunos conceptos con otros sistemas de informacion como lo son los administradores de bases de datos y los sistemas para toma de decisiones.

CONCLUSIONES

Los sistemas recuperadores de informacion son en la actualidad herramientas esenciales en todas las areas; ya que pemiten organizar y recuperar informacion bibliografica y/o textual en unos cuantos minutos, dandonos a conocer todos los escritos existentes de un area especifica; permitiendo almacenar estrategias y reportar o recibir las actualizaciones mensualmente mediante su servicio de diseminacion selectiva de informacion, proporcionando grandes beneficios.

Las estructuras de datos empleadas permiten optimizar el tiempo de respuesta de las busquedas, para los metodos frecuentemente usados. Mientras no se modifiquen los procesos del pensamiento humano para realizar estas tareas, los metodos actuales en la computadora seguiran siendo optimos; es decir, que se cuenta con la mejor representacion de los datos para los fines que se persiguen.

El futuro de los recuperadores de informacion al igual que otros sistemas se veran ampliamente beneficiados conforme se encuentren soluciones a las tecnicas avanzadas de recuperacion probabilistica, analisis de cumulos, conjuntos difusos e inteligencia artificial. Cuando los avances permitan la implantacion de un experto linguistico, haciendo uso de la inteligencia artificial para la seleccion o asociacion de terminos en una consulta, entonces se tendra un recuperador en lenguaje natural y se habra llegado al punto en que con un concepto vago se logre percibir el tema central de la consulta. Este experto linguistico es complementario y no competitivo con un experto artificial sobre el tema particular que se este consultando, lo cual encontraria los documentos mas relevantes a traves de guias y conceptos ligeramente estructurados y organizados. El primero estaria al nivel de un diccionario reporteador de definiciones, asociador de ideas o revisor gramatical. Mientras que, el segundo estaria al nivel de hacer una consultoria especializada.

El desarrollo e investigacion de cada una de las disciplinas que engloban la teoria de sistemas de informacion ofrece nuevas herramientas para la realizacion de sistemas mas eficaces y confiables.

El tener una base de datos de tesis de la facultad de ingenieria proporcionara muchas ventajas a la comunidad universitaria y sera un primer paso para automatizar la biblioteca, sirviendo ademas como experiencia para futuras bases de datos. Algunas de las ventajas mas sobresalientes son: la obtencion de informacion en forma agil, la posibilidad de elegir la tematica de investigacion o localizacion de fuentes de consulta para futuros trabajos, el proporcionar una vista global de los proyectos, temas y areas para el desarrollo uniforme de las investigaciones que se llevan a cabo, el hacer de conocimiento publico los trabajos que se realizan en la universidad, el dar a conocer la trayectoria de los temas de interes y muchas mas.

Si cada una de las facultades de la UNAM, aprovecharan la metodologia, el formato y el programa de manera que todos manejaran el mismo tipo de informacion, se podria tener una red de todas las tesis de la UNAM en cualquier area, teniendo con ello un sistema que podria ser consultado desde cualquier facultad; o bien, podria ofrecerse al publico a traves de SECOBI que es el Servicio de Consulta a Bancos de Informacion que es una subdireccion del CONACYT (Consejo Nacional de Ciencia Y Tecnologia), y con ello ayudar en la labor de crear bancos nacionales de informacion, evitando el tener que consultar informacion nacional a traves de sistemas extranjeros.

La construccion automatica de tesauros consume demasiados recursos tanto de la computadora, como humanos y economicos; asi como tiempo en el analisis y construccion del sistema; por lo que, un sistema de este tipo solo se justifica cuando se manejan grandes volumenes de informacion resultando un elemento de mucha ayuda.

El contar con un recuperador de tesis va a fomentar el interes en el desarrollo de este tipo de herramientas. Ademas de que sera una forma de tener actualizada a la comunidad universitaria en tecnologias de vanguardia, despertando con ello la labor de investigacion.

ANEXOS

ANEXO A

Herramientas para Procesar el Lenguaje de Consulta

INTRODUCCION

La estructuracion de mandatos en una linea de texto constituye la conceptualizacion de un lenguaje. Para interpretar cada mandato es necesario reconocer cada una de las palabras y operadores que intervienen. Para ello se requiere de dos tipos de analisis que son el lexical y el sintactico.

El analisis lexical separa el texto en los elementos que lo componen.

El analisis sintactico se encarga de ver las relaciones estructurales que existen entre estos elementos.

El analisis lexical y sintactico se puede llevar a cabo en forma independiente de la generacion o interpretacion.

El tipo de lenguaje y la extension del texto influyen en el analisis.

La unidad elemental del texto es el caracter.

Los caracteres se agrupan en simbolos o tokens, que son las unidades de lenguaje minimas que tienen significado.

El objetivo del analisis lexical es subdividir el texto en tokens e identificarlos dentro de un marco de referencia. A su vez el analisis sintactico verifica que la construccion del mandato sea adecuado al lenguaje.

Las estructuras sencillas simplifican la sintaxis de las instrucciones. La jerarquia posicional de los operadores exhibe la estructura sintactica, de tal forma que permite una interpretacion de los comandos casi directa.

A pesar de que los requerimientos de varios traductores difieren en extension, la naturaleza de los analisis lexical y sintactico es casi el mismo para todos.

Dependiendo de la extension de la gramatica el identificador y la relacion pueden ser simbolos terminales, en una gramatica reducida o no terminales, en una gramatica extensa. Para indicar este doble papel que tienen los terminales se les denomina semi-terminales.

Una gramatica puede utilizarse para los siguientes propositos:
- En la generacion de cadenas en el lenguaje.
- Para determinar la estructura y validez de una cadena dada.

A la aceptacion o rechazo de una cadena como miembro del lenguaje se le llama reconocimiento.

A la determinacion de la estructura gramatical se le conoce como analizador (parsing).

El reconocimiento de los errores sintacticos en un texto no ayuda al usuario, siendo mas conveniente que el analisis de validez o error indique el tipo de errores en que se incurre.

Por esta razon durante la traduccion del texto se utiliza un analizador en vez de un reconocedor.

El analizador de un lenguaje generalmente puede contar con un analizador lexical para reducir cadenas de caracteres a identificadores, constantes, palabras reservadas y simbolos de operacion. La tarea de un analizador es aplicar una gramatica a un texto en lenguaje original y determinar el arbol sintactico; sin embargo, no todos los analisis describen el arbol directamente.

Los metodos de analisis pueden dividirse en dos grandes grupos, cuyos nombres indican la direccion que se sigue en la construccion del arbol de analisis (arbol de parse):
(definir un arbol, simbolo distinguido, etc.)
- El analisis de lo general a lo particular (top-down parsing) principia con el simbolo distinguido en la raiz que por lo general se coloca en el tope del arbol y avanza hacia abajo a la cadena de terminales en las hojas.
- El analisis de lo particular a lo general (bottom-up parsing) principia a partir de los simbolos terminales en las hojas y avanza hacia arriba a la raiz del arbol.

PRECEDENCIA

Debido a que el texto de un lenguaje es casi invariablemente unidimensional; es decir de grado uno, en donde grado es el numero de argumentos requeridos por una funcion, existen tres formas de colocar un operador binario con respecto a sus argumentos. Cada uno de ellos sigue una convencion notacional cuyo nombre refleja la colocacion del operador.
- En la notacion prefija el operador precede a ambos argumentos.
- En la notacion postfija el operador sigue a ambos argumentos
- En la notacion infija el operador aparece entre sus argumentos.

Cuando el operador es unario, este precede a su argumento en la notacion prefija y sigue al argumento en notacion postfija. En la notacion infija se pueden tomar dos decisiones debido a la simetria que existe. Aunque existe la convencion de que un operador infijo unario precede a su argumento.

La notacion prefija tiene la ventaja de ser similar a la notacion utilizada en las matematicas tradicionales, en donde el resultado de aplicar la funcion f a sus argumentos a y b se escribe como f(a,b), y donde la composicion de funciones se escribe f(h(x)).

A las notaciones prefija y postfija frecuentemente se les llama notaciones polacas. La notacion polaca permite utilizar mas de dos argumentos lo cual no es posible con la notacion infija; sin embargo, la notacion infija ha sido adoptada virtualmente para todos los lenguajes de recuperacion porque resulta muy familiar.

La determinacion del grado de un operador requiere de un analisis sintactico, pero debido a que este analisis es muy sencillo, puede ejecutarse facilmente junto con el analisis lexical.

El orden en el que se aplican los operadores depende de la prioridad o precedencia de las diferentes ocurrencias de los operadores.

La precedencia tiene dos aspectos: el inherente y el posicional.

La precedencia inherente es aquella otorgada por la prioridad que tienen unos operadores sobre otros, como es el caso de la multiplicacion y la suma; en donde la multiplicacion tiene precedencia inherente sobre la suma, siendo esta la razon por la que se ejecuta primero la multiplicacion sin importar su posicion.

Cuando los operadores tienen una precedencia inherente igual, el algoritmo les asigna una prioridad posicional (generalmente de izquierda a derecha).

Una gramatica de operadores es aquella en la que ninguna regla tiene la forma:

a=..bc..;

donde b y c son simbolos no terminales.

Esta restriccion garantiza que en ninguna etapa de la generacion o analisis del lenguaje puede existir alguna cadena que contenga dos simbolos terminales adyascentes.

Entre dos simbolos terminales cualquiera de una gramatica de operacion, se pueden tener una o mas de las tres relaciones de precedencia importantes. Cuando en una gramatica de operadores un par arbitrario de terminales mantienen mas de una de las tres relaciones de precedencia, entonces la gramatica es una "gramatica con precedencia de operadores".

Las relaciones de precedencia que se tienen en cada par de terminales se puede representar mediante una matriz que se conoce como matriz de precedencia.

Para una gramatica grande, la matriz de precedencia puede tener el inconveniente de ser muy grande. Desafortunadamente existen matrices de precedencia para las cuales no se tienen funciones apropiadas de precedencia doble; sin embargo, para la mayoria de los lenguajes representables por gramaticas de precedencia de operadores es posible construir funciones de precedencia doble.

ANALISIS LEXICO:

La primera tarea de un analisis lexical es agrupar caracteres en tokens y verificar que los simboles terminales de cada token pertenezcan a la gramatica.

Los simbolos terminales con frecuencia se clasifican lexicalmente de acuerdo a su proposito en grupos llamados tipos. La longitud de un tipo puede variar conforme a su origen. dLos tipos mas grandes son las palabras, las constantes y los operadores, cada uno de los cuales puede dividirse en subtipos. Una palabra puede ser una palabra reservada, una palabra clave o un identificador.

Una constante puede ser numerica o no. Existen distinciones mas detalladas dentro de cada uno de estos subtipos. Los operadores incluyen entre otros, operadores aritmeticos, simbolos de agrupacion como parentesis y las marcas de puntuacion que se utilizan para separar listas de operandos.

El analizador lexical determina por lo menos el tipo al que pertenece cada token, y con excepcion de algunas palabras puede reconocer el subtipo al que corresponde. Ademas de proporcionar la identidad del token, entrega al analizador sintactico el token mismo o una direccion dentro de una tabla. Pudiendo existir las siguientes tablas: para palabras, para constantes numericas, para constantes de cadenas de caracteres y para operadores; o tal vez, se utilice una sola tabla que contenga todo.

La determinacion de palabras reservadas puede ser ejecutada por el explorador o por el analizador, para lo cual debe verificarse cada palabra contra una tabla de palabras reservadas. Debido a que los limites entre el analisis lexical y sintactico no es fijo la division del trabajo entre el explorador y el analizador es un tanto arbitraria.

Para identificar y clasificar tokens, es necesario que el explorador aisle unos de otros. El principio y el fin de cada token debe ser reconocible. La libertad que se tiene en el formato del texto original afecta sustancialmente el aislamiento de los tokens. Cuando las posiciones de los tokens es fija entonces no se necesita hacer una verificacion de los caracteres para determinar el principio. Similarmente, si la longitud de los tokens es fija, el final de cada ellos tambien puede identificarse sin tener que verificar caracteres. Algunas veces uno o mas tokens sirven como delimitadores que marcan la extension de aquellos que son de longitud variable.

En un lenguaje con formato libre, generalmente resulta necesario ejecutar el aislamiento de tokens concurrentemente con su identificacion.

En muchos lenguajes el espacio sirve como delimitador, de hecho cuando se permite un espacio generalmente muchos espacios producen el mismo efecto.

Muchos lenguajes permiten comentarios, los cuales deben eliminarse del texto antes de que se analice sintacticamente.

La facilidad para reconocer los comentarios varia inversamente con la libertad permitida para insertarlos. Por ejemplo el reconocimiento de un comentario en lenguajes en donde un caracter especifico se coloca en una posicion especifica para indicar que toda la linea es un comentario (es un formato fijo), es muy sencilla. El uso de caracteres reservados hace que el reconocimiento de los comentarios sea mas facil que si se utilizaran caracteres multiproposito como delimitadores del token. Generalmente los tokens dentro de un comentario son completamente ignorados.

ANEXO B

[illegible]

```
program btrieve; {B:TREE+}
const
  maxkeylen = 20;
  order     =  5;
  roto      =  3;

type
  cad  =string[maxkeylen];
  items=record
          map,times:integer;
          key:string[maxkeylen];
        end;
  nodes=record
          page:array [1..order] of items;
          path:array [0..order] of integer;
          used:integer;
        end;

  lista_ligada=^lista;
  lista=record
          num:integer;
          sig,ant:lista_ligada;
        end;

var
  bkpt         :lista_ligada;
  btree        :file of nodes;
  ch           :char;
  fin,pantalla:boolean;
  sal          :text;
```

```
        procedure blist;
        var
          j,jj    :integer;
          b       :nodes;

        begin
          if filesize(btree)>0
            then
              begin
                seek(btree,0);
                jj:=0;
                repeat
                  read(btree,b);
                  with b do
                    begin
                      writeln(sal,'++++ R: ',jj:3,'':3,'U: ',used);
                      for j:=1 to b.used do
                      writeln(sal,j:3,'':3,path[j-1]:3,'':3,path[j]:3,'':3,page
          end;
        writeln(sal);
        jj:=succ(jj);
      until eof(btree);
      writeln(sal,'-------------------------------------------------------');
    end
  else writeln(sal,'=-=-=-=-= Archivo vacio =-=-=-=-=');
end;

procedure get_reg(dp:integer; var buffer:nodes);
begin
  seek(btree,dp);
  read(btree,buffer);
end;

procedure put_reg(dp:integer; buffer:nodes);
begin
  seek(btree,dp);
  write(btree,buffer);
end;

procedure captura;
type
  stack_type=record
               key:string[maxkeylen];
               ant,pos,bak:integer;
             end;

var
  b,bn,b_aux: nodes;
  palabra   : cad;
  i,dsk_ptr,
  stk,old_dsk
            : integer;
  stack     : array[1..2] of stack_type;
```

```
function lee_lista(p:lista_ligada):integer;
begin
  if p^.sig=nil
    then lee_lista:=p^.num
    else
      if p^.sig^.num < 0
        then lee_lista:=p^.num
        else lee_lista:=lee_lista(p^.sig);
end;

procedure mete_lista(var p:lista_ligada);
begin
  if p=nil
    then
      begin
        new(p);
        p^.num:=dsk_ptr;
        p^.sig:=nil;
      end
    else
      if p^.num < 0
        then p^.num:=dsk_ptr
        else mete_lista(p^.sig);
end;

procedure saca_lista(var p:lista_ligada);
begin
  if p^.sig=nil
    then p^.num:=-1
    else
      if p^.sig^.num < 0
        then p^.num:=-1
        else saca_lista(p^.sig);
end;

{ 1 }
  procedure mete_dato;
  begin
    with b.page[i] do
      begin
        times:=1;
        map:=0;
        key:=palabra;
      end;
    put_reg(dsk_ptr,b);
  end;
```

```
{ 2 }
procedure crea_reg;
begin
  for i:=1 to order do
    begin
      with b.page[i] do
        begin
          times:=0;
          map:=0;
          key:='';
        end;
      b.path[i]:=-1;
    end;
  b.path[0]:=-1;
  b.used    :=1;
  i         :=1;
  mete_dato;
end;

{ 3 }
procedure recorre(desde,valor:integer);
var
  j : integer;
begin
  for j:=b.used downto desde do
    begin
      b.page[j+1]:=b.page[j];
      b.path[j+1]:=b.path[j];
    end;
  b.path[desde]        := b.path[desde-1];
  b.page[desde].key    := stack[valor].key;
  b.page[desde].map    := 0;
  b.page[desde].times  := 1;
  b.path[desde-1]      := stack[valor].ant;
  b.path[desde]        := stack[valor].pos;
  b.used               := succ(b.used);
end;
```

```
{ 4 }
procedure encuentra_pos;
var
  ok:boolean;
begin
  i:=1;
  ok:=false;
  while (i<=b.used) and not ok do
    begin
      if b.page[i].key < palabra
        then i:=succ(i)
        else ok:=true;
    end;
  if b.path[i-1]<>-1
    then
      begin
        mete_lista(bkpt);
        dsk_ptr:=b.path[i-1];
        get_reg(dsk_ptr,b);
        encuentra_pos;
      end;
end;

{ 5 }
procedure copia;
var
  k:integer;
begin
  for k:=(roto+1) to order do
    begin
      bn.page[k-roto]:=b.page[k];
      bn.path[k-roto]:=b.path[k];
    end;
  bn.path[0]:=b.path[roto];
  b.used:=roto-1;
  bn.used:=order-roto;
end;

{ 6 }
procedure elige(chango:integer);
var
  ok:boolean;
begin
  i:=1;
  ok:=false;
  while (i <= b.used) and not ok do
    begin
      if b.page[i].key < stack[chango].key
        then i:=succ(i)
        else ok:=true;
    end;
  recorre(i,chango);
end;
```

```
{ 7 }
procedure raiz;
begin
  if dsk_ptr = old_dsk
    then
      begin
        bn.used           := 1;
        bn.page[1].key    := stack[stk].key;
        bn.page[1].map    := 0;
        bn.page[1].times  := 1;
        bn.path[0]        := filesize(btree);
        bn.path[1]        := stack[stk].pos;
        put_reg(0,bn);
        put_reg(filesize(btree),b );
        stk:=pred(stk);
      end;
end;

{ 8 }
procedure mete;
var k:integer;

procedure hay_espacio;
begin
  elige(stk);
  put_reg(dsk_ptr,b);
  stk:=pred(stk);
  saca_lista(bkpt);
end;

procedure no_hay_espacio;
begin
  stk:=succ(stk);
  stack[stk].key:=b.page[roto].key;
  stack[stk].ant:=dsk_ptr;
  stack[stk].pos:=filesize(btree);
  copia;
  if stack[stk-1].key < stack[stk].key
    then elige(stk-1)
    else begin b_aux:=b; b:=bn; elige(stk-1); bn:=b; b:=b_aux; end;
  put_reg(stack[stk].ant,b);
  put_reg(stack[stk].pos,bn);
  stack[stk-1]:=stack[stk];
  stk:=pred(stk);
  old_dsk:=dsk_ptr;
  dsk_ptr:=lee_lista(bkpt);
  saca_lista(bkpt);
  if stk > 0
    then
      begin
        get_reg(dsk_ptr,b);
        if dsk_ptr = 0 then raiz;
        mete;
      end;
end;
```

```
begin
  if stk > 0
    then
      if b.used < order
        then hay_espacio
        else no_hay_espacio;
end;

begin
  repeat
    write('palabra: ');
    readln(palabra);
    dsk_ptr:=0;
    if palabra<>''
      then
        begin
          mete_lista(bkpt);
          if filesize(btree) > 0
            then
              begin
                get_reg(dsk_ptr,b);
                encuentra_pos;
              end
            else fillchar(b,sizeof(b),0);
          stk:=1;
          stack[stk].key:=palabra;
          stack[stk].ant:=-1;
          stack[stk].pos:=-1;
          mete;
    end;
  until palabra='';
end;

procedure busqueda;
begin
end;

procedure salida;
begin
  pantalla:=not pantalla;
  close(sal);
  if pantalla
    then assign(sal,'con:')
    else assign(sal,'lst:');
  rewrite(sal);
end;
```

```
procedure menu;
begin
  writeln('- 1 - Captura');
  writeln('- 2 - Busca');
  writeln('- 3 - Lista');
  if pantalla
    then writeln('- 4 - PANTALLA / impresora')
    else writeln('- 4 - pantalla / IMPRESORA');
  writeln;
  writeln('- 0 - F I N');
  writeln;
  write('? ');
  repeat
    read(kbd,ch);
  until ch in ['0'..'4'];
  writeln(ch);
  case ch of
    '0':fin:=true;
    '1':captura;
    '2':busqueda;
    '3':blist;
    '4':salida;
  end;
end;

procedure abre;
begin
  assign(btree,'btree.ndx');
  (*$i-*)
  reset(btree);
  (*$i+*)
  if ioresult<>0
    then rewrite(btree)
    else
      begin
        write('Deseas borrar el archivo anterior');
        repeat
          read(kbd,ch);
          ch:=upcase(ch);
        until ch in ['S','N'];
        writeln(ch);
        if ch = 'S'
          then
            begin
              close(btree);
              rewrite(btree);
            end;
      end;
end;

procedure cierra;
begin
  close(btree);
end;
```

```
{------------ M A I N -------------}
begin
  bkpt:=nil;
  pantalla:=true;
  assign(sal,'con:');
  rewrite(sal);
  fin:=false;
  abre;
  repeat
    clrscr;
    menu;
    if not fin
      then
        begin
          write('..Oprime cualquier tecla..');
          read(kbd,ch);
        end;
  until fin;
  cierra;
end.
```

ANEXO C

Programa Recuperador con Archivos Invertidos

```
program INVERTIDOS;
type
  ptr_reg=^registr;
  registr=record
            rg:integer;
            nx:ptr_reg;
          end;

  liga=^caja;
  caja=record
         tx:string[15];
         cn:integer;
         sg:liga;
         rc:ptr_reg;
       end;

var
  raiz:                liga;
  palabra:             string[15];

procedure sort;
var
  root,pp:             liga;
  ocurrencia:          integer;

procedure tree(var p:liga);
begin
  if p=nil
    then
      begin
        new(p);
        p^.tx:=palabra;
        p^.cn:=ocurrencia;
        p^.sg:=nil;
        p^.rc:=nil;
      end
    else
      if p^.tx>palabra
        then
          begin
            new(pp);
            pp^.tx:=palabra;
            pp^.cn:=ocurrencia;
            pp^.sg:=p;
            pp^.rc:=nil;
            p:=pp;
          end
        else
          if p^.tx<palabra
            then tree(p^.sg)
            else writeln('*** existia una palabra repetida ***');
end;
```

```
procedure lectura(p:liga);
begin
  if p<>nil
    then
      begin
        palabra:=p^.tx;
        ocurrencia:=p^.cn;
        tree(root);
        lectura(p^.sg);
      end;
end;

begin
  root:=nil;
  lectura(raiz);
  mark(raiz);
  release(raiz);
  raiz:=root;
  mark(root);
  release(root);
end;

procedure lee(p:liga);
begin
  if p<>nil
    then
      begin
        writeln(p^.tx,'':20-length(p^.tx),p^.cn:2);
        lee(p^.sg);
      end;
end;

procedure escribe;
type
  inv=record
        wrd:string[15];
        occ:integer;
        ptr:integer;
      end;
var
  de:file of inv;
  rec_de:inv;
  ocurrencias:      integer;
```

```
procedure mete(var p:liga);
begin
  if p=nil
    then
      begin
        new(p);
        p^.tx:=palabra;
        p^.cn:=ocurrencias;
        p^.sg:=nil;
        p^.rc:=nil;
      end
    else mete(p^.sg);
end;

begin
  raiz:=nil;
  clrscr;
  writeln('Espere  P.F.');
  assign(de,'a:inverted.lst');
  reset(de);
  while not eof(de) do
  begin
    read(de,rec_de);
    palabra:=rec_de.wrd;
    ocurrencias:=rec_de.occ;
    mete(raiz);
  end;
  close(de);
end;

procedure recupera;
var
  text:              liga;
  result,etapa,
  truncacion,len:    integer;
  found:             boolean;
  comando:           string[20];

procedure busca(p:liga);
begin
  if p<>nil
    then
      begin
        if copy(p^.tx,1,truncacion) = palabra
          then
            begin
              writeln(p^.cn:5,'':5,p^.tx);
              result:=result+p^.cn;
              found:=true;
              text:=p;
              busca(p^.sg);
            end
          else busca(p^.sg);
      end;
end;
```

```
procedure com_ctrl;
begin
  comando:=copy(palabra,3,length(palabra)-2);
  writeln('COMANDO: ',comando);
end;

begin
  etapa:=1; palabra:=' ';
  repeat
    comando:='';
    writeln; write('?'); readln(palabra);
    if length(palabra)>2
      then if copy(palabra,1,2)='..'
             then com_ctrl;
    if comando=''
      then
        begin
          truncacion:=pos('?',palabra);
          if truncacion<>0
            then
              begin
                truncacion:=truncacion-1;
                len:=length(palabra);
                palabra:=copy(palabra,1,truncacion);
              end
            else
              begin
                len:=length(palabra);
                truncacion:=len+1;
              end;
          writeln;
          found:=false; result:=0;
          text:=nil; busca(raiz); writeln;
          if not found
            then writeln('TERMINO FUERA DEL DICCIONARIO: ',palabra);
          writeln('  *',etapa,'* ',result:7,' RESULTADOS');
          etapa:=etapa+1;
          writeln('  COMANDO, O ETAPA DE CONSULTA ',etapa:3);
        end;
  until (comando='st') or (comando='ST');
end;

begin
  escribe;
  sort;
  repeat
    recupera;
  until (comando='st') or (comando='ST');
end.
```

ANEXO D

Programa Generador de Indices y Diccionarios

```
Program retrieve;
type
  texto  =array [1..1000] of char;

  informa=texto;

  pointer=record
            reg:integer;
            nxt:integer;
            apn_ind:integer;
          end;

  index=  record
            wrd:string[15];
            cnt:integer;
            apn_pnt:integer;
          end;

  liga=   ^lista;

  lista=  record
            txt:string[15];
            apn:liga;
          end;

  p_lista_registros=^lista_registros;

  lista_registros=record
                    n:integer;
                    nx:p_lista_registros;
                  end;

  p_lista_invertida=^lista_invertida;

  lista_invertida=record
                    tx:string[15];
                    cn:integer;
                    sg:p_lista_invertida;
                    rc:p_lista_registros;
                  end;


var
  inf      :file of informa;
  ptr      :file of pointer;
  ndx      :file of index;
  root     :liga;
  invert   :p_lista_invertida;
  command  :char;
  quit     :boolean;
```

```
{_____L_I_S_T_____}
procedure list;
var
  i,j       :integer;
  info      :texto;
begin
  assign(inf,'b:texto.dat');
  {$i-} reset(inf); {$i+}
  j:=1;
  if ioresult=0
    then
      while not eof(inf) do
      begin
        clrscr;
        writeln('LIST: [',j:4,']');
        j:=j+1;
        read(inf,info);
        i:=1;
        while info[i]<>chr(0) do
        begin
          write(info[i]);
          i:=i+1;
        end;
        delay(2000);
      end;
  close(inf);
  clrscr;
end;

{_____A_P_P_E_N_D_____}
procedure append;
var
  info,blanco          :texto;
  ch          :char;
  i,p         :integer;
  linea       :string[80];
  palabra     :string[15];

procedure inversion;

procedure add_record(var pp:p_lista_registros);
begin
  if pp=nil
    then
      begin
        new(pp);
        pp^.n:=filesize(inf)-1;
      end
    else add_record(pp^.nx);
end;
```

```
procedure agrega(var p:p_lista_invertida);
begin
  if p<>nil
    then
      begin
        if p^.tx<>palabra
          then agrega(p^.sg)
          else
            begin
              p^.cn:=p^.cn+1;
              writeln('ya estaba la palabra,"',palabra,'"',
                      ' ahora son ',p^.cn,' ocurrencias');
              add_record(p^.rc);
            end;
      end
    else
      begin
        writeln(palabra,': es una palabra nueva en la lista');
        new(p);
        p^.tx:=palabra;
        p^.sg:=nil;
        p^.cn:=1;
        new(p^.rc);
        p^.rc^.n:=filesize(inf)-1;
        p^.rc^.nx:=nil;
      end;
end;

begin
  palabra:=' ';
  while palabra<>'' do
    begin
      gotoxy(1,22);
      clreol;
      write('palabra: ');
      readln(palabra);
      if palabra<>''
        then agrega(invert);
    end;
end;
```

```
procedure store_descriptores;
var
  iin:p_lista_invertida;
  inn:p_lista_registros;
  n_ptr,n_ndx:integer;
  rec:index;
  reg:pointer;

begin
  assign(ndx,'b:inverted.lst');
  rewrite(ndx);
  n_ndx:=0;

  assign(ptr,'b:pointer.lst');
  rewrite(ptr);
  n_ptr:=0;

  iin:=invert;
  while iin<>nil do
  begin
    rec.wrd:=iin^.tx;
    rec.cnt:=iin^.cn;
    rec.apn_pnt:=n_ptr;
    write(ndx,rec);

    inn:=iin^.rc;
    while inn<>nil do
    begin
      reg.reg:=inn^.n;
      if inn^.nx<>nil
        then reg.nxt:=n_ptr+1
        else reg.nxt:=-1;
      reg.apn_ind:=n_ndx;
      write(ptr,reg);
      inn:=inn^.nx;
      n_ptr:=n_ptr+1;
    end;

    iin:=iin^.sg;
    n_ndx:=n_ndx+1;
  end;
  close(ptr);
  close(ndx);
end;
```

```
    begin
      writeln('APPEND:');
      assign(inf,'b:texto.dat');
      ($i-) reset(inf); ($i+)
      if ioresult<>0
        then rewrite(inf)
        else begin seek(inf,filesize(inf)); end;
        fillchar(blanco,1000,' ');
      repeat
        clrscr;
        info:=blanco;
        p:=1;

        repeat
          write(p:3,': A+ ');
          readln(linea);
          if (linea<>'') and (linea<>'.')
            then
              for i:=1 to length(linea) do
                begin
                  info[p]:=linea[i];
                  p:=p+1;
                end;
          info[p]:=chr(10);
          p:=p+1;
          info[p]:=chr(13);
          p:=p+1;
        until (p>=900) or (linea='.');

        info[p]:=chr(0);
        write(inf,info);
        inversion;
        writeln('"RETURN" para continuar, "ESC" para finalizar la insercion
    read(kbd,ch);
  until ch=chr(27);
  close(inf);
  store_descriptores;
end;

{_____S_C_A_N_____}
procedure scan;
begin
  clrscr;
  if command in ['q','Q'] then quit:=true;
  if command in ['a','A'] then append;
  if command in ['l','L'] then list;
end;
```

```
procedure reservadas(var p:liga);
var
  palabra              :string[15];
  res                  :file of string[15];

procedure inserta(var p:liga);
begin
  if p=nil
    then
      begin
        new(p);
        p^.apn:=nil;
        p^.txt:=palabra;
      end
    else inserta(p^.apn);
end;

begin
  assign(res,'b:tokens.dat');
  {$i-} reset(res); {$i+}
  if ioresult=0
    then
      begin
        while not eof(res) do
          begin
            read(res,palabra);
            write(palabra:20);
            inserta(p);
          end;
        writeln;
        write('Si deseas agregar palabras, ');
      end
    else
      begin
        rewrite(res);
        writeln;
        write('Para iniciar el archivo, ');
      end;
  writeln('a continuacion, debes escribir las palabras');
  writeln('que quedaran EXCLUIDAS del archivo invertido (indizado)');
  writeln('..para finalizar, oprime "RETURN" en un renglon en blanco');
repeat
  readln(palabra);
  if palabra<>''
    then
      begin
        write(res,palabra);
        inserta(p);
      end;
until palabra='';
close(res);
end;
```

```
procedure invertidas(var p:p_lista_invertida);
var
  txt:string[15];
  rec:index;
  reg:pointer;
  i:integer;

procedure registros(var ppp:p_lista_registros);
var
  ii:integer;

procedure add_record(var p4:p_lista_registros);
begin
  if p4=nil
    then
      begin
        new(p4);
        p4^.n:=ii;
        p4^.nx:=nil;
      end
    else add_record(p4^.nx);
end;

begin
  ii:=i;
  assign(ptr,'b:pointer.lst');
  {$i-} reset(ptr); {$i+}
  if ioresult=0
    then
      while not (ii < 0) do
      begin
        seek(ptr,ii);
        read(ptr,reg);
        add_record(ppp);
        ii:=reg.nxt;
      end;
  close(ptr);
end;

procedure lista(var pp:p_lista_invertida);
begin
  if pp=nil
    then
      begin
        new(pp);
        pp^.tx:=rec.wrd;
        pp^.sg:=nil;
        pp^.rc:=nil;
        registros(pp^.rc);
        pp^.cn:=rec.cnt;
      end
    else lista(pp^.sg);
end;
```

```
begin
  assign(ndx,'b:inverted.lst');
  {$i-} reset(ndx); {$i+}
  i:=0;
  if ioresult=0
    then
      while not eof(ndx) do
      begin
        read(ndx,rec);
        write(rec.wrd:20);
        lista(p);
        i:=i+1;
      end;
  close(ndx);
end;

procedure l;
var
  r:pointer;
  i:integer;
begin
  i:=0;
  writeln('rec  apn_ind  nxt  reg');
  writeln('----------------------');
  assign(ptr,'b:pointer.lst');
  {$i-} reset(ptr); {$i+}
  if ioresult=0
    then
      BEGIN
        while not eof(ptr) do
        begin
          read(ptr,r);
          writeln(i:5,r.apn_ind:7,r.nxt:5,r.reg:5);
          i:=i+1;
        end;
        writeln('"RETURN" para continuar');
        readln;
      END;
  close(ptr);
end;
```

```
    procedure m;
    var
      i:integer;
      r:index;
    begin
      i:=0;
      writeln('rec      wrd                cnt  apn_pnt');
      writeln('-------------------------------------');
      assign(ndx,'b:inverted.lst');
      {$i-} reset(ndx); {$i+}
      if ioresult=0
        then
          begin
            while not eof(ndx) do
            begin
              read(ndx,r);
              writeln(i:5,'   ',r.wrd,'':15-length(r.wrd),r.cnt:5,r.apn_pnt:5)
          i:=i+1;
        end;
        writeln('"RETURN" para continuar');
        readln;
      end;
  close(ndx);
end;

{_____________________________M_A_I_N_____________________________}
begin
  root:=nil;
  reservadas(root);
  l;
  m;
  invert:=nil;
  invertidas(invert);
  quit:=false;
  repeat
    write('+ ');
    readln(command);
    scan;
  until quit;
end.
```

BIBLIOGRAFIA

1.- Martin, James.

2.- LeFkovitz, David. Data managment for on-line systems. 1974.

3.- Britton Lee Inc. Notas. 1982.

4.- CONACYT. "Manual ISDS". 1984.

5.- Gregory, D.; Joseph, J.; Cogan, J. "The DMS II primer". 1982.

6.- UNESCO. CCF: "Formato de comunicacion comun". 1984.

7.- Telesystems QUESTEL. "Manuel d'utilisateur". Paris.

8.- SDC ORBIT. ORBIT user manual. USA.

9.- Salton,G.; Mc Gill, M. "Introduction to Modern Information Retrieval". Mc Graw Hill, 1983.

10.- Wiederhold, Gio. "Database Design". Mc Graw Hill, 1983.

11.- Keren, Carl. "International Inventory of Software Packages in the Information Field". Paris:UNESCO, 1983

12.- Barret, William A.; Goudi, John D. "Compiler construction: Theory and practice". Chicago: Science Research Associates, Inc., 1979. 661 p.

13.- Lee, John A.N. "The anatomy of a compiler". 2nd edition New York: d. Van Nostrand Company, 1974. 470 p.

14.- Tremblay, Jean-Paul; Sorenson, Paul. "An introduction to data structures with applications". 2nd edition. Auckland: Mc. Graw Hill, 1984. 861 p.

15.- Coelho, Helder; Cotta, Jose Carlois; Morriz Pereyra, Luis. "How to solve it with prolog". 4th edition Lisboa: Ministerio do Equipamento Social; Laboratorio Nacional de Engenieria Civil, 1985. 215 p.

16.- Byte: The small systems journal. Phenix: Mc Graw Hill, 1986. 11(12).

17.- PC Magazine: The independant guide to IBM-Standard personal computing. New York: Ziff-Davis Publishing Co., 1986. 5(20)

18.- Information specialisee et banques de donees. 2a. Francia: Centre National Recherche Scientifique. pp. 19-37.

19.- Teleinformatica en Francia. Boletin especial. Mexico: Centro Frances de Informacion Tecnica e Industrial. 32 p.

20.- Seminario Conjunto Mexico/Francia sobre Bases y Bancos de Datos. SPP: Coordinacion General de los Servicios Nacionales de Estadistica, Geografia e Informatica.; Embajada de Francia: Mision Interministerial de la Informacion Cientifica y Tecnica; Centro Frances de Informacion Tecnica e Industrial; Direccion General de Integracion y Analisis de la Informacion. 6 al 9 de septiembre, 1982. 61 p.

21.- Creacion de bases de datos. SECOBI-CONACYT; Centro Cientifico Frances de la Embajada de Francia; 1 al 15 de julio, 1985.

22.- Amiel, Philippe. "Le Informatique au bout de la lange" Science S et Avenir. Numero especial. Hors. Serie. No. 53. pp. 78-81.

23.- "Quand le dictionaire s'automatise". Science S et Avenir. Numero especial. Hors. Serie. No. 53. pp. 94-100

24.- Atkinson, Steve D.; Watkins, Steven G. "Managing data base Information: A microcomputer application in a computer search service". Online. January, 1985. pp. 52-63

25.- Bartschi, Martin. "An Overview of Information retrieval subjects" pp. 67-84

26.- Bottom, Joseph; Berhard, Alan; Anderson, Kevin. "The art of modeling" Datamation pp. 140-146

27.- Commiot, Dominique; Ronai, Maurice. "Industrie du savoir, industrie de l'intelligence". Science S. et Avenir. Numero especial. Hors. Serie. No. 53

28.- Egyhazy, Esaba J. "Micro- computersw and relational database managment systems: a new strategy for decentralizing database" Database. Fall 1984. pp 15-20.

29.- Hopkinson, Alan "Developing the common communication format" Information Development 2(2) April, 1986

30.- Jarvis, John F. "Robotics" pp. 283-292

31.- Kanovi, Henry; Canegheni, Michele Van. "Prolog: un langage pour la gestion ?". Informatique et Gestion (149) Mars, 1984. pp. 64-71

32.- Mazion, Regine. "Banques d'images". Science S. et Avenir. Numero especial. Hors. Serie no. 53. pp 66-74.

33.- Molino, Enzo. "Bases de datos: consideraciones en los paises en desarrollo" pp. 259-266

34.- Nageswara rao, S.V.; Shitharama Jyenogar, S.; Veni Madhavau, C.E. "A comparative study of multiple attribute tree and inverted file structures for large bibliographic files" Information Processing and Management. 21(5), 1985. pp. 433-442.

35.- Nan, Dana S. "Expert computer systems" Computer. Feb. 1983. pp. 63-85

36.- Oh, Se-Young. "A walsh-hardware based distributed storage device for the associative search of information". IEEE: Transactions on pattern Analysis and Machine Intelligence. Vol Pami 6(5), sep., 1984. pp. 617-628.

37.- Poitevin, Jean Francios. "La traduction assistee par ordinateur" Science S. et Avenir. Numero especial. Hors. Serie no. 53. pp. 83-92.

38.- Pratt, G.E.C. "Using the micro-computer to simplify database access; designing interfaces to complex files" pp. 131-138.

39.- Roiller, Marc. "Du manuscript au compuscrit" Science S. et Avenir. Numero especial. Hors. Serie no. 53. pp. 38-43

40.- Ronai, Maurice. "Les trois visages des banques de donnes" Sciences S. et Avenir. Numero especial. Hors. Serie no 53. pp 22-29

41.- Ross, Douglas T. "Applications and extensions of SADT" pp. 25-[34]

42.- Zarri, Gian Piero. "Expert systems and information retrieval an experiment in the domain of bibliographical data management" Developments in Expert Systems. pp. 201-220

43.- Zenzo, Silvano di "Multiple Boolean Algebras and their applications to fuzzy sets". Information Sciences. 35, 1985. pp. 111-132.

44.- Borko, H. "Design of Information and Services". A. Rev. Inf. Sci. Technol., 1967, 2:35-61.

45.- Burns, R.W. Jr. "A generalized methodology for library systems analysis". College and Res. Libr., 1971, 32 (4): 295-303.

46.- Costello, J.C. "The charter: A must for effective information system planning and design". J. Chem. Docum. 1967, 4(1): 12-20.

47.- Griffiths, J.M. "Application of minicomputers and microcomputers to information handling". PGI 81/WS/28,

Paris, UNESCO, nov 1981.

48.- Johnson, R.A.; Kast, F.E.; Rosenweig, J.E. "Theorie, conception et gestion des systemes". Paris, DUNOD.

49.- Katter, R.V. "Design and evaluation of information systems". A. Rev. nf. Sci. Technol., 1969, 4: 31-70

50.- Kraft, D.H. "A decision theory view of the information retrieval situation: an operation research approach". J. Am. Soc. Inf. Sci., 1973, 24 (5): 368-375

51.- Liston, D.M.; Schoene M.L. "A systems approach to the design of information systems". J. Am. Soc. Inf. Sci., 1971, 22: 115-122

52.- Mitroff I.; Williams J.; Pathswohl E. "Dialectical inquiring systems: a new methodology for information science". J. Am. Soc. Inf. Sci., 1972 (nov.dec.): 365-378

53.- Scheffler F.L. "A novel philosophy for the design of information storage and retrieval systems appropiate for the 70". S.J. Am. Soc. Inf. Sci., 1973 (May-June): 205-209

54.- Senko M.E. "Information Systems: records, relations, sets, entities and things". Inf. Systems, 1975, 1: 3-13

55.- SIS.- MERISE - SD. "Methode de developpement de systemes d'information". dec 1980

56.- Taylor C.I. "The information center: considerations in planning". Proc. Am. Soc. Inf. Sci., 1971, 8 (7-11 nov.): 1376-1386

57.- Wasserman A.I. "Information system design methodology". J. Am. Soc. Inf. Sci., 1980 (January): 5-24

58.- Zimmerman P.J. "Principles of design for information systems". J. Am. Soc. Inf. Sci., 1977, 28 (Jul.): 183-191

59.- Amsterdam, J. "Programming Project: Data compression with Huffman Coding". BYTE vol. 11, Num. 5, May. 1986, p.99+

60.- Fogel, J.F. "El surgimiento de la inteligencia artificial". Contextos vol. 3, num. 50, abr.30, 1985, p. 42-47 (6p)

61.- Ballard, D.; Brown, C. "VISION: Technology is still being challenged to create reliable real-time vision systems". BYTE vol. 10, Num. 4, Abr. 1985, 245+ (12p)

62.- Stevens, J. "Revers engineering the brain" BYTE vol. 10, Num. 4, Abr, 1985, 287+ (8p)

63.- Michaelsen, R.; Michie D.; Boulanger A. "The technology of expert systems". BYTE vol. 10, Num. 4, -Abr, 1985, 303+

64.- Minsky, M. "Por que la gente cree que las computadoras no pueden pensar ? ". Contextos vol. 3, Num. 50, Abr. 30, 1985, 48-55

65.- Curtice, R. "Getting the database right". Datamation, vol. 32, num. 19, oct. 1986, p.99+ (3p)

66.- DeMaria, R. "REFLEX: An analytical database loaded with unique features". BYTE vol. 11, Num. 8, Ago. 1986, 277-278

67.- Thompson, D.; Campbell M.; "An introduction to MINISIS", Manual de MINISIS, IDRC, Ottawa, Canada, January 1984, 1-27

68.- Lancaster, F. "Information Retrieval Systems: Characteristics, Testing and Evaluation", John Wiley & Sons, 1979

69.- Batten, W.E.; Ashword W. "Handbook of Special Librarianshipand Information Work", Aslib 1975

70.- Shapiro, E. "Text Databases", BYTE, Vol.9, Num. 11, oct. 1984, 147-150

71.- Krajewski R. "Database types", BYTE, Vol. 9, Num. 11, oct. 1984, 135+ (4p)

72.- "Programmable relational databases", PC Magazine, Vol. 5, Num. 12, Jun. 1986, 125-126

73.- Krasnoff, B.; Dickinson J. "Project Database II", PC Magazine, Vol., 5 Num. 12, jun. 1986, 106+

74.- Atkins, R; Mazur, W. "The Dayflo Architecture", BYTE, Vol. 9, Num. 11, oct 1984, 155+

NOTAS