Algoritmo de compresión para el almacenamiento de información bibliográfica

(Primera de dos partes)

JUAN VOUTSSÁS MÁRQUEZ
Centro Universitario de Investigaciones Bibliotecológicas
de la UNAM, 04510, México D.F., Tel: 56-23-03-29
E-Mail: voutssas@servidor.unam.mx

MIGUEL AGUSTÍN RUIZ VELASCO Y ROMO
Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas
de la UNAM, 04510, México D.F., Tel: 56-22-36-76
E-Mail: mrvyr@servidor.unam.mx

Artículo

RESUMEN
Los bancos de datos bibliográficos desarrollados en computadores de tipo personal se han visto limitados en cuanto al número de fichas y por su rendimiento en función del tamaño de la plataforma en donde se desarrollan e instalan. Los manejadores comerciales de bases de datos para estos equipos han sido construidos de acuerdo con necesidades de tipo general en el mercado y no contemplan las características propias de la información bibliográfica, por lo que decrece su rendimiento rápidamente en función al tamaño del banco de datos. En esta primera parte del documento se analiza esa problemática y las características propias de la información bibliográfica en lo tocante a su inclusión en bancos de datos electrónicos, y se presenta un modelo de compresión de datos bibliográficos en forma de algoritmo que permite entre un 40 y 70% de compresión sin menoscabar las características propias de la información bibliográfica. En la segunda parte del documento se presentan las técnicas para crear y comprimir índices preconstruidos de recuperación y archivos de recuperación por palabras en búsqueda libre, así como las técnicas para accederlos y serle presentados al usuario final. En esa parte se concluye que bancos de datos de cientos de miles de fichas y millones de palabras de recuperación pueden comprimirse en el espacio de un *cd-rom (650 Megabytes)*, y aun extrapolarse estos valores a cotas mucho mayores.

Palabras Clave: Bases de Datos, Automatización de Índices, Algoritmos.

COMPRESSION ALGORITHMS FOR THE STORAGE OF BIBLIOGRAPHIC INFORMATION
(First of two parts)

JUAN VOUTSSÁS-MÁRQUEZ
MIGUEL AGUSTÍN RUIZ-VELASCO Y ROMO

ABSTRACT
Bibliographical data bases which have been developed in PC computers have been limited regarding total number of fiches and their performance due to the size of the platform in which those data bases are developed and installed. Commercial PC data base management softwares have been constructed with a general approach, thinking in standard applications, and do not consider the particular features of the bibliographic information. Thus, they decrease in performance exponentially in relation with the size of the data base. In this first part of the survey, the problems are discussed, as well as those typical features of the bibliographic information regarding to its inclusion in computerized data bases. A bibliographic data compression model is introduced as an algorithm, allowing between 40 and 70 % of compression rate without loosing information quality. In the second document, procedures for creation and compression of preconstructed indexes will be presented, as well as retrieval files for word free-searching. Some techniques for creation and retrieval of both access paths will be fully discussed. In that part the final conclusion shows that data bases with several hundreds of thousands of records owning several millions of retrieval words can be compressed to the available space of a cd-rom (650 Mb), and even expanded to greater figures.
Key Words: Database, Index Automation, Algorithms.

INTRODUCCIÓN

Desde el advenimiento de la primera computadora personal a principios de los años ochenta, el deseo de crear, explotar y distribuir bases bibliográficas electrónicas de forma local surgió en todas las bibliotecas que las adquirían, pero al intentar la creación de estas bases el personal técnico responsable de ello enfrentó problemas antiguos sobre las proporciones de la plataforma de cómputo de la que se disponía. No todos los potenciales creadores de bancos de datos electrónicos pueden tener acceso a computadores de plataformas poderosas dentro del mercado de cómputo. Muchas veces se tienen bancos de datos susceptibles de ser creados pero el equipo disponible no parece tener el suficiente poder para contender aceptablemente con las dimensiones de tales bancos. Otras veces cuando se piensa en comprar pareciera que el equipo precisara capacidades y costos que superan el alcance de las bibliotecas. En otros casos, el banco de datos resultante ocupa tantos recursos que queda fuera del alcance de una computadora promedio para el gran público potencialmente usuario, limita su distribución y parece quedar restringido a ser usado en plataformas de alta capacidad, y por lo tanto precio, dentro de instituciones de cierta capacidad económica.

Durante la década de los noventa, el manejo de bancos de información bibliográfica automatizada observaba una relación muy directa entre el límite del tamaño del banco de datos y la plataforma de cómputo en la que éste se encontraba instalado. El patrón, por lo tanto, cumplía fielmente las siguientes implicaciones**:** *una plataforma pequeña limita a que en ella por fuerza sólo puedan montarse bancos de datos pequeños; la instalación de un banco de datos grande implica forzosamente el uso de una plataforma grande.* Dicho de otra forma: el tamaño de un banco de datos es directamente proporcional al tamaño de la plataforma en la que se instala.

Por lo tanto es necesario definir qué entendemos por banco pequeño o grande y plataforma pequeña o grande. Para fines de manejo de información bibliográfica en el medio bibliotecario mexicano, de manera *totalmente convencional* y únicamente para el ámbito de este trabajo, definimos entonces por plataforma 'pequeña' a aquella que está constituida por una computadora personal (PC) con un solo procesador de tipo *Intel*, que opera bajo el sistema operativo DOS o Windows, y tiene una velocidad *pequeña*. Como la velocidad de los procesadores ha estado cambiando muy rápidamente a lo largo de este proyecto, fijar un número de Megahertzios como "pequeño" o "grande" resulta muy inestable y es una cifra que se hace obsoleta rápidamente. Definamos entonces como velocidad "grande" a la máxima velocidad del procesador que esté a la venta en un momento dado, y "pequeña" a la velocidad menor o igual a la mitad de ese máximo. Es decir, si la máxima velocidad de procesador que está a la venta al momento de leerse este documento es, digamos 1000 Megahertzios, entonces de 500 a 1000 Megahertzios será considerada como una velocidad "grande"; y por debajo de 500 Megahertzios será considerada una velocidad "pequeña". Extrapólense estos números al momento de la lectura. Los principios enumerados aquí se han comportado linealmente durante varios años. Definiremos también como plataformas "grandes" a aquellas de tipo *risc* que corren bajo sistemas operativos *Unix* o semejantes. Y un banco de datos *pequeño* será aquel que se encuentra por debajo de las 150,000 fichas o registros.

Estas son definiciones arbitrarias y las usaremos sólo para el desarrollo de este proyecto pues es muy difícil establecer una línea precisa sobre dónde acaba una cosa y dónde empieza la otra, y en todo caso siempre podrían definirse también toda una variedad de entes 'medianos'. Para simplificar las consideraciones de este trabajo hacemos esta división que parece simple y efectiva.

Existen o han existido en el mercado manejadores genéricos de bases de datos para plataformas pequeñas, tales como dBASE, Foxbase, Clipper, etcétera que han sido utilizados en la inmensa mayoría de bancos construidos al efecto.

Por otro lado existen también manejadores especializados de información bibliográfica en plataforma PC, basados en los manejadores genéricos de bases de datos como los enunciados anteriormente, los cuales tienen la característica de que a medida que se incrementa el banco de información decrece la capacidad de actualización y consulta, y en general también su rendimiento.

Por último existen motores de bases de datos para plataformas grandes, tales como Oracle, Sybase, Informix, etcétera, que generalmente no están disponibles en plataforma PC, dado que normalmente son instalados en sistemas servidores y tienen un costo muy alto relacionado con el tipo de plataforma y el número de licencias.

La pregunta es ¿puede romperse el esquema anterior y por ende llevarse las plataformas pequeñas más allá de su rendimiento típico hacia cotas más parecidas a las de plataformas grandes en el manejo de información bibliográfica ?

Para poder llevar una plataforma pequeña hacia rendimientos mayores pueden hacérsele ciertas modificaciones tales como incrementar la velocidad del procesador a la máxima que haya en el mercado; instalar más de un procesador en la computadora; cambiar el sistema operativo por un Linux; agregarle considerables cantidades de memoria *ram*; instalarle discos de gran capacidad de almacenamiento, etcétera. Pero entonces ya no será una plataforma típica; es decir, la que podemos encontrar en la generalidad de las bibliotecas mexicanas. Tal solución, por decirlo de alguna forma, sería un tanto 'artificial' y fuera de lo estándar.

Existe una solución que puede considerarse como más eficiente y digamos, más 'natural', y que consiste en desarrollar algoritmos especializados para el almacenamiento y recuperación de la información bibliográfica, que integrados a programas de computadora puedan ser utilizados en plataformas pequeñas consideradas 'típicas' o promedio, y que elevan el rendimiento a cotas muy cercanas al rango de las plataformas que pueden ya ser consideradas como grandes. Decimos algoritmos especializados en información bibliográfica porque este tipo de información tiene características que la hacen muy distinta del manejo de información típica o comercial para la cual están diseñados los manejadores de marca.

Para la información de tipo bibliográfica o catalográfica, puede aceptarse que los procesos de actualización del acervo sean relativamente lentos (del rango de horas o días) a medida que se incrementa la información, pero el proceso de consulta deberá ser siempre lo más rápido posible (del rango de segundos). Esto solamente se logra con programas basados en algoritmos que enfoquen correctamente el rendimiento que se requiere para optimizar el servicio que obtiene el usuario que los consulta.

OBJETIVO

El objetivo del proyecto consiste entonces en diseñar y construir algoritmos especializados que optimicen el comportamiento de los procesos de almacenamiento y recuperación bibliográfica, y que puedan ser integrados a programas de computadora capaces de ser ejecutados en lo que hemos definido como plataformas pequeñas pero que manejan volúmenes de información muy superiores, las cotas típicas de este tipo de plataformas. Se aprovechará en lo posible las facilidades que dan los manejadores "típicos" de estas aplicaciones dBase para facilitarles a quienes programen en ellos la inserción de los algoritmos.

La metodología que se utilizó para este proyecto es la siguiente:

- Definición de las características propias de la naturaleza de este tipo de información.
- Análisis de las características propias de los paquetes manejadores de bases de datos típicos para estas plataformas.
- Diagnóstico de la problemática que presentan los paquetes comerciales contra la información bibliográfica.
- Diseño de los algoritmos de optimización correspondientes.
- Evaluación del comportamiento de los algoritmos y determinación de los resultados.
- Conclusiones

CARACTERÍSTICAS DE LA INFORMACIÓN BIBLIOGRÁFICA

Como ya se ha mencionado, un buen número de los problemas del manejo automatizado de información bibliográfica surge de sus características tan particulares. El tipo de información que se maneja, como en toda base de datos, puede modelarse utilizando varias técnicas, entre las que seleccionamos la de "relación de entidades", pues sin ser la más moderna es la que de forma más sencilla permite representar las características propias de los atributos de la información y sus métodos de organización (Keuffel, 1996). De inicio, podemos definir este tipo de estructuras bajo el esquema básico: Banco - Archivo - Registro - Campo - Subcampo; donde:

- **Banco** contiene la totalidad de los archivos de información del sistema. que se añadió.
- **Archivo**: contiene un conjunto de registros.
- **Registro**: contiene una unidad de información (ficha) mediante un conjunto de campos.
- **Campo**: contiene la información de cada elemento de la ficha, la cual a su vez puede o no contener subcampos.
- **Subcampo**: contiene parcialmente la información del campo.

La información bibliohemerográfica tiene propiedades particulares que la hacen un modelo "*sui géneris*" en lo que toca a las formas de normalización de datos y a su almacenamiento en un computador. Si bien las características de este tipo de información son muy numerosas, los elementos principales de ella que inciden fuertemente en cualquier diseño para procesamiento electrónico son los siguientes:

- Campos de longitud variable. Por ejemplo, la longitud de un autor que es totalmente variable en longitud, sobre todo en autores corporativos o reuniones. Otros ejemplos son los títulos o las notas, cuya longitud es muy variable.

- Rompimiento de un campo dado en subcampos, como por ejemplo el pie de imprenta.
- Existencia de algún campo de 0 a **n** veces. Por ejemplo, los encabezamientos de materia.
- Búsqueda por índices múltiples. Muchos de los campos de una ficha se requieren como llaves de recuperación: número de clasificación, autores, título, ISBN, pie de imprenta, encabezamientos de materia, serie, etcétera. Estos a su vez pueden agruparse entre diversos campos para formar un solo índice: autores con coautores, encabezamientos con otros temas, etcétera.
- Búsqueda por palabras. En la recuperación bibliográfica no sólo se desea recuperar por el texto completo de un autor, el título, el tema, el pie de imprenta, etcétera. También se desea recuperar por conjuntos de palabras, una palabra y hasta trozos de palabra incorporados dentro de ellas.

Estas son las características básicas de la información bibliográfica, pero ¿cuáles son las características propias de los paquetes manejadores genéricos a estas plataformas con respecto a las necesidades que presenta dicha información?

Longitud fija

Comencemos con la longitud fija: en la inmensa mayoría de los manejadores de bases de datos relacionales para ir almacenando cada ente de un registro (cada campo de una ficha) se debe definir por lo general un campo alfanumérico con una *longitud fija* de **n** caracteres. En información bibliográfica ello genera problemas a la hora de almacenar en un archivo de computador dado que la longitud generalmente no será la adecuada, ya que:

- Si la longitud del dato a guardar es menor a la longitud definida del campo, se generará "poro", es decir, caracteres blancos ociosos.
- Si la longitud del dato es mayor a la longitud del campo, habrá truncamiento del dato y por ende pérdida de información.
- La existencia de información para ese dato podrá ser de 0 a 1 veces solamente; es decir, el dato puede existir sólo una vez, o ninguna.
- Cuando la existencia es 0, es decir, no existe información para ese dato, se producirá desperdicio total de caracteres, también llamado poro total.

Algunos manejadores de este tipo de información, como por ejemplo "Microisis", resuelven parcialmente la posibilidad de existencia múltiple de campos, insertando un caracter predefinido en el texto, el cual indica que se trata de un siguiente campo repetitivo, pero continuarán las otras limitaciones en lo que respecta a tener definido un campo fijo.

Longitud Variable

En algunos tipos de manejadores se puede definir un campo alfabético de longitud variable, pero por lo general se restringe a una ocurrencia por campo; esto no resuelve el problema de la ocurrencia del campo, el cual sigue existiendo sólo de 0 a 1 veces.

En el caso anterior se resuelven algunos problemas como el poro y la longitud, pero no queda resuelta la ocurrencia múltiple. Esto también es resuelto por algunos manejadores insertando un caracter predefinido que identifique el inicio del siguiente campo.

Algunos manejadores presentaron el problema en forma total; otros resuelven la longitud variable pero no la ocurrencia múltiple; otros más, los subcampos pero no la longitud fija, etcétera. Casi siempre se encontraron soluciones a alguno o a varios de estos problemas en un cierto manejador, pero no encontramos ninguno en el mercado que resolviera *simultáneamente todas las características enunciadas de la información bibliográfica*.

Además del hecho de que las consideraciones anteriores han sido tomadas en cuenta en el algoritmo, para poder tener las facilidades de transferencia interbibliotecaria de información a nivel internacional, se debe respetar el tipo de codificación propuesta por la Biblioteca del Congreso de los Estados Unidos (formato *marc*) y algunos estándares internacionales como el ISO-2709.

ESTIMACIÓN DE VOLÚMENES DE INFORMACIÓN EN BASES DE DATOS

Como hemos descrito, normalmente las bases de datos tienen campos de longitud fija (antes de comprimir) que al unirse forman registros. Usemos un archivo de fichas de tesis simplificado como ejemplo para poder visualizar este concepto de forma más gráfica. Su diagrama conceptual sería:

Identif.	Autor	Título	Institución	Escuela	Carrera	Nivel	Páginas	Asesor

El diagrama anterior representa un archivo de fichas de tesis donde cada renglón es un 'registro', es decir, los datos completos de la ficha de una tesis. Cada registro a su vez contiene elementos, representados como columnas, llamados 'campos'. A un archivo formado de esta manera, se le llama un archivo 'plano'. A un archivo de este

tipo en un manejador convencional es necesario asignarle una longitud a cada campo para poder escribirle datos. Asignemos longitudes en el ejemplo:

Campo	Longitud
Identif.	10
Autor	250
Título	200
Institución	60
Escuela	40
Carrera	30
Nivel	10
Páginas	10
Asesor	40
Total	650

De acuerdo con la definición anterior, la cual es típica de un manejador de bases de datos de uso general para PC, la longitud de cada registro (renglón) sería de 650 caracteres o *'bytes'*, ya que este número es igual a la suma de todas las longitudes de los campos del registro. La cuantificación del espacio de disco que ocupa este archivo plano es igual al producto del número de registros que tenga ese archivo, multiplicado por la longitud en caracteres de cada registro; es decir, el ancho por la profundidad. Supongamos un archivo donde tenemos 183,000 registros, es decir fichas; tomando entonces en cuenta 650 caracteres por cada registro, el producto sería:

183,000 x 650 = 118'950,000 caracteres de computadora, o sea *bytes*. Es decir, necesitamos 119 Megabytes aproximadamente de espacio disponible en nuestro disco del computador para alojar este archivo plano. Nótese que al guardar la información en un esquema de este tipo, al registrar un título cuya longitud mida, digamos 50 caracteres, al haber definido 200 para este campo dejamos 150 sin utilizar, los cuales de todos modos son incluidos en el registro y quedan como 'poro'. Por supuesto puede reducirse la definición de la longitud del campo, pero entonces corremos el riesgo de que un buen número de títulos queden truncados; todos los que excedan esa longitud de campo.

Establezcamos otro ejemplo de una pequeña ficha de tesis cuyo registro tiene 4 campos con las siguientes longitudes.

Campo	Longitud
Autor	25
Título	30
Carrera	16
Asesor	24

Longitud del registro: 95 caracteres. Llenemos un par de registros con datos; su imagen sería la siguiente:

Autor	Título	Carrera	Asesor
Castillo Montes, Carmen G	Servicios hemerográficos en la	Ingeniería en Co	González Velázquez, Erne
Torres Suárez, Alejandro	Performance de servicios	Licenciado en In	

Hagamos ahora una segunda definición de longitudes mayores para el mismo registro:

Campo	Longitud
Autor	40
Título	200
Carrera	20
Asesor	40

Longitud total de cada registro: 300 caracteres. Si llenáramos un par de registros con datos, su imagen sería la siguiente:

autor	Título	carrera	Asesor
Castillo Montes, Carmen Guadalupe del	Servicios hemerográficos en la biblioteca de DGSCA	Ingeniería en Computación	González Velázquez, Ernesto
Torres Suárez, Alejandro	Performance de servicios	Licenciado en Informática	

Donde cada registro ocupa 300 caracteres, independientemente de los caracteres escritos. Como puede observarse en estos ejemplos, la primera definición es bastante más económica en caracteres (95) por registro, pero la información queda truncada. En la segunda la información no tiene ese problema, pero la cantidad de espacio desperdiciado es enorme, y ese espacio crece en la misma proporción que el archivo. Sigue existiendo el problema de que podría haber campos tan largos que no entren completos.

Sería pues conveniente que en el momento de almacenar la información bibliográfica, se contara con una estructura que resuelva la problemática enunciada anteriormente y que aparezca por lo general en los manejadores de bases de datos convencionales a esta plataforma.

Así el objetivo del algoritmo para la preservación integral de la información bibliográfica al momento de almacenar deberá cumplir las cuatro siguientes premisas:

- Que *toda* la información de cualquier campo de toda ficha pueda ser almacenada sin truncamientos o abreviaturas forzadas.
- Que *no se genere poro* en la base de datos por la inclusión de campos fijos muy grandes; es decir, que se maximice el aprovechamiento del espacio.
- Que la información almacenada conserve todas las características inherentes a la información bibliográfica: campos repetitivos o nulos, longitud variable de los campos; fichas de cualquier longitud para describir adecuadamente lo que se desea; posibilidad de abrir n subcampos dentro de un campo; compatibilidad con estándares convencionales (*marc*, ISO 2709, etcétera).
- Que la ficha almacenada respete toda su tipografía original (mayúsculas y minúsculas), diacríticos, signos de puntuación, etcétera.

CARACTERÍSTICAS DEL ALGORITMO DE COMPRESIÓN DE DATOS

El algoritmo que se utilice para la compresión de datos deberá tomar en cuenta entonces que no deberá dejar porosidad en los campos, y por otro lado que la información no sea truncada por falta de capacidad del campo, lo cual generaría información incompleta. Debe preservarse de esta forma el 100% de la información deseada.

El algoritmo de compresión de datos que ideamos proviene del concepto propuesto por *marc* para 'etiqueta', concepto que implica que para representar la información bibliográfica se requiere asociar cada campo de una ficha con una etiqueta, la cual normalmente viene del criterio de etiquetas numéricas definido en el formato *marc*, o de un criterio arbitrario definido por el usuario.

El principio del algoritmo de compresión se basa en el hecho de que al asociar la etiqueta con la información del campo pueden ponerse uno a continuación del otro sin dejar espacios en blanco al final de cada campo, tal como lo establece *marc*. Para efectos prácticos dicha información está escrita en "prosa", sin ningún tipo de separador como normalmente se usa en los archivos de texto, tal como CrLf (control de línea nueva). Hasta aquí el formato *marc* aparece tal cual puede ser observado en una cinta de este tipo. La innovación en este algoritmo consiste en el hecho de transportar esta imagen de registro físicamente a un archivo de computadora, haciendo que esa imagen de registro *marc* compactado sustituya al archivo típico plano de longitud fija que hemos descrito previamente.

Ilustremos con un ejemplo lo anterior: Definamos ahora una estructura que ya no sería archivo plano. Definamos primero la codificación por campos asociados a una etiqueta (los números de etiqueta son arbitrarios y van precedidos del signo "$" para evitar que sean confundidos con el texto mismo):

Campo	Descripción
$100	autor
$245	título
$300	carrera
$700	asesor

No definamos longitudes de campo; la longitud bajo este nueva estructura es, por lo tanto, 'la que sea necesaria'. Al ir capturando datos en este archivo la información de este ejemplo, expresada en líneas separadas, quedaría en un archivo de captura *temporal*, como duplos (campo-información):

$100	Castillo Montes, Carmen Guadalupe del
$245	Servicios hemerográficos en la biblioteca de DGSCA
$300	Ingeniería en Computación
$700	González Velázquez, Ernesto
FIN	
$100	Torres Suárez, Alejandro
$245	Performance de servicios
$300	Licenciado en Informática
FIN	

El algoritmo final consiste en juntar cada duplo de campo uno a continuación del otro hasta completar una ficha, y a su vez una ficha a continuación de la otra, formando una cadena de caracteres que contiene a *todo* el archivo. La información quedará compactada como duplos ($campo, información), y al final de cada ficha para separarla de la que sigue existirá un caracter de fin de registro (en este ejemplo utilizaremos el caracter @) como separador entre fichas). El archivo resultante tendrá *un sólo registro* formado por esta cadena de caracteres que comprende a todas las fichas. Este archivo no es ya un archivo plano, lo definimos como un archivo 'binario' y recalcamos, con un sólo registro. Su imagen sería:

$100 Castillo Montes, Carmen Guadalupe del $245 Servicios hemerográficos en la biblioteca de DGSCA $300 Ingeniería en Computación $700 González Velázquez, Ernesto @ $100 Torres Suárez, Alejandro$245 Performance de servicios$300 Licenciado en Informática @

El archivo en el recuadro debe leerse como *un sólo renglón continuo*. En esta página es imposible representarlo en esa forma, pero tenga en cuenta el lector que no son cuatro registros o renglones, sino un sólo registro continuo que puede almacenarse así en un disco de un computador.

La imagen anterior de este par de registros es la que estamos acostumbrados a ver típicamente en un registro al estilo *marc* para exportación de datos bibliográficos. Esta imagen no tiene poro ni truncamiento. La pregunta es ¿puede este tipo de registros sustituir a los registros planos de longitud fija que hemos definido previamente? Si esto fuese posible, la compactación de datos sería enorme y eliminaría todo el poro y desperdicio, además de que no existiría el problema de truncamiento de textos en los campos.

Si visualizamos la información anterior encerrada en el recuadro como *un sólo registro* físico podemos observar que los dos registros de datos ocupan solamente 244 caracteres, a diferencia del ejemplo anterior donde los dos registros ocupan 600 caracteres. Si extrapolamos los ejemplos descritos a continuación, podríamos construir un archivo que tuviera un sólo registro físico, es decir un sólo renglón pero miles y miles de registros lógicos (fichas) pegados uno tras el otro en una cadena sin fin que tendría la siguiente imagen:

registro 1@registro 2@registro 3@registro 4@……registro n-2@registro n-1@registro n@

Donde *n* es un número enorme de fichas limitado sólo por la capacidad de almacenamiento del disco. En este caso, como puede verse, no hay truncamiento de la información debido a longitudes fijas ni tampoco hay poro o desperdicio. Cada campo toma exactamente lo que necesita y es recuperable gracias a la etiqueta. Al multiplicar este ahorro por decenas de miles o cientos de miles de fichas la diferencia de uso del disco es muy sustancial. Este es el concepto de la codificación que establece *marc* y lleva físicamente a un archivo de disco en un computador. El concepto de este procedimiento, como puede verse, es simple pero efectivo.

Una vez establecido el registro básico que nos resuelve estos problemas, podemos hacer más complejo el registro para ir resolviendo algunas otras necesidades ya estipuladas, tales como: a) el rompimiento de un campo en subcampos y b) las ocurrencias múltiples de un campo.

Codificación por campos y subcampos

Como lo hemos establecido en algunos casos es necesario subdividir el campo a su vez en subcampos, lo que permite distinguir elementos menores dentro de cada campo. Este registro es conceptualmente similar al anterior, a excepción de que se trata de un tipo de registro que cuenta con campos y subcampos. Los anteriores generalmente no existen en muchos tipos de bases de datos. La forma de construir un registro es generando un *tuplo* (Campo, marcas, duplo (&subcampo, información)). Las marcas son referidas a 2 caracteres exactos que son utilizados en la codificación tipo *marc*, los cuales tienen sentido en función del tipo de registro que se esté utilizando.

Campo	Subcampo	Descripción
$100	&a	autor
$245	&a	título
$300	&a	carrera
$700	&a	asesor

La información de este ejemplo, expresada en líneas separadas quedará como :

$100	bb	&a	Castillo Montes, Carmen Guadalupe del
$245	bb	&a	Servicios hemerográficos en la biblioteca de DGSCA
$300	bb	&a	Ingeniería en Computación
$700	bb	&a	González Velázquez, Ernesto
FIN			
$100	bb	&a	Torres Suárez, Alejandro
$245	bb	&a	Performance de servicios
$300	bb	&a	Licenciado en Informática
FIN			

La información quedará compactada como tuplos ($campo, marcas, duplos (&Subcampo, información)), y al final existirá un caracter de fin de registro:

$100 bb &a Castillo Montes, Carmen Guadalupe del $245 bb &a Servicios hemerográfi cos en la biblioteca de DGSCA $300 bb &a Ingeniería en Computación$700 bb &aGon zález Velázquez, Ernesto@ $100 bb &a Torres Suárez, Alejandro$245 bb &aPerforma nce de servicios $300 bb &a Licenciado en Informática @

Ocurrencias múltiples de un campo

Ejemplos de información con ocurrencia de campos de 0 a **n** veces:
Codificación por campos:

Campo	Descripción
$100	Autor
$245	título
$300	Institución carrera
$700	asesor

La información de este ejemplo, expresada en líneas separadas quedará como:

$100	Castillo Montes, Carmen Guadalupe del
$100	Taboada Garcés, Sergio
$100	Ordaz Solís, Susana
$245	Servicios hemerográficos en la biblioteca de DGSCA
$300	UNAM, Ingeniería en Computación
$700	González Velázquez, Ernesto
$700	Peña Torres, Marcela
FIN	
$100	Torres Suárez, Alejandro
$100	Ochoa Hernández, Mario
$245	Performance de servicios
$300	UNAM, Licenciado en Informática
FIN	

La información quedará compactada como duplos ($campo, descripción), y al final existirá un caracter de final de registro:

$100 Castillo Montes, Carmen Guadalupe del $100 Taboada Garcés, Sergio $100 Ordaz Solís, Susana $245 Servicios hemerográficos en la biblioteca de DGSCA $300 UNAM, In

geniería en Computación $700 González Velázquez, Ernesto $700 PeñaTorres, Marcela @ $100 Torres Suárez, Alejandro $100 Ochoa Hernández, Mario $245 Performance de servicios $300 UNAM, Licenciado en Informática @

Vista de la información con codificación por campos y subcampos tipo *marc*:

Campo	Subcampo	Descripción
$100	&a	autor
$245	&a	título
$300	&a &b	Institución carrera
$700	&a	asesor

La información de este ejemplo, expresada en líneas separadas quedará como:

$100	bb	&a	Castillo Montes, Carmen Guadalupe del
$100	bb	&a	Taboada Garcés, Sergio
$100	bb	&a	Ordaz Solís, Susana
$245	bb	&a	Servicios hemerográficos en la biblioteca de DGSCA
$300	bb	&a &b	UNAM Ingeniería en Computación
$700	bb	&a	González Velázquez, Ernesto
$700	bb	&a	Peña Torres, Marcela
FIN			
$100	bb	&a	Torres Suárez, Alejandro
$100	bb	&a	Ochoa Hernández, Mario
$245	bb	&a	Performance de servicios
$300	bb	&a &b	UNAM Licenciado en Informática
FIN			

La información quedará compactada como tuplos ($campo, marcas, duplos (&Subcampo, descripción)), y al final existirá un caracter de fin de registro:

$100 bb &a Castillo Montes, Carmen Guadalupe del $100 bb &a Taboada Garcés, Ser
gio $100 bb &a Ordaz Solís, Susana $245 bb &a Servicios hemerográficos en la bibliot
eca de DGSCA$300 bb &a UNAM &b Ingeniería en Computación$700 bb &a González Ve-
lázquez, Ernesto $700 bb &a Peña Torres, Marcela @ $100 bb &a Torres Suárez, Al
ejandro $100 bb &a Ochoa Hernández, Mario $245 bb &a Performance de servicios
$300 bb &a UNAM &b Licenciado en Informática @

El lector debe imaginarse el registro anterior ilustrado en el recuadro no como seis renglones escritos uno abajo del otro, sino como *un sólo* renglón que ocupa desde el primero hasta el último caracter. Por razones del texto esto no puede ser representado aquí, pero es su concepción real. Si uno observa estos dos registros lógicos escritos en un sólo registro físico observará que poseen una serie de ventajas: no consumen espacio de más, ni dejan poro, como en registros de longitud fija; pueden subdividirse los campos en subcampos, con objeto de procesar solamente partes de ese campo; permiten la ocurrencia de 0 a n veces de un campo, sin mayor problema. Además, pueden construirse físicamente en un manejador 'típico' de PC con el concepto de 'archivo binario' (*binary file*) y un pequeño archivo índice que puede construirse con un archivo plano tipo .dbf con un esquema de "*listas*" muy simple:

Número de ficha	*Número del caracter de inicio*	*longitud de la ficha*
00001	1	262
00002	263	140
0000n	nnn	xxx

Standish (1980) presenta en su capítulo 5.3 (*List representation techniques)* una reseña muy completa de varias técnicas para manejo de listas. Entre ellas hemos seleccionado la que aquí se presenta. Ésta técnica en particular funciona así: el primer campo nos informa de cuál número de ficha se trata; el segundo número nos indica en cuál carácter del archivo de un sólo registro comienza nuestra ficha, y el tercer número nos dice cuál es la longitud de esa ficha. Así, dado un número cualquiera de ficha, como todo está en un orden creciente, la ubicamos rápidamente por el número de registro. Leemos los tres números enunciados y gracias a ellos podemos extraer el contenido exacto de una ficha cualquiera. Standish (1994) en esta obra, además de hacer una recopilación más exhaustiva de los tipos diferentes de técnicas de representación de "*listas*" que pueden utilizarse en forma alternativa para este tipo de almacenamiento de datos, presenta diversas técnicas para asociarlas a programas de computadora.

Al estudiar las variantes que existen en los archivos de manejadores típicos de plataformas pequeñas hallamos varias alternativas de solución y entre ellas se encontró una que reviste particular interés. Se trata de una estructura llamada 'archivo binario' (binary file) que consiste en información que puede registrarse en un archivo en 'tandas' o unidades de longitud variable. Llevado a su límite, TODO el conjunto de fichas, no importando su longitud, puede escribirse en *un sólo* registro físico; es decir, el archivo contiene un único registro formado por todas las fichas del archivo escritas en una sola cadena que tiene la estructura presentada anteriormente. Por el tipo de acceso para lectura y escritura de ciertos sistemas operativos propios de los procesadores *intel* típicos, se encontró que el límite máximo de longitud para cada ficha es de 64 kilobytes - 1; esto es, 65,535 caracteres por ficha, cantidad que sobrepasa con mucho lo que mide una ficha en la práctica. Si la escribiéramos en cartón sería algo así como una ficha con 200 tarjetas de continuación.

Aplicando los rendimientos de este algoritmo a bases de datos ya construidas, se tuvieron los siguientes comportamientos:

En el capítulo "Estimación de volúmenes de información en bases de datos" de este trabajo hicimos una estimación de aproximadamente 119 Megabytes para un archivo básico de tesis que contenía 180,393 fichas o registros de acuerdo con unos campos definidos. Al aplicar los procedimientos de compactación de datos del algoritmo propuesto realizados en el *cd-rom* Tesiunam obtuvimos que dichos archivos ocuparon finalmente 51.6 Megabytes, lo que implicaba en ese caso que teníamos un 57 % de ahorro sobre el número inicial de 119 Megabytes que se había calculado al definir esa base de datos sin los algoritmos de compresión. Al extrapolar las compresiones a otras bases de datos catalográficas (Librunam, Seriunam, Colmex, etcétera) se obtuvieron diferentes niveles de ahorro teórico del espacio ocupado por los bancos

definidos sin compresión y con compresión. Los niveles de reducción de espacio estuvieron entre el 40 y el 70 % después de utilizar el algoritmo de compresión. Para las pruebas se construyeron las bases de datos utilizando los archivos binarios ya mencionados. (Universidad Nacional Autónoma de México, 1992).

Cabe resaltar en este punto que tales cantidades de ahorro corresponden a los archivos básicos de información; es decir, el archivo donde se encuentran las fichas originales. Toda base de datos bibliográfica tiene además otra serie de archivos que permiten tener 'llaves' de recuperación por autor, título, tema, etcétera, así como archivos para recuperación por palabras, etcétera, los cuales serán tratados posteriormente. El algoritmo de compresión establecido aquí y los números de rendimientos presentados sólo tienen que ver con el archivo original de fichas y no con los archivos auxiliares de recuperación por llaves y palabras. Aun así, como puede verse, el factor de compresión es sumamente considerable.

RESUMEN DEL ALGORITMO

Haciendo un resumen de todo lo anteriormente expuesto, el algoritmo, ya expresado, como tal, quedaría en la forma siguiente:

" Inicio:
  - Crear archivo que servirá como índice; caracter de inicio = 1
  - Crear archivo binario vacío.

" Lectura:
  - Leer una ficha del archivo original.

" Proceso:
  - Dividir la ficha en campos.
  - Asociar al principio de cada campo una etiqueta típica.
  - Dividir cada campo en subcampos, si los tuviera.
  - Asociar al principio de cada subcampo que hubiese, un separador (si sólo existe un subcampo, el paso se puede omitir).
  - Truncar todos los espacios en blanco sobrantes al principio y al final de los campos y subcampos.
  - Pegar todos los campos de la ficha en una sola cadena y agregar un caracter de fin de ficha al final.
  - Obtener la longitud total de esa cadena que forma toda la ficha.
  - Escribir esa cadena al final del archivo binario. (Cadena anterior = Cadena anterior + cadena nueva)
  - Escribir en el archivo índice un registro que incluya el número de la ficha, el caracter de inicio y la longitud de la cadena (véase esquema archivo índice).
  - Actualizar: caracter de inicio = caracter de inicio + longitud de la cadena. Hay más fichas? : Ir a lectura.

" Cierre de archivos

" Fin.

CONCLUSIONES

Como puede observarse la compresión de datos es una técnica sumamente rentable y los ahorros de espacio que pueden obtenerse con ella son muy considerables. Además, su uso no implica la pérdida de las características deseables que les son propias a las fichas bibliográficas o catalográficas. Aun en la época actual, en la que parece haber la sensación de que los recursos de disco magnético y otros tipos de almacenamiento son ilimitados y baratos, la compresión puede significar la diferencia entre que un banco de datos quepa o no en algún dispositivo. Ha sucedido ya en la práctica que un banco dado, que se pretende editar en un disco compacto (*cd-rom*), exceda la capacidad de un disco antes del algoritmo de compresión, y sin embargo quepa después de ser aplicado éste. Ello ha significado la diferencia de que un banco se edite en un solo plato en vez de requerir dos. Al margen del costo que ello representa, la comodidad adicionada al usuario es sumamente considerable. Todo aquel que haya trabajado con un banco de datos que se presenta en dos discos que hay que estar intercambiando constantemente en la tornamesa *cd* sabe la diferencia que esto implica.

Resumiendo las ventajas que ofrece la utilización del algoritmo de comprensión podemos mencionar lo siguiente:

- Longitud total de cada ficha de 65,535 caracteres como máximo; esto es, prácticamente ilimitada.
- Almacenamiento íntegro de las fichas catalográficas.
- No hay truncamiento de las fichas.
- No hay poro.
- El modelo general de las fichas, evita tener que crear estructuras particulares propias para cada banco de datos.
- Campos de longitud variable. Mientras la suma de los campos no exceda el total de la longitud posible de la ficha, cada campo puede medir lo que se requiera.
- Existencia de cada campo de 0 a *n* veces.
- Compresión final entre el 40 y el 70 % del espacio original requerido.
- El programa de computador para compresión es rápido en su aplicación.

Como se ve las ventajas de utilizar el algoritmo de compresión de datos son muchas y no se requieren muchos recursos de máquina o de programación para llevar esto a cabo. Si bien su construcción no es trivial, tampoco representa un esfuerzo que sobrepase los estándares de un buen programador.

Esta técnica de compresión, trasladada a un programa de computadora, ha sido utilizada con éxito en más de veinte bancos en disco compacto o disco flexible, lo que ha permitido economizar muy sustancialmente el espacio originalmente consumido por las fichas y proporcionado más espacio libre para los mecanismos de almacenamiento y recuperación de cada banco de datos. Bancos tan grandes como Librunam 1992 (500,000 fichas), Tesiunam (180,000 fichas), Catálogo del Colegio de

México (300,000 fichas), Biblat (300,000 fichas), ocuparon en promedio menos del 40% de un disco compacto en relación con la información original de sus fichas, lo que implica que dejaron alrededor del 60% de espacio disponible para las "llaves" y mecanismos de recuperación. En su forma original, esto es, antes de las técnicas de compresión, esos archivos excedían la capacidad de almacenamiento de un cd-rom, por lo que era muy difícil editarlos sin ayuda de ellas en un sólo disco sin sacrificar parte de la versatilidad de las "llaves" de recuperación. Se sabe incluso del caso de un banco de datos, (Shock Waves in Medicine, *1995),* que se entregó en sólo dos discos flexibles de 1.44 Mb de capacidad cada uno; esto es, 2.88 Mb en total, banco que incluye casi 3,000 fichas, múltiples "llaves" de recuperación, y el programa de explotación y archivos de ayuda en tres idiomas. Esto sería imposible sin emplear estas exhaustivas técnicas de compresión de fichas.

El algoritmo ya programado es además muy eficiente en cuanto a los recursos utilizados. En una computadora PC con procesador 80486 de 66 Mhz y 8 Mb de *ram* el tiempo para comprimir y preparar 100,000 fichas apenas llevaba siete minutos. En un procesador Pentium de 150 Mhz y 16 Mb de *ram* el tiempo se abatía a menos de tres minutos, y en uno de 500 Mhz se reducía a menos de un minuto. Así, como puede observarse el comportamiento de plataformas *pequeñas* es muy aceptable aun si hablamos de bancos de un cuarto o de medio millón de fichas, tanto en lo relativo a los procesadores utilizados, como a memorias, discos duros, etcétera. Al principio de este trabajo en pruebas piloto, llegamos a crear y operar bancos con más de un millón de fichas en plataformas definidas como "pequeñas".

Lo más interesante del algoritmo de compresión es que a pesar de que sus primeros prototipos fueron diseñados cuando aparecían los primeros procesadores 80486 de 50 Mhz. y los discos duros solían tener capacidades de 400 megabytes, el algoritmo sigue siendo vigente. Las innovaciones en estos computadores han permitido seguir estirando el límite superior de la capacidad de procesamiento y almacenamiento de un computador. Como el algoritmo racionaliza los espacios en disco y aprovecha al máximo la velocidad del procesador sin pasos ociosos, cada generación nueva de ellos no lo hace obsoleto sino que extiende más y más su rendimiento. Los tamaños de procesador, memoria y disco duro actuales nos permiten suponer que podemos seguir utilizando la técnica de compresión pensando en bancos de datos no de algunos cientos de miles de fichas, sino de *algunos millones de fichas*, en una plataforma que sigue empleando computadoras de tipo *personal.* Si bien el motivo por el cual se ideó la compresión fue llevar las computadoras "pequeñas" a rendimientos mucho mayores, los cuales han sido rebasados ampliamente por las innovaciones tecnológicas, el principio sigue siendo válido y útil. Lo que cambia a cada día es la definición de plataforma y banco pequeños, pero los incrementos en rendimiento que llevan hacia cotas más propias de otros equipos de categoría y costo superior son perfectamente extrapolables hacia números cada día mayores.

LISTA DE BANCOS DE DATOS EDITADOS CON EL ALGORITMO

- ✓ Universidad Nacional Autónoma de México (1992*). ARIES: Acervo de Recursos de Instituciones de Educación Superior. 1992-1993.* 2ª. ed. Disco compacto y guía. México : UNAM, Dirección General de Intercambio Académico. Dirección General de Servicios de Cómputo para la Administración. [dos bancos, 20,000 registros]

- ✓ Universidad Nacional Autónoma de México (1992). *Catálogo de libros de las bibliotecas de la UNAM [LIBRUNAM].* Disco Compacto. México : UNAM, Dirección General de Bibliotecas; Dirección General de Servicios de Cómputo para la Administración. [un banco, 480, 588 registros]

- ✓ Universidad Nacional Autónoma de México (1992). *Tesiunam: Catálogo de tesis de la UNAM y otras instituciones*. Disco compacto y guía. México : UNAM, Dirección General de Bibliotecas, Dirección General de Servicios de Cómputo para la Administración. [un banco, 180,000 registros]

- ✓ Universidad Nacional Autónoma de México (1993). *BIBLAT: Bibliografía sobre América Latina e Información*. Disco Compacto y guía. México : UNAM, Centro de Información Científica y Humanística. [cuatro bancos, 290,000 registros].

- ✓ Universidad Nacional Autónoma de México (1993). *Catálogo de libros de las bibliotecas de la UNAM [LIBRUNAM].* Disco compacto. México : UNAM, Dirección General de Bibliotecas; Dirección General de Servicios de Cómputo para la Administración. [un banco, 500,000 registros]

- ✓ Universidad Nacional Autónoma de México (1993). IRESIE: Banco de Datos sobre Educación 1996. 2ª. ed. Disco compacto y guía. México : UNAM, Centro de Investigaciones y Servicios Educativos. Dirección General de Servicios de Cómputo para la Administración. [cuatro bancos, 43,000 registros]

- ✓ Universidad Nacional Autónoma de México (1993). *SERIUNAM: Banco de datos de publicaciones periódicas 1993.* Disco compacto y guía. México : UNAM, Dirección General de Bibliotecas; Dirección General de Servicios de Cómputo para la Administración. [un banco, registros]

- ✓ El Colegio de México (1994). *CD-Colmex: Catálogo de la Biblioteca "Daniel Cosío Villegas" 1940-1993.* Disco compacto y guía. México : UNAM, Centro de Información Científica y Humanística. [un banco, 31,609 registros bibliográficos con 3'164,132 acervos]

- ✓ Universidad Nacional Autónoma de México (1994). *BIBLAT: Bibliografía sobre América Latina e Información*. 2ª. ed. Disco Compacto y guía. México : UNAM, Centro de Información Científica y Humanística. [cuatro bancos, 300,000 registros].

- ✓ Universidad Nacional Autónoma de México (1994). *ARIES: Acervo de Recursos de Instituciones de Educación Superior. 1993-1994*. 3ª. ed. Disco compacto y guía. México : UNAM, Dirección General de Intercambio Académico. Centro de Información Científica y Humanística. [dos bancos, 27,000 registros]

- ✓ El Colegio de México (1995). *CD-Colmex: Catálogo de la Biblioteca "Daniel Cosío Villegas" 1940-1995*. 2ª. ed. Disco compacto y guía. México : UNAM, Centro de Información Científica y Humanística. [un banco, 296,000 registros]

- ✓ El Colegio Mexiquense (1995). *Bibliografía del Estado de México*. Disco compacto y guía. México : UNAM, Centro de Información Científica y Humanística. [un banco, 8,550 registros con imágenes]

- ✓ Secretaría de Salud (1995). *RENCIS: Catálogo Colectivo de Publicaciones Seriadas.* 4ª. ed. Disco compacto y guía. México : UNAM, Centro de Información Científica y Humanística. [un banco, 6959 registros, 20,565 acervos]

- ✓ Universidad Nacional Autónoma de México (1995). *ARIES: Acervo de Recursos de Instituciones de Educación Superior. 1994-1995*. 4ª. ed. Disco compacto y guía. México : UNAM, Dirección General de Intercambio Académico. Centro de Información Científica y Humanística. [dos bancos, 28,000 registros]

- ✓ Universidad Nacional Autónoma de México (1995). *Ciencias Sociales y Humanidades.* Disco compacto y guía. México : UNAM, Coordinación de Humanidades. Centro de Información Científica y Humanística. [dieciocho bancos, 77,000 registros]

- ✓ Universidad Nacional Autónoma de México (1995). *Shock Waves in Medicine.* Dos disquetes 3.5 " y guía. México : UNAM, Instituto de Física, Centro de Información Científica y Humanística. [un banco, 3000 registros]

- ✓ Universidad Nacional Autónoma de México (1995). *SERIUNAM: Banco de datos de publicaciones periódicas 1995*. 2ª. ed. Disco compacto y guía. México : UNAM, Dirección General de Bibliotecas; Centro de Información Científica y Humanística. [un banco, 42029 registros bibliográficos con 5'499,258 acervos]

- ✓ Fideicomiso para la Cultura México-USA (1996). *Abimex: Antigua Bibliografía Mexicana.* Disco compacto y guía. México : UNAM, Centro de Información Científica y Humanística. [un banco con imágenes, 10,417 registros].

- ✓ Secretaría de salud (1996). *Bibliomex Salud: Bibliografía Mexicana en Biomedicina y Salud.* Disco Compacto y guía. México : UNAM, Centro de Información Científica y Humanística. [un banco, 14,391 registros].

- ✓ Universidad Nacional Autónoma de México (1996*). ARIES: Acervo de Recursos de Instituciones de Educación Superior. 1995-1996.* 5ª. ed. Disco compacto y guía. México : UNAM, Dirección General de Intercambio Académico. Centro de Información Científica y Humanística. [dos bancos, 34,000 registros]

- ✓ Universidad Nacional Autónoma de México (1996). IRESIE: Banco de Datos sobre Educación 1996. 3ª. ed. Disco compacto y guía. México : UNAM, Centro de Investigaciones y Servicios Educativos. Centro de Información Científica y Humanística. [cuatro bancos, 50,000 registros]

- ✓ Universidad Nacional Autónoma de México (1997). *ARIES: Acervo de Recursos de Instituciones de Educación Superior. 1996-1997.* 6ª. ed. Disco compacto y guía. México : UNAM, Dirección General de Intercambio Académico. Centro de Información Científica y Humanística. [dos bancos, 38,000 registros]

- ✓ Secretaría de Salud (1997). *RENCIS: Catálogo Colectivo de Publicaciones Seriadas*. 6ª. ed. Disco compacto y guía. México : UNAM, Centro de Información Científica y Humanística. [un banco, 5,335 registros, 18,339 acervos]

- ✓ Universidad Nacional Autónoma de México (1998). *ARIES: Acervo de Recursos de Instituciones de Educación Superior. 1997-1998.* 7ª. ed. Disco compacto y guía. México : UNAM, Dirección General de Intercambio Académico. Centro de Información Científica y Humanística. [dos bancos, 42,000 registros]

- ✓ Secretaría de Salud (1999). *RENCIS: Catálogo Colectivo de Publicaciones Seriadas*. 8ª. ed. Disco compacto y guía. México : UNAM, Dirección General de Servicios de Cómputo Académico. [un banco, 8,926 registros, 29,791 acervos]

- ✓ Universidad Nacional Autónoma de México (1999). *ARIES: Acervo de Recursos de Instituciones de Educación Superior. 1999-2000.* [8ª] 7ª. ed. Disco compacto y guía. México : UNAM, Dirección General de Intercambio Académico. Dirección General de Servicios de Cómputo Académico. [dos bancos, 44,000 registros]

REFERENCIAS BIBLIOGRÁFICAS

Keuffel, Warren. *Battle of the modeling techniques.* DBMS On-line. August 1996. *http://www.dbmsmag.com/9608d14.html*, 1996.

Knuth, Donald. The Art of Computer Programming, vol. 1: Fundamental Algorithms ; Reading, Mass.: Addison-Wesley, 2nd ed, 1973.

Knuth, Donald. *The Art of Computer Programming: vol. 3: Sorting and Searching*; Reading, Mass.: Addison-Wesley, 1973.

Martin, James. *Database Analysis and design*. Prentice-Hall, 1992.

National Institute for Standards and Technology. *Dictionary of Algorithms, Data Structures and Problems. http://hissa.nist.gov/dads/terms.html*, 1999

Standish, T. *Data Structure Techniques.* Addison-Wesley. pp. 191-210, 1980.

Standish, Thomas. *Data structures, Algorithms and software principles*; Addison-Wesley, 1994. 748 pp.,

Sunday, Daniel. A very fast substring search algorithm. En: *Communications of the ACM*, vol. 33, No. 8, August 1998. pp. 132-142.

University of Western Australia. *Data Structures and Algorithms. http://swww.ee.uwa.edu.au/~plsd210/ds/ds_ToC.html*, 1998. Capítulos 3: Data Structures. Capítulo 4: Searching. Capítulo 5: Searching Revisited.

Voutssás, J. Ruiz-Velasco, M. *Algoritmos Especializados para Almacenamiento y Recuperación de Información Bibliográfica.* México, 2000 : UNAM, Centro Universitario de Investigaciones Bibliotecológicas.