

Ma. Isabel García Hidalgo

Laura Freidberg Gojman

MÁS ALLÁ DE LOS MANEJADORES DE BASES DE DATOS

UNA APLICACIÓN BIBLIOGRÁFICA



EL COLEGIO DE MÉXICO

MÁS ALLÁ DE LOS MANEJADORES DE BASES DE DATOS
Una aplicación bibliográfica

UNIDAD DE CÁLCULO

**MÁS ALLÁ DE LOS MANEJADORES
DE BASES DE DATOS**
Una aplicación bibliográfica

Ma. Isabel García Hidalgo
y
Laura Freidberg Gojman



EL COLEGIO DE MÉXICO

Open access edition funded by the National Endowment for the Humanities/Andrew W. Mellon Foundation Humanities Open Book Program.



*The text of this book is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License:
<https://creativecommons.org/licenses/by-nc-nd/4.0/>*

Portada de Pablo Reyna y Francisco Moreno

Primera edición, 1991

**D.R. © El Colegio de México
Camino al Ajusco 20
Pedregal de Santa Teresa
10740 México, D.F.**

ISBN 968-12-0465-4

Impreso en México / *Printed in Mexico*

ÍNDICE

Prólogo	7
Introducción	11
I. Estructuración de la información para recuperación bibliográfica	13
II. Técnicas de indexamiento	17
III. Un esquema estructural para la información bibliográfica	21
IV. El sistema de recuperación bibliográfica	29
Conclusiones	43
Ancxos	
A. Programas y algunas de las tablas que utilizan	47
B. Ejemplos de productos	91
C. Adaptación del formato MARC para El Colegio de México	99
Bibliografía	125

PRÓLOGO

En este libro se presenta una solución computacional a un problema de recuperación de información bibliográfica. Esta solución, desarrollada por las autoras del libro, es una aplicación de un método general de tratamiento de información no numérica. Dicho método surgió a partir de una serie de soluciones a problemas que nos fueron planteados desde el ámbito de las ciencias sociales y las humanidades. El libro incluye reflexiones sobre la complejidad de los procesos de recuperación de información. Posteriormente se describen, desde un punto de vista formal, las componentes esenciales del sistema de cómputo que son el corazón del método mencionado. Por último, se anexan algunos programas con el propósito de que el lector llegue a manejar el conocimiento necesario para aplicar este método a sus propios problemas.

Esbozamos a continuación las condiciones en las que surge este trabajo, con el fin de que el lector dé la importancia apropiada al método y a la solución bibliográfica particular que se describe en la obra.

Durante la década de los años ochenta, en El Colegio de México como en muchos otros lugares, se dió un incremento considerable en la demanda de solución computacional a problemas de carácter no numérico. Muchos de ellos estaban relacionados con algún tipo de acervo documental; entre otros podemos mencionar el manejo de archivos con información histórica, la creación de catálogos de material literario, el manejo de toda suerte de bibliografías especializadas, el tratamiento de fichas temáticas y de investigación, y la producción de juegos de fichas catalográficas.

Los primeros años de esta década constituyen el último período en la vida de El Colegio en el que se dispone solamente de un equipo electrónico centralizado en la Unidad de Cómputo. A partir de 1984 El Colegio participa del acelerado proceso de incorporación a sus quehaceres de investigación de la moderna tecnología computacional. En el presente El Colegio cuenta con microcomputadoras al alcance de cada investigador, una red local, y acceso a redes internacionales. Junto con esto, se usa multitud de paquetes para aplicaciones específicas tales como procesadores de palabras, manejadores de bases de datos, hojas de cálculo, paquetes estadísticos y tipográficos, y otros. Podemos decir que en la década de los ochenta El Colegio vive su gran transformación en el uso de equipo electrónico.

La transformación tecnológica mencionada influyó sobre el planteamiento de soluciones computacionales a los problemas de tratamiento de información no numérica. La mayoría de ellos fueron planteados antes de que en El Colegio se contara con microcomputadoras; algunos lo fueron con-

tando ya con microcomputadoras pero antes de disponer de manejadores de bases de datos capaces de manipular registros de longitud variable. Todos los problemas planteados exigieron, en su momento, una solución computacional que apoyara el avance de los correspondientes proyectos de investigación. Además, todos esos proyectos tuvieron un carácter duradero, es decir, la solución se debería aplicar durante muchos años.

Exigencias y condiciones como las anteriores, sumadas a la transformación tecnológica acelerada, obligan al investigador de cómputo a construir modelos —estructuras y algoritmos— con base en los cuales se implementen los sistemas computacionales que den solución al problema planteado. Los modelos creados deben posibilitar la producción de resultados específicos en el momento deseado. Al mismo tiempo, los modelos deben permitir la redefinición de los productos resultantes —sin menoscabo del trabajo ya realizado por el investigador en ciencias sociales o humanidades— con el objeto de aprovechar las ventajas de las herramientas computacionales que el desarrollo tecnológico va poniendo a nuestra disposición.

El planteamiento de una solución computacional depende en forma absoluta de los recursos de cómputo en los que se implementará dicha solución. La solución puede variar dependiendo de la cantidad de memoria disponible, la cantidad de espacio en disco, el lenguaje de programación o paquete de aplicación específica que se utilizará, etc. Ahora bien, los proyectos que hemos mencionado se caracterizaban por exigir el manejo de grandes volúmenes de información con un alto grado de variabilidad de una a otra unidad de información. Los usuarios no podían tomar la decisión de abreviarla o manejarla en forma incompleta debido al gran valor que dicha información tenía para ellos. El equipo electrónico disponible se caracterizaba por la escasez de espacio en disco. Por último, el instrumental computacional con el que contábamos fue, cuando más, un manejador de bases de datos de registro fijo. Fueron estas condiciones las que nos llevaron a considerar indispensable el tratamiento de información en formatos de longitud variable.

La necesidad de programar tantos sistemas de cómputo como productos requerían los investigadores de los proyectos, todos haciendo un manejo de información documental codificada bajo formatos de longitud variable, llevó a la creación de una metodología computacional para facilitar el desarrollo de tales sistemas. La metodología incluye: la acomodación de un esquema estructural a los datos por analizar, el uso de un formato variable con identificadores y separadores de campos para almacenar la información, la aplicación de análisis léxico dirigido por tablas en las que se representan autómatas, y el diseño y programación de subrutinas de aplicaciones específicas. La facilidad en la aplicación de este método estriba en que todos los sistemas utilizan el mismo algoritmo para construir la tabla que representa a los autómatas y el mismo algoritmo de utilización de esa tabla para efectuar el reconocimiento de la información.

La obra que presentamos a la consideración del lector aborda una aplicación de la metodología mencionada. Corresponde al tratamiento de la in-

formación bibliográfica de las obras de la Biblioteca Daniel Cosío Villegas de El Colegio de México, en particular, del Catálogo de Obras sobre Asia y África. El proyecto de automatización de este catálogo se inició en el año de 1983. El banco de información se empezó a conformar en un equipo PDP/11-70, se continuó en un equipo ALTOS 586, y hoy está siendo manejado en un sistema personal de IBM. Programas de cómputo acordes con la metodología mencionada —desarrollados en Fortran para la PDP— pudieron apoyar el perfeccionamiento en la aplicación de las normas de catalogación. Posteriormente, nuevos programas creados bajo el mismo método —en lenguaje C para el sistema UNIX del equipo ALTOS— permitieron la producción de juegos de fichas catalográficas, así como la elaboración de la bibliografía de obras sobre Asia y África. Fue ésta la etapa en la que se inició el uso complementario de nuestros programas y el manejador de bases de datos INFORMIX. Por último, programas similares —en C de ALTOS— se utilizaron para la construcción de registros MARC que trasferidos al sistema personal IBM son directamente importados al paquete MICROISIS. Toda esta trayectoria de avance y cambio para apoyar computacionalmente un proyecto continuo —como es el de catalogación y clasificación de obras de nuestra Biblioteca— ha podido ser manejada con programas diseñados bajo la metodología que hemos mencionado.

Los dos primeros capítulos del libro están encaminados a resaltar la importancia y los problemas de la labor de estructuración de la información al interior de cada unidad y en su conjunto, en virtud de que la exactitud en un proceso de recuperación de información depende directamente del nivel de detalle que se alcance en la discriminación de las partes componentes de cada unidad de información y de la codificación de las relaciones establecidas dentro del conjunto de unidades.

En el capítulo tercero se analiza el formato MARC, en particular su adaptación a la catalogación de El Colegio de México (MARCOLMEX), que es el esquema estructural elegido para catalogar la información bibliográfica.¹ En el mismo capítulo se discute el formato de longitud variable que es el elegido para almacenar en la computadora la información catalográfica.

En el capítulo cuarto se presenta formal y detalladamente el modelo computacional que posibilita el análisis de la información y la recuperación bibliográfica. Concientes de que existe una distancia entre la descripción de un modelo y su implementación, decidimos incluir algunos programas —en lenguaje C— en los que el lector pueda observar la forma en que se imbrican las subrutinas de aplicaciones específicas al reconocimiento léxico.

El método de programación que esta obra ejemplifica, ha ido construyéndose a partir de generalizaciones en la aplicación de soluciones particulares. Debemos mencionar algunos sistemas computacionales claves sin los cuales no hubiera sido posible adquirir la experiencia, en el tratamiento de

¹ La definición del MARCOLMEX se debe al grupo de profesionales del Departamento de Procesos Técnicos de la Biblioteca Daniel Cosío Villegas. En este libro se reproduce como anexo el MARCOLMEX con el único propósito de darle al lector todos los elementos que le permitan juzgar el alcance de la solución computacional al problema bibliográfico.

información documental, requerida para plantear las generalizaciones. Ellos son: la parte componente del Sistema Computacional del Diccionario del Español de México mediante la que se efectúa análisis morfológico, el sistema para la producción de la serie de Guías de Protocolos del Archivo General de Notarías de la Ciudad de México, y el sistema para la generación del Catálogo de Textos Literarios Novohispanos. Queremos agradecer a los investigadores responsables de estos proyectos su confianza al poner en nuestras manos sus problemas y darnos la oportunidad de resolverlos y aprender de ello.

Desde el punto de vista de cómputo, el proyecto de automatización de la biblioteca de El Colegio de México —dirigido por Álvaro Quijano— y el proyecto de automatización del catálogo de obras sobre Asia y África —coordinado por Thiago Cintra y Celma Agüero— han sido un mismo proyecto. En algunas ocasiones pudimos ver al segundo como proyecto piloto, pero siempre fue nuestro objetivo aportar soluciones que sirvieran al primero. A los coordinadores del proyecto sobre Asia y África les damos las gracias por darnos la libertad de experimentar soluciones para apoyar su proyecto. A Porfirio Díaz, Patricia González y Ángela Ojeda se debe que el banco de información bibliográfica sobre Asia y África sea una realidad, ya que durante varios años se encargaron de recatalogar los elementos faltantes de la información y verter en la computadora los casi 18 000 registros catalográficos. Nuestro profundo agradecimiento a Álvaro Quijano, a Clotilde Tejeda, y a todo el equipo de catalogadores de nuestra Biblioteca por haber compartido con nosotros sus conocimientos y sus problemas, dando lugar con ello al trabajo que se presenta en este libro.

INTRODUCCIÓN¹

En este trabajo se presenta un sistema computacional para manejar información bibliográfica. Esta información se caracteriza por su alto grado de variabilidad en lo que se refiere a categorías y tamaños.

Las características fundamentales del sistema que se presenta son:

1) Trabaja sobre información codificada en un formato que no impone limitaciones de tamaño o número de ocurrencias de los campos de una unidad de información. Para manejar adecuadamente este tipo de información utiliza algoritmos rápidos de reconocimiento, identificación y selección de los campos de una unidad de información, así como algoritmos de transformación de los contenidos de campos seleccionados.

2) Da la posibilidad de que la transformación de contenidos para campos seleccionados tenga como producto un subconjunto, en formato fijo, del total de la información.

El sistema permite que los usuarios definan, de acuerdo con sus necesidades o etapa de sus investigaciones, cuáles campos de información incluirán en cada subconjunto —en formato fijo— que van a crear.

Los subconjuntos se obtienen de la información completa por medio del sistema computacional desarrollado como parte de este trabajo, y pueden ser posteriormente procesados con un manejador de bases de datos.

Al usar las herramientas de selección y agrupamiento que proporcionan los manejadores de bases de datos se pueden construir nuevos subconjuntos tomando en cuenta ya no los campos sino los contenidos de los campos seleccionados en la etapa de definición de la base de datos.

El sistema permite recuperar la información completa de las unidades seleccionadas en el manejador de bases de datos. Estas pueden procesarse nuevamente por el sistema y obtener, así, para algunas aplicaciones, nuevos productos que no podían conseguirse con el manejador.

En el caso que nos ocupa, el material bibliográfico con el que se trabaja corresponde a obras sobre Asia y África catalogadas y clasificadas, utilizando el formato MARC, por la Biblioteca de El Colegio de México que es una biblioteca especializada en Ciencias Sociales y Humanidades. Además de los usuarios normales de una biblioteca, el material bibliográfico sobre Asia y África es utilizado por investigadores sumamente especializados, para quienes la recuperación que puede hacerse bajo los criterios de catalogación y clasificación de la propia biblioteca resulta poco precisa. Con el sistema

¹ El trabajo que se discute en este libro tuvo como primer resultado la tesis de Laura Freidberg para obtener el grado de Matemática.

propuesto se da a estos investigadores la posibilidad de utilizar criterios más finos sin invalidar los criterios más generales que dan buenos resultados de recuperación a usuarios menos especializados. Esto puede hacerse agregando descriptores especializados al conjunto de información original, así como resúmenes escritos en lengua natural en los cuales estén marcadas algunas palabras claves.

El proceso de agregar información especializada no afecta a los usuarios no especializados ya que sus productos pueden ser obtenidos sin ninguna modificación. De modo paralelo, los usuarios especializados pueden incluir directamente en sus bases de datos estos nuevos descriptores o las palabras claves marcadas en los resúmenes, y conservar o descartar los descriptores más generales.

I. ESTRUCTURACIÓN DE LA INFORMACIÓN PARA RECUPERACIÓN BIBLIOGRÁFICA

El término recuperación de información se usa en la literatura sobre el tema en dos sentidos distintos. A continuación se discute la diferencia entre la acepción genérica del término *recuperación de información* y la acepción específica que tiene cuando se considera en relación con el término *recuperación bibliográfica*.

Un científico que realiza una investigación está involucrado, casi sin excepción, en dos tipos de búsqueda que, aunque están relacionados en su quehacer, conviene mantener separados en el análisis si se quiere evitar confusión.

Es posible que desee obtener respuestas muy detalladas a ciertas preguntas específicas que surgen de su investigación o necesitar las referencias de toda la literatura que contiene material relevante a su problema. Llamaremos a lo primero *recuperación de información* y a lo segundo *recuperación bibliográfica*.

La respuesta al primer tipo de búsqueda sería una o más declaraciones afirmativas, mientras que para el segundo sería una lista de referencias bibliográficas, con o sin notas, resúmenes o copias de la literatura relevante.

Para que el conocimiento sobre un tema sea utilizado en un sistema de recuperación de información, debe estructurarse como un conjunto de proposiciones que conserven la información sobre hechos relativos al tema, o la que es considerada de interés y valor para el sistema. Debe hacerse hincapié en que no cualquier documento puede ser usado en un recuperador de información debido a que generalmente contienen proposiciones meta-lingüísticas sobre los resultados obtenidos en la literatura original, en vez de un sumario de los resultados en sí mismos.

En cambio, para que una referencia bibliográfica sea seleccionada mediante un sistema de recuperación bibliográfica es necesario que se use un conjunto de símbolos cuya función es dar una pista al usuario de qué tan útil puede serle el documento. Estos símbolos pueden ser:

a) Asociados a la referencia bibliográfica. En este caso se caracterizan por ser palabras sueltas, frases, encabezamientos de materia, descriptores, etc., que se pueden tomar directamente del documento original o que pertenecen a algún lenguaje controlado (tesaurus, etc.).

b) Obtenidos a través de un análisis ya sea de la información contenida en las referencias bibliográficas, o de la contenida en las notas o resúmenes asociados a dichas referencias. Estos símbolos se pueden obtener, por ejemplo, calculando la frecuencia de las palabras que aparecen (sin tomar en

cuenta palabras gramaticales como artículos, preposiciones, etc.) y marcando o eligiendo las más frecuentes.

Usaremos el término “descriptor” como nombre genérico de los símbolos asociados o los elementos obtenidos mediante el análisis.

Estos descriptores señalan, de alguna manera, el tipo de información contenida en el documento, pero no incluyen —como podría pensarse— la información en sí misma. Es un error suponer que una palabra o una frase contiene información en el mismo sentido que la tiene una proposición; la información contenida en una proposición no es la suma o combinación de la información contenida en las frases que la constituyen.

Podemos explicar la confusión que ha existido entre recuperación de información y recuperación bibliográfica con base en los siguientes hechos: para llegar a responder una consulta a un sistema de recuperación de información a menudo se necesita hacer primero una recuperación bibliográfica con el propósito de seleccionar los documentos que contienen información relevante, y luego obtener la respuesta a la pregunta solicitada buscando en el lugar preciso de los documentos seleccionados. Además, algunas veces se hace tan sólo una recuperación bibliográfica porque no se tiene un buen sistema de recuperación de información directo. Bajo estas circunstancias la recuperación bibliográfica es un sustituto muy pobre de un sistema de recuperación de información.

Sin embargo, ningún sistema de recuperación de información (si existe) podrá remplazar de manera total la lectura de la literatura original que contiene la información, ya que el valor de ésta consiste principalmente en la estimulación que da a través de su línea general de argumentación y método, más que su riqueza en hechos específicos. Por esto los sistemas de recuperación bibliográfica no deben verse sólo como una primera fase de los sistemas de recuperación de información sino que tienen una importancia propia.

Con la distinción entre estas dos formas de recuperación debe dejarse claro también que los métodos para resolver cada uno de los problemas no son necesariamente los mismos, aunque los métodos desarrollados para alguno de estos sistemas puedan aplicarse, con o sin adaptaciones, al otro; y las consideraciones teóricas sobre uno sean relevantes al otro.

Este trabajo se centra en el problema de recuperación bibliográfica.

Discutamos a continuación, dentro del marco de la recuperación bibliográfica, la necesidad de estructurar la información —material textual— y algunos de los problemas que se presentan.

Supongamos, por ejemplo, que tenemos una obra titulada *Escritos sobre la historia del México antiguo*, cuyo autor corporativo es la Coordinación de Historia de la Facultad de Filosofía y Letras de la Universidad Nacional Autónoma de México; la obra ha sido editada por Editorial México, en México, en 1980. Supongamos también que elegimos un encabezamiento de materia o un descriptor temático como “México - Historia - 1492-1521”.

Es claro que cada ocurrencia del concepto “México” tiene un valor diferente: en el título mencionado el término “México” refiere al tema sobre

el cual se escribe; en la segunda ocurrencia es parte del nombre que precisa cuál institución es autora de la obra; en la tercera, es parte del nombre de la editorial que publicó el libro; en la cuarta, hace una referencia geográfica al lugar donde se publicó; y, en la última, es el tema que ha sido aprobado por el catalogador siguiendo los estándares de una institución específica. Difícilmente encontraremos un problema de recuperación para el cual no sea necesario discriminar los diferentes valores de estas ocurrencias.

El valor que tiene cada ocurrencia de un dato específico, por ejemplo la palabra "México", es a su vez información que debe captarse para realizar una adecuada recuperación en la computadora.

Podríamos decir que la información *Título* cualifica a la información "Escritos sobre la historia del México antiguo"; la información *Autor* cualifica a "Universidad Nacional Autónoma de México, Facultad de Filosofía y Letras, Coordinación de Historia"; la información *Editorial* cualifica a "Editorial México"; el concepto "México" tiene la cualidad de ser el *Lugar de publicación*; y, "1980" la de ser el *Año de publicación*.

Cuando hemos podido establecer reglas precisas y consistentes para discriminar los campos de autor, de título, etc., en una obra, decimos que esa información está "bien estructurada".

Una *referencia bibliográfica* es, entonces, el conjunto de datos discriminados que identifican a la obra. Denominaremos *unidad de información* de un sistema de recuperación bibliográfica a las referencias bibliográficas junto con la información asociada a la que hemos llamado genéricamente descriptores.

Muchas veces no es posible discriminar o predefinir partes específicas en la información. En esta situación diremos que la información está "mal estructurada". Debe quedar claro que lo bien o mal estructurado de la información depende del propósito de la recuperación. Así, un resumen estará bien estructurado si se ha discriminado del resto de la información y se desea usar como una componente de la referencia bibliográfica. Sin embargo, un resumen estará mal estructurado en el momento en que nuestro propósito sea analizarlo para discriminar la información que contiene.

No debemos olvidar que el valor de estos resúmenes es que proporcionan una gran cantidad de información al lector, ya que están escritos en lengua natural y conservan la línea general de argumentación o método, además de la descripción de los hechos específicos fundamentales; esa información no puede proporcionarse al lector mediante un análisis por computadora debido a que carece de una estructura adecuada para ello.

La aplicación de los criterios de discriminación —usualmente llamada análisis e indexamiento— nos permite imponer una estructura a la información que será la que se aproveche durante la recuperación. La exactitud del proceso de recuperación dependerá directamente del nivel de detalle que se haya alcanzado en la discriminación de las categorías de información.

Desde hace mucho tiempo los científicos de la computación se han esforzado por proponer métodos automáticos para efectuar análisis del material textual con los que se pueda dar estructura a información mal estructu-

rada. Entre los métodos propuestos se conocen el uso de las frecuencias de los términos en un texto, la aplicación del análisis contextual y el sintáctico, los métodos semánticos, etcétera; . . . la lista es larga y mal estructurada.

Con estas técnicas se han obtenido listas, producidas por computadora, de términos asociados con el material textual. El hecho de que estas listas de descriptores sean tan diferentes entre sí, en función de las técnicas aplicadas, el orden de aplicación, el material analizado y el programa utilizado, muestra que aún no es eficiente la estructuración automática de la información. No se puede cambiar la información mal estructurada a bien estructurada con técnicas computacionales por sí solas aún teniendo un propósito definido; seguirá siendo una tarea humana en la que la computadora deberá usarse como una herramienta de ayuda.

Ni el hombre ni las máquinas pueden, por sí mismos, realizar con éxito las tareas de análisis e indexamiento necesarias para una recuperación de información eficiente. Se requiere una gran cantidad de juicio humano y de capacidad intelectual, que no han podido ser incorporados a procesos computacionales. Pero también se requiere el uso de las computadoras por la rapidez de sus operaciones y las facilidades que ofrecen para manejar grandes volúmenes de información, lo cual está más allá de las posibilidades prácticas del hombre.

Para que una solución sea eficiente habrá que encontrar el justo medio en la separación de las tareas que realice el hombre y las que se efectúen mediante las computadoras. En virtud de que el "justo medio" depende de cada problema específico, una solución formada de herramientas que puedan ser adaptadas de manera simple para los diversos problemas será eficiente. Uno de los propósitos de este trabajo es presentar herramientas adaptables.

En la mayoría de las áreas en las que se intenta el uso de sistemas de información se tiene una mezcla intrínseca de información bien estructurada y mal estructurada. Esta mezcla es el elemento esencial para que la información sea útil ya que si se separa perdería gran parte de su significado para los usuarios.

El uso que el hombre haga de los sistemas computacionales para apoyo a sus tareas de análisis e indexamiento de la información, debe permitir el manejo conjunto de los dos tipos de información, la mal estructurada y la bien estructurada. En algunos casos el hombre podrá servirse de la información bien estructurada para organizar la información mal estructurada y posibilitar con ello un avance en la estructuración.

II. TÉCNICAS DE INDEXAMIENTO

En la sección anterior hemos analizado la importancia y el significado de estructurar la información para un sistema de recuperación bibliográfica y hemos concluido que debe permitirse el manejo conjunto de la información bien estructurada y la mal estructurada.

En virtud de que el almacenamiento de información está íntimamente relacionado con los propósitos de la recuperación queremos distinguir entre dos modalidades de búsqueda: aquélla en la que el usuario ha hecho la selección de la obra antes de utilizar el sistema de recuperación bibliográfica, y aquélla en la que el usuario hará la selección mediante el uso del sistema.

En el primer caso se conocen los elementos necesarios para identificar unívocamente a la obra seleccionada —por lo general autor y título— y se desea recuperar la unidad de información completa que dará al usuario otros datos como la signatura topográfica o el resumen de la obra, que le permitirán respectivamente localizar ésta o juzgar si su selección previa fue adecuada.

En el segundo caso es necesario manejar el conjunto completo de información para seleccionar las obras que se refieren a un tema determinado. Con el fin de efectuar esta selección se usan —ya sea aislados o combinados mediante relaciones lógicas— los descriptores que, como habíamos mencionado, tienen la función de dar una pista al usuario de qué tan útil puede serle el documento.

Para asignar dichos descriptores al material textual existen principalmente dos formas. Una de éstas, que se ha utilizado tanto en métodos manuales como automáticos, es elegir como descriptores los términos tal como se encuentran en el material textual que se está describiendo. A ésta se le denomina *indexamiento libre*. La otra es construir los descriptores y organizarlos para después asociarlos al material textual. Esta forma se conoce como *indexamiento de tesouro*. Cada una tiene algunas ventajas, así como ciertas desventajas.

A continuación discutiremos los dos métodos con el propósito de mostrar más adelante que el sistema descrito en esta obra ofrece herramientas para aprovechar complementariamente ambos métodos y, por lo tanto, cubrir las necesidades de usuarios de diversos niveles.

En el *indexamiento libre*, el indexador puede, en teoría, elegir cualesquiera expresiones, sean palabras sueltas o frases, que aparezcan en el material textual que está describiendo —expresiones textuales— o bien expresiones derivadas de éstas y, en general, cualquier término que considere adecuado para representar el tema o materia del que trata el documento.

Podría pensarse que en estas condiciones se tiene una libertad absoluta de elegir los descriptores; sin embargo, debe tenerse presente que existen limitantes como son la tradición, el sentido común, restricciones del lenguaje ordinario, entrenamiento psicológico e institucional y acondicionamiento cultural, las cuales restringen en diversa medida la libertad de indexamiento.

En la práctica el indexamiento libre es aquel en el que los términos o expresiones que pueden ser escogidos como descriptores deben ser seleccionados únicamente entre los términos o expresiones que aparecen en el material textual; es decir, los indexadores examinan el material textual y subrayan o indican los términos que consideran que son importantes o indicativos de la materia o asunto al que se refiere el documento.

Los resultados de un indexamiento libre pueden presentar severas inconsistencias debidas a las distintas apreciaciones de un indexador con respecto a otros, así como las de un indexador en distintos momentos o circunstancias. Las inconsistencias se minimizan cuando el indexador y el buscador son especialistas en un área del conocimiento, ya que ambos perciben de manera similar la organización de los conceptos de su área de especialidad y, además, utilizan de manera natural un vocabulario estandarizado tomado de la terminología de su materia.

Por tanto, el indexamiento libre es ventajoso siempre y cuando los indexadores y quienes consultan sean especialistas en la materia a la que se refieren los documentos.

En el indexamiento de tesauro, los descriptores se eligen de listas que contienen términos previamente seleccionados asociados a cierta información sobre su uso; éstas se organizan o estructuran mediante relaciones entre los términos. La mayoría de los tesauros son de propósitos especializados en el sentido de que sus términos se toman del vocabulario de una disciplina particular. Junto con cada término listado se pueden incluir algunas referencias a otros términos que tienen el mismo significado, significados más específicos o más genéricos, o términos cuyos significados están relacionados.

El propósito de los tesauros es organizar los conceptos de un área específica del conocimiento en campos semánticos o conceptuales. La organización se establece mediante dos tipos de relaciones: las jerárquicas y las de afinidad.

La organización jerárquica incorpora los grados relativos de especificidad o generalidad entre los distintos conceptos de un campo semántico. Esta organización no corresponde a un ordenamiento de lo más importante o más usado hacia lo menos importante o menos usado, sino que se basa en conexiones que van de lo general hacia lo específico, como son las relaciones entre un conjunto y sus elementos, el todo y sus partes, el género y sus especies, etcétera.

La relación de afinidad se emplea para establecer un nexo entre dos conceptos estrechamente relacionados entre los que no existe una verdadera relación jerárquica, como los diversos aspectos de un proceso (por ejemplo, el aspecto fisiológico y el aspecto anatómico de un proceso biológico).

Por tanto, un tesauro permite ampliar las asociaciones entre los concep-

tos, facilitar la búsqueda de conceptos periféricos a partir de un tema central y pasar de un campo conceptual a otros relacionados con él.

La organización conceptual de un tesoro se posibilita debido al establecimiento de un vocabulario estándar controlado. Éste debe seleccionarse de la terminología del área específica del conocimiento a la que se refiere el tesoro, de manera que constituya un vocabulario mínimo al cual pueda reducirse el universo que se pretende estructurar mediante el uso de sinónimos; en dicho vocabulario deben, además, quedar representadas todas las relaciones conceptuales jerárquicas y de afinidad.

Las ventajas del indexamiento mediante tesauros son: 1) la existencia de una especie de mapa que indica cómo los conceptos están relacionados entre sí, lo cual ayuda a los indexadores y a los buscadores a entender la estructura del campo; 2) el uso de un vocabulario estándar con referencias cruzadas entre sus términos —jerárquicas, de preferencia, de afinidad—, lo cual asegura que se pueda elegir el término que apropiadamente represente a un concepto dado dentro de la estructura clasificadora del tesoro y que se use únicamente el término preferente o autorizado de entre el conjunto de sinónimos para indexar un concepto.

Las referencias cruzadas garantizan que al hacer una búsqueda se elija el término correcto, ya que si un indexador usara más de un sinónimo para representar un mismo concepto, los documentos se encontrarían dispersos bajo todos estos sinónimos, y un buscador que eligiera uno de éstos y encontrara documentos indexados bajo éste supondría que ha encontrado el término correcto y detendría su búsqueda sin saber que hay otros documentos útiles indexados bajo los sinónimos. El uso de las referencias jerárquicas y de afinidad permiten a un buscador ser más general o más particular sistemáticamente, si su primera elección de los términos de búsqueda produce ya sea muy pocas o demasiadas referencias al material almacenado.

Uno de los problemas clave en la construcción de un tesoro es la determinación inicial de los candidatos a descriptores. El mejor procedimiento para esto es que especialistas de una disciplina hagan la selección original. De aquí se desprende una propuesta de uso complementario del indexamiento libre y el de tesoro. La propuesta incorpora como primer requerimiento que se incluyan en las unidades de información resúmenes o sinopsis del documento original que, como habíamos dicho, proporcionan una gran cantidad de información al lector.

El tipo de resumen que proponemos que se incluya es el que, en ocasiones, producen los propios autores de los libros o artículos científicos; es decir, aquéllos hechos por especialistas. Simultáneamente a la producción de estos resúmenes los especialistas efectuarían un indexamiento libre sobre ellos produciendo así una sinopsis integrada que es, al mismo tiempo, un resumen en lengua natural con todas las ventajas que ya han sido mencionadas, y una serie de palabras clave que permitirán la recuperación de la unidad de información.

Con estas palabras clave se obtendrá la lista de candidatos a descriptores para la formación del tesoro. De esta manera, la lista de candidatos

incorpora el juicio de los especialistas, es decir, los elementos de la organización conceptual y la estandarización terminológica de su área de especialidad, que ellos manejan en forma natural.

Los especialistas envueltos en el proceso deben intentar indexar varios documentos para establecer el vocabulario mínimo que constituirá el tesau-ro, definir las relaciones jerárquicas y de afinidad entre los términos, y determinar en dónde son necesarias explicaciones que hagan más claros a los descriptores y dónde son relevantes, necesarias o deseables reglas interpretativas y aplicativas.

Por último, el indexamiento con el tesau-ro resultante puede ser auxiliado por procesos computarizados gracias al indexamiento libre que se tenía.

El agregado automático de los términos autorizados en el tesau-ro, correspondientes a determinadas palabras clave marcadas mediante el previo indexamiento libre es un ejemplo de tales procesos computarizados.

De la discusión anterior queda clara la necesidad de que la información que se almacena debe poseer una estructura tal que posibilite recuperación basada en indexamiento tanto libre como con tesau-ro para poder satisfacer las necesidades de cualquier tipo de usuario.

III. UN ESQUEMA ESTRUCTURAL PARA LA INFORMACIÓN BIBLIOGRÁFICA

En el capítulo anterior se discutió fundamentalmente la relación entre las características de la información y los posibles resultados de los procesos de recuperación. En este capítulo analizaremos la estructura específica que se ha decidido imponer a la información bibliográfica del Catálogo de Obras de Asia y África (COAA), así como la forma de almacenamiento en la computadora de dicha información. Dejaremos para el siguiente capítulo la descripción de los métodos computacionales mediante los cuales se efectúan los procesos de recuperación y transformación de la información.

Como se mostró en los capítulos anteriores, al imponer una estructura a la información se establecen las posibilidades de identificación de sus partes. Durante la definición de un esquema estructural para información bibliográfica, es necesario considerar el máximo nivel de detalle en la discriminación de las partes debido a que los procesos computacionales de discriminación no siempre son posibles mientras que los de agrupación son relativamente simples. Parafraseando en lenguaje aritmético diríamos que la definición del esquema estructural corresponde a la definición del conjunto de números primos; la imposición de este esquema a la información bibliográfica corresponde a encontrar —discriminar— los factores primos de cada unidad de información; y los procesos de agrupación corresponden al cálculo de submúltiplos a través de la operación con los factores primos.

Al tratar de definir el esquema estructural que debería imponerse a la información bibliográfica del COAA se consideró que, por lo menos, debería incluir discriminadamente la información de las fichas catalográficas que tradicionalmente habían sido producidas por la Biblioteca de El Colegio de México y que eran las que constituían dicho catálogo. Además, debería posibilitar que a cada unidad de información se agregaran los descriptores especializados que los investigadores del Centro de Estudios de Asia y África (CEAA) requerirían durante sus procesos de manipulación bibliográfica, así como los resúmenes de las obras correspondientes.

En 1982 la Biblioteca "Daniel Cosío Villegas" (BDCV) de El Colegio de México había decidido adoptar el formato MARC (*MACHine Readable Cataloging*) para la catalogación de sus obras, debido a que iniciaba la automatización de sus procesos técnicos.

A pesar de que el proyecto de automatización del COAA había nacido como un proyecto independiente del de automatización de la BDCV, se consideró importante conservar el mismo esquema estructural que usara la BDCV con el propósito de que ambos proyectos utilizaran el mismo sistema

computacional de manejo y recuperación de información catalográfica cuyo desarrollo se iniciaba en aquellos momentos. En realidad, en ese tiempo el proyecto de automatización del COAA se veía como un proyecto piloto de lo que sería el gran proyecto de automatización de la BDCV. Una de las ideas fundamentales era que al utilizar un estándar como MARC se posibilitaba una modalidad de catalogación compartida entre la Biblioteca como entidad normativa y los Centros de Estudios como entidades especializadas. Dicha modalidad consistiría en que la Biblioteca catalogara la información básica y de uso general de cada obra y los Centros agregaran la información propia de sus especialidades (descriptores y resúmenes). El uso de MARC no sólo permitiría obtener las ventajas de la catalogación compartida al interior de El Colegio de México sino que también permitiría el intercambio de información con otras bibliotecas tanto a nivel nacional como internacional ya que la comunidad bibliotecaria lo ha adoptado no sólo como su esquema estructural estándar sino también como un formato de intercambio del material catalográfico. Particularmente, en México, el formato MARC es utilizado por la Biblioteca Nacional, así como por el Sistema de Bibliotecas de la UNAM entre otras.

A continuación se proporciona un brevísimo resumen del surgimiento y alcance del formato MARC con el propósito de asentar la importancia de su adopción y poder exponer su composición y su forma de uso dentro de El Colegio de México.

Las investigaciones para establecer la posibilidad de usar técnicas automatizadas en la Library of Congress (LC) de Estados Unidos se iniciaron a fines de la década de los cincuenta. En 1965, con el apoyo del Council on Library Resources y del Committee on Automation of the Association of Research Libraries se llevó a cabo una conferencia en la que participaron representantes de universidades, de agencias de investigación, de agencias del gobierno y de la industria privada. En dicha conferencia se llegó a la conclusión de que la existencia de registros catalográficos estandarizados legibles por computadora facilitaría la automatización de las bibliotecas participantes. Dichos registros catalográficos deberían incluir todos los datos que aparecen en las tarjetas catalográficas junto con la información adicional necesaria para lograr la implantación de técnicas y procedimientos que permitieran el manejo automático de catálogos, búsquedas, indexamientos, y recuperación bibliográfica. Para facilitar la mencionada estandarización se llegó al acuerdo de que la LC diseñara, produjera y distribuyera los registros. Así, en diciembre de 1965 surgió el proyecto piloto denominado MARC bajo la dirección de Henriette D. Avram.

En abril de 1966 la LC estableció el formato MARC I para monografías y produjo cintas con registros MARC, los cuales fueron enviados, para su prueba, a las 16 bibliotecas participantes en el proyecto piloto a partir de octubre del mismo año. Con la experiencia ganada en el proyecto piloto y por el interés mostrado por la Bibliografía Nacional Británica, se diseñó el formato MARC II cuya característica principal fue conformar un formato de comunicación e intercambio de datos bibliográficos a nivel internacional.

El propósito al crear MARC II fue diseñar un formato que representara físicamente, en un medio legible por computadora, la información bibliográfica pertinente para todas las formas de material, tales como libros, series, mapas, música, artículos en revistas, etc. El formato resultante está formado por los siguientes tres componentes: 1) la estructura, o el "contenedor vacío"; 2) los designadores de contenido (etiquetas, indicadores y códigos de subcampo) usados para identificar explícitamente o caracterizar adicionalmente los elementos de los datos; y, 3) el contenido; es decir, los datos en sí mismos (nombre del autor, título, etc.).

La estructura de los registros MARC es lo suficientemente detallada que permite la inclusión o exclusión de los elementos de los datos según las necesidades de los usuarios. Durante el año de 1982 cuando se realizó el estudio para definir el formato de captura de los registros catalográficos de la BDCV, se revisó el MARC original y se seleccionó la lista de las etiquetas utilizables por El Colegio de México. Dicha lista incluyó todos los casos eventualmente posibles, aún aquellos que jamás se habían usado en la BDCV, como fue el caso del ISBN. Se adoptaron los códigos alfabéticos para subcampos del MARC original y se respetaron las etiquetas reservadas por la LC, por ejemplo, en la codificación de encabezamientos de materia. La adaptación del MARC a la catalogación de El Colegio de México (MARCOL-MEX), la cual se incluye como anexo en este trabajo, consta de 408 campos entre etiquetas y códigos de subcampos.

FORMA DE ALMACENAMIENTO DE LA INFORMACIÓN

Una vez definidas las categorías de información; es decir, los elementos del esquema estructural que van a diferenciarse, existen básicamente dos formas de identificarlos dentro de la computadora: el llamado formato de longitud fija o *formato fijo* y el formato de longitud variable o *formato variable*.

En el primero, la identificación de los diversos elementos se logra mediante su posición en el registro del archivo en computadora, o de manera equivalente a través de un nombre asignado a cada una de las posiciones predefinidas en el registro, como se hace en los manejadores de bases de datos. Por ejemplo, diríamos que el *Autor* ocupará los caracteres 1 a 60 del registro, el *Título* los caracteres del 61 al 180, etc. Nos referimos a la información misma a través de los nombres o las posiciones de los *campos*. Empezamos a hablar de los elementos de la estructura a través de *campos* y *contenidos de los campos*. Por ejemplo, el campo de Título (o los caracteres del 61 al 180 del registro) podría tener el contenido *Escritos sobre la historia del México antiguo*.

El uso de un formato fijo requiere que se hayan preestablecido todos los campos posibles en que se deseen subdividir o discriminar las unidades de información; en los casos de campos que puedan repetirse, deberá preestablecerse un máximo de repetición; también deberán imponerse longitudes máximas y en el caso de que un campo particular excediera al máximo esta-

blecido, dicho contenido tendría que abreviarse.

Los procesos de recuperación basados en formato fijo son los más eficientes, ya que se buscan los contenidos que se quieren seleccionar solamente en los campos deseados, y las manipulaciones para recuperación con condiciones lógicas son simples de desarrollar.

Este tipo de formato para captación y almacenamiento de información es satisfactorio cuando todas las unidades de información tienen contenidos para todos los campos preestablecidos en la estructura, y las longitudes de los contenidos para cada campo dado son similares en todas las unidades de información.

Sin embargo, cuando la información que se intenta manejar es muy variable en longitudes, posibilidades de repetición o ausencia de determinados campos, el desperdicio de espacio, así como el tamaño que la base puede llegar a tener produce un costoso manejo de los datos.

En el formato variable la identificación de los diversos elementos que se haya decidido discriminar se hace a través de la asociación de *etiquetas* a cada campo. Éstas pueden ser un nombre o una etiqueta numérica. Por ejemplo, el concepto *Título* o la etiqueta 245 de MARC podría ser la etiqueta asociada a cada ocurrencia de un título en una unidad de información. Hablamos entonces de *etiquetas* y *contenidos* para un campo específico. En el caso más simple, la información en formato variable está constituida por ocurrencias de parejas etiqueta-contenido. Tanto aquéllas como éstos pueden ser configuraciones similares de caracteres y además de longitudes variables, lo que hace necesario el uso de un carácter separador como “)” que permita la diferenciación entre las etiquetas y los contenidos. La información se almacena en la computadora como una cadena de etiquetas y contenidos separadas por el carácter especial.

Los procesos de recuperación de información almacenada en formato variable requieren del análisis de toda la información, de ahí la importancia de implantar algoritmos muy rápidos para dicho análisis.

Una vez elegido MARC como el esquema estructural que se impondrá a la información catalográfica del COAA y tomando en cuenta la característica de variabilidad de dicha información, se decidió que la manera más apropiada de almacenarla sería en formato variable.

Presentamos a continuación algunas unidades de información con el propósito de ejemplificar el grado de variabilidad real de la información catalográfica del COAA:

035> bb \$a 001707>
FCAP> 83/08/11> F1> 1969> PAIS> ne> ILUS> a> IDIOMA> eng>
040> bb \$a CM \$c CM>
092> bb \$a 221.063 \$b In618/1968>
111> 20 \$a International Organization of Old Testament Scholars \$b
Congress \$c Roma \$d 1968>
245> 10 \$a Congress volume \$b Rome, 1968>
260> 0b \$a Leiden, Netherlands \$b E. J. Brill \$c 1969>
300> bb \$a 244 p. \$b ilus. \$c 25 cm.>
440> b0 \$a Supplements to Vetus Testamentum \$v 17>
690> b0 \$a Libros sagrados>
690> b0 \$a Sagradas escrituras>
691> b0 \$a Asia \$x Religión>
695> 0b \$a Biblia \$p A. T. \$x Congresos>
695> 0b \$a Biblia \$p A. T. \$x Discursos, ensayos, conferencias>
901> bb \$a 137905>
FIN>

035> bb \$a 001717>
FCAP> 83/08/15> F1> 1957> PAIS> ne> IDIOMA> eng>
040> bb \$a CM \$c CM>
041> 1b \$a eng>
092> bb \$a 221.4 \$b M294m>
130> 00 \$a Manual de disciplina \$l Inglés>
245> 14 \$a The manual of discipline \$c Translated and annotated with
an introduction by P. Wernberg-Moller>
260> 0b \$a Leiden, Netherlands \$b E. J. Brill \$c 1957>
300> bb \$a 57 p. \$c 25 cm.>
440> b0 \$a Studies on the Texts of the Desert of Judah \$v 1>
691> b0 \$a Mar Muerto, Rollos>
690> b0 \$a Manuscritos judíos>
691> b0 \$a Jerusalem \$x Destrucción 70 D.C.>
691> b0 \$a Asia \$x Religión>
695> 0b \$a Biblia \$x Arqueología>
901> bb \$a 138480>
FIN>

035> bb \$a 001943>
 FCAP> 83/08/26> F1> 1974> PAIS> us> CONT> b> IDIOMA> eng>
 040> bb \$a CM \$c CM>
 092> bb \$a 016.95204 \$b W262a>
 100> 10 \$a Ward, Robert Edward>
 245> 14 \$a The allied occupation of Japan, 1945-1952 \$b an annotated
 bibliography of Western-language materials \$c Compiled and
 edited for the Joint Committee on Japanese Studies of the
 Social Science Research Council-American Council of Learned
 Societies and the Center for Japanese Studies of the University
 of Michigan by. . . and Frank Joseph Shulman. With the
 assistance of Masashi Nishihara and Mary Tobin Espey>
 260> 0b \$a Chicago, Ill. \$b American Library Association \$c 1974>
 300> bb \$a xx, 867 p. \$c 25 cm.>
 691> b0 \$a Japón \$x Historia \$y Ocupación aliada, 1945-1952 \$x
 Bibliografía>
 691> b0 \$a Asia \$x Historia \$y 1945-1952 \$x Bibliografía>
 700> 10 \$a Shulman, Frank Joseph \$d 1943— \$e coed.>
 901> bb \$a 120474/121029>
 FIN>

035> bb \$a 001957>
 FCAP> 83/08/26> F1> 1917> PAIS> us> CONT> h> IDIOMA> eng>
 040> bb \$a CM \$c CM>
 092> bb \$a 221.48 \$b S215o>
 100> 10 \$a Sanders, Henry Arthur \$d 1868->
 245> 14 \$a The Old Testament manuscripts in the Freer collection \$c by
 Henry A. Sanders. . .>
 260> 0b \$a New York \$b The Macmillan company \$c 1917>
 300> bb \$a 357 p. \$b facsim. \$c 28 cm.>
 440> b0 \$a University of Michigan Studies Humanistic series \$v 8>
 505> 0b \$a I. The Washington manuscript of Deuteronomy and
 Joshua.— II. The Washington manuscript of the Psalms>
 691> b0 \$a Asia \$x Religión>
 695> 0b \$a Biblia \$h Manuscritos griegos \$p A.T. \$k Selecciones>
 690> b0 \$a Manuscritos griegos \$x Facsímiles>
 730> 00 \$a Biblia \$p A.T. \$t Salmos \$l Griego \$f 1917>
 710> 20 \$a Freer gallery of art \$c Washington, D.C. Mss. (greek)>
 901> bb \$a 016884>
 FIN>

Al producir una unidad de información los catalogadores seleccionan únicamente los campos pertinentes a la obra en tanto que no consideran las etiquetas no aplicables. Compárense, por ejemplo, las unidades primera y cuarta; la primera incluye el campo de Ilustración (etiqueta ILUS) que no aparece en la cuarta, mientras que la cuarta incluye una nota de contenido

(etiqueta 505) y asientos secundarios (etiquetas 710 y 730) que no aparecen en la primera.

Por otra parte, el nivel de detalle en la discriminación de las partes de la información al que obliga MARC establece muchas posibilidades de elección dentro de un grupo de etiquetas dado; por ejemplo, para el grupo de etiquetas de asiento principal (etiquetas 100, 110, 111 y 130) el catalogador deberá seleccionar una de ellas de acuerdo con el tipo de elemento que fungirá como asiento principal de la unidad catalogada; compárense el autor personal (etiqueta 100) del tercer ejemplo con la conferencia o reunión (etiqueta 111) del primero y con el título uniforme (etiqueta 130) del segundo.

Otro factor involucrado en la variabilidad es que la longitud del contenido de un campo dado varía considerablemente de una unidad de información a otra; compárense, por ejemplo, los 35 caracteres del título (etiqueta 245) de la primera unidad con los 406 caracteres correspondientes al título de la tercera unidad. Queremos hacer notar que en casi todos los casos, como en los ejemplos aquí mencionados, la variabilidad en la longitud está relacionada con el nivel de detalle en la catalogación que es permitido por MARC. La existencia de subcampos da al catalogador la posibilidad de incorporar más información de manera estructurada en una unidad determinada. Por ejemplo, en la primera unidad únicamente se tiene información sobre el título propiamente dicho (subcampo \$a de la etiqueta 245) y sobre el subtítulo (subcampo \$b de la etiqueta 245), mientras que en la tercera unidad además de la información sobre título y subtítulo se tiene la mención de responsabilidad (subcampo \$c de la etiqueta 245).

Por último, MARC permite la repetición de algunas etiquetas o grupos de etiquetas, en particular, los encabezamientos de materia (etiquetas 690, 691, 692, 693, 694 y 695). Es importante considerar el hecho de que mientras más encabezamientos se asignen a una unidad determinada, habrá más puntos de acceso para la recuperación de dicha unidad. Sin embargo, habrá unidades de información que, por su tipo, no llevan encabezamientos de materia, por ejemplo, las unidades correspondientes a obras literarias.

Aunando todos los factores de variabilidad que hemos mencionado en los cuatro párrafos anteriores, podemos concluir que el grado de variabilidad de la información catalogada bajo el esquema estructural de MARC es muy alto. Si intentáramos abordar el problema de análisis de esta información bajo un esquema en formato fijo, tendríamos que preestablecer máximos de longitud para el contenido de las 408 partes en que se puede subdividir una unidad de información —etiquetas y subcampos del MARC—, así como máximos de repetición para los campos repetibles, lo cual nos daría una longitud del orden de 10 000 caracteres para una unidad de información. Simplemente para almacenar las cuatro unidades de información que nos han servido de ejemplo bajo un esquema en formato fijo, tendríamos que reservar 1 316 caracteres para cada una; mientras que almacenadas en formato variable la ficha de longitud mínima ocupa 589 caracteres y la de longitud máxima ocupa 857; por lo tanto, la cantidad de caracteres usados en formato fijo comparada con la cantidad de caracteres usados en formato

variable por la ficha promedio guardan una razón de dos a uno. Hacemos notar que las cuatro fichas con las que ejemplificamos no forman una muestra en la que estén representadas las posibilidades de variación que permite MARC y que entre más fichas se consideren la razón de uso de espacio en memoria aumentará.

IV. EL SISTEMA DE RECUPERACIÓN BIBLIOGRÁFICA

En este capítulo describiremos los métodos computacionales mediante los cuales efectuamos los procesos de recuperación bibliográfica. Para realizar los procesos sobre información almacenada en formato variable se requiere localizar rápidamente las ocurrencias de determinadas cadenas de caracteres en un texto para, posteriormente, realizar las operaciones de selección y transformación de la información.

Se verá que el sistema que efectúa los procesos de recuperación es un autómata finito que reconoce el lenguaje conformado por las etiquetas y códigos de subcampos de MARC. Durante el análisis léxico se efectúa el rastreo y la identificación de las etiquetas; al llegar a un estado terminal se toma la decisión de cómo manipular el contenido correspondiente a la etiqueta identificada, o sea, cuáles operaciones de selección o transformación hay que aplicar al contenido del campo.

A continuación presentamos los principales conceptos matemáticos que nos permitirán dar una clara descripción de los algoritmos, estructuras y métodos computacionales que conforman este trabajo.

Un *símbolo* es una entidad abstracta que no se define formalmente pero que tiene un significado intuitivo suficientemente claro. Las letras, los dígitos y, en general, todos los caracteres son ejemplos de símbolos usados frecuentemente.

Una *cadena o palabra* es una secuencia finita de símbolos yuxtapuestos. La *longitud de una cadena* w se denota por $|w|$ y es el número de símbolos que componen la cadena. La *cadena vacía* se denota por ϵ y es la cadena que consiste de cero símbolos.

Un *alfabeto* es un conjunto finito de símbolos. Un *lenguaje formal* sobre un alfabeto es un conjunto de cadenas de símbolos de dicho alfabeto.

Una *gráfica*, denotada $G = (V, E)$, consiste de un conjunto finito V de vértices o nodos y un conjunto E de parejas de vértices llamadas lados.

Una *trayectoria* en una gráfica es una secuencia de vértices v_1, v_2, \dots, v_k con $k \geq 1$, tal que hay un lado (v_i, v_{i+1}) para todo i con $1 \leq i < k$. La longitud de la trayectoria es $k-1$.

Una *gráfica dirigida* o una *digráfica*, que se denota $D = (V, E)$, consiste de un conjunto V de vértices y un conjunto E de parejas ordenadas de vértices llamadas arcos. Si $v, w \in V$ denotamos un arco de v a w por $v \rightarrow w$.

Un *recorrido* en una digráfica es una secuencia de vértices v_1, v_2, \dots, v_k con $k \geq 1$, tal que $v_i \rightarrow v_{i+1}$ es un arco para cada i , con $1 \leq i < k$. Se dice que el recorrido es de v_1 a v_k .

Si $v \rightarrow w$ es un arco se dice que v es un *predecesor* de w y que w es un *sucesor* de v .

Un *árbol dirigido y ordenado* es una digráfica con las siguientes propiedades:

1) Hay un vértice, llamado *raíz*, que no tiene predecesores y desde el cual hay un recorrido hacia cualquier vértice.

2) Cualquier vértice que no sea la raíz tiene exactamente un predecesor.

3) Los sucesores de cada vértice están ordenados de izquierda a derecha.

En la literatura especializada se utiliza una terminología especial para los árboles que difiere de la terminología general para gráficas: el sucesor de un vértice se llama *hijo* y el predecesor es llamado *padre*; a un vértice que no tiene hijos se le llama *hoja* y a los otros vértices se les llama *vértices interiores*.

Un *autómata finito* es un modelo matemático de un sistema con entradas y salidas discretas. El sistema puede estar en cualquiera de un número finito de configuraciones o "estados". El estado del sistema resume la información concerniente a las entradas pasadas que es necesaria para determinar el funcionamiento del sistema en las subsecuentes entradas. En computación se encuentran muchos ejemplos de sistemas de estados finitos y la teoría de autómatas finitos es una herramienta de diseño muy útil para estos sistemas.

Un *autómata finito (AF)* consiste de un conjunto finito de estados y un conjunto finito de transiciones de estado a estado que ocurren con símbolos de entrada tomados de un alfabeto Σ . Para cada símbolo de entrada hay exactamente una transición hacia afuera de cada estado (posiblemente hacia sí mismo). Un estado usualmente denotado q_0 , es el estado inicial, en el cual empieza el autómata. Algunos estados se designan como estados finales.

Se asocia una digráfica, llamada *diagrama de transiciones*, a cada AF de la siguiente forma: los vértices de la gráfica corresponden a los estados del AF, si hay una transición del estado q al estado p con una entrada a , entonces hay un arco etiquetado con a del vértice o estado q al p en el diagrama de transición. Un AF *acepta* una cadena x si la secuencia de transiciones correspondientes a los símbolos de x lleva del estado inicial a algún estado final.

Formalmente se denota un *autómata finito* por un quinteto $(Q, \Sigma, \delta, q_0, F)$ donde Q es un conjunto finito de estados, Σ es un alfabeto finito de entrada, $q_0 \in Q$ es el estado inicial, $F \subset Q$ es el conjunto de estados finales, y δ es una función de transición $\delta: Q \times \Sigma \rightarrow Q$. Esto es $\delta(q, a)$ es un estado para cada estado $q \in Q$ y símbolo de entrada $a \in \Sigma$.

Una vez dadas las definiciones matemáticas fundamentales, entraremos de lleno a la explicación formal de los métodos computacionales mediante los que se efectúan los procesos de recuperación bibliográfica.

Recordará el lector que la información catalográfica con la que trabajamos ha sido estructurada con base en el esquema de MARC, y se almacena en formato variable como ocurrencias de parejas etiqueta-contenido. Los procedimientos de recuperación bibliográfica constan de dos partes: la pri-

mera, a la que llamamos análisis léxico, es la que efectúa la localización dentro de la información del COAA de las etiquetas MARC que identifican los campos de la estructura; y la segunda, a la que llamaremos procesamiento semántico, que es la aplicación de operaciones de selección y transformación a la información que constituye el contenido de la etiqueta localizada durante el análisis léxico.

La familia de programas mediante los cuales se efectúan los diversos procesos de recuperación bibliográfica, son implantaciones computacionales eficientes de autómatas finitos que realizan el análisis léxico, aunadas a subrutinas o procedimientos en las que se realiza el procesamiento semántico.

El autómata finito que realiza el reconocimiento de la clase restringida de expresiones regulares constituida por el lenguaje de MARC es construido automáticamente a partir de una descripción en la que cada etiqueta o código de subcampo del lenguaje MARC está asociada al operador semántico que se desea aplicar al contenido de esa etiqueta durante el momento del procesamiento de la información.

El análisis léxico está controlado por un autómata finito cuyo diagrama de transiciones es un árbol que representa a las etiquetas de la manera que se describe a continuación. Para que esta descripción sea clara, iremos construyendo el árbol correspondiente a algunas etiquetas MARC sin considerar los códigos de subcampos. Cada etiqueta se representa como una secuencia de caracteres alfanuméricos terminada en el carácter separador. La lista de dichas secuencias se debe ordenar alfabéticamente para que pueda construirse adecuadamente la tabla que representará al árbol.

020>	240>	505>	740>	FIN>
035>	241>	507>	800>	FSPUB>
040>	245>	520>	810>	IDIOMA>
041>	250>	533>	811>	ILUS>
043>	260>	546>	830>	IND>
049>	300>	590>	840>	LEST>
050>	400>	690>	901>	LTIP>
080>	410>	691>	902>	LNBIB>
082>	411>	692>	AP>	LNDES>
092>	440>	693>	BIO>	LFOR>
098>	490>	694>	CONF>	PAIS>
099>	500>	695>	CONT>	PUBOF>
100>	501>	700>	EDCONM>	REPROD>
110>	502>	710>	F1>	
111>	503>	711>	F2>	
130>	504>	730>	FCAP>	

Figura 1. Lista de etiquetas MARC.

La raíz del árbol tiene tantos hijos como caracteres diferentes puede ha-

ber en el lugar de más a la izquierda de cada etiqueta MARC. Los arcos que unen a la raíz con cada uno de sus hijos estarán etiquetados con dichos caracteres. En la figura 2 el arco con la marca 1 del primer nivel corresponde tanto a la etiqueta 100> como a la 110>, la 111> y la 130>. De la misma manera, a los arcos con las marcas 2, 3, etcétera, les corresponden sus respectivas listas de etiquetas MARC.

Recursivamente, tomando todas las etiquetas MARC que tuvieron el mismo carácter en el lugar de más a la izquierda, construimos subárboles en cada nodo del primer nivel y obtendremos un segundo nivel en el árbol.

Siguiendo este proceso llegará el momento en que todas las etiquetas MARC estén representadas en el árbol mediante un recorrido desde la raíz hasta una hoja. El carácter separador ">" se considera como el último carácter de cada etiqueta MARC, de manera que todos los arcos etiquetados con ese carácter llegan a una hoja de donde se hace depender un *nodo agregado* en donde está codificado un operador semántico.

El árbol y los nodos agregados se implantan como una tabla que es generada automáticamente por el programa *Arbol* incluido en el anexo A. Los estados del autómata, es decir, los nodos del árbol, junto con los nodos agregados se numeran por niveles desde la raíz hacia las hojas de izquierda a derecha, de manera que todos los hijos de un nodo determinado están numerados consecutivamente. Esta numeración se corresponde con el número del registro de la tabla. Existen, por lo tanto, dos tipos de registros: los que corresponden a los estados del autómata y los correspondientes a los nodos agregados; el primer carácter de cada registro marca la diferencia entre un estado del autómata y un nodo agregado. El carácter "*" marca los estados del autómata, mientras que cualquier otro carácter se podrá utilizar como el código del operador semántico.

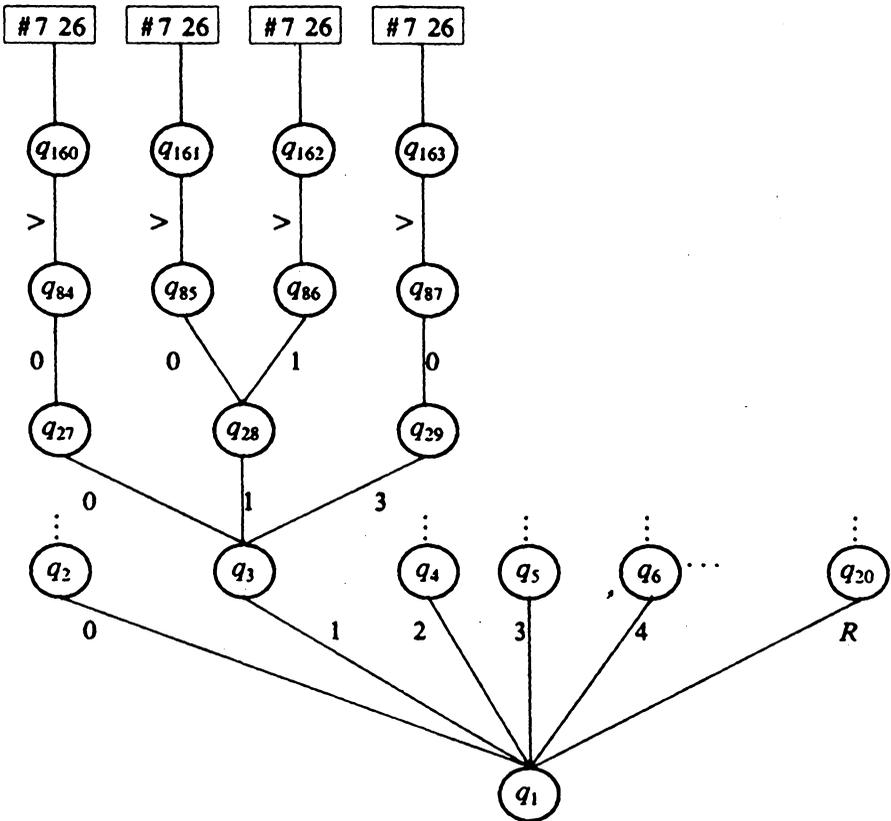


Figura 2. Diagrama de transiciones del autómata.

1: * <	19	2	44: * 0	1	116	87: * 0	1	163
2: * 0	6	21	45: * 1	2	117	88: * 0	1	164
3: * 1	3	27	46: * 3	1	119	89: * 1	1	165
4: * 2	3	30	47: * 4	1	120	90: * 5	1	166
5: * 3	1	33	48: * 0	1	121	91: * 0	1	167
6: * 4	4	34	49: * 1	2	122	92: * 0	1	168
7: * 5	5	38	50: * 3	1	124	93: * 0	1	169
8: * 6	1	43	51: * 4	1	125	94: * 0	1	170
9: * 7	4	44	52: * 0	2	126	95: * 0	1	171
10: * 8	4	48	53: * P	1	128	96: * 1	1	172
11: * 9	1	52	54: * I	1	129	97: * 0	1	173
12: * A	1	53	55: * O	1	130	98: * 0	1	174
13: * B	1	54	56: * D	1	131	99: * 0	1	175
14: * C	1	55	57: * 1	1	132	100: * 1	1	176
15: * E	1	56	58: * 2	1	133	101: * 2	1	177
16: * F	5	57	59: * C	1	134	102: * 3	1	178
17: * I	3	62	60: * I	1	135	103: * 4	1	179
18: * L	4	65	61: * S	1	136	104: * 5	1	180
19: * P	2	69	62: * D	1	137	105: * 7	1	181
20: * R	1	71	63: * L	1	138	106: * 0	1	182
21: * 2	1	72	64: * N	1	139	107: * 3	1	183
22: * 3	1	73	65: * E	1	140	108: * 6	1	184
23: * 4	4	74	66: * T	1	141	109: * 0	1	185
24: * 5	1	78	67: * N	2	142	110: * 0	1	186
25: * 8	2	79	68: * F	1	144	111: * 1	1	187
26: * 9	3	81	69: * A	1	145	112: * 2	1	188
27: * 0	1	84	70: * U	1	146	113: * 3	1	189
28: * 1	2	85	71: * E	1	147	114: * 4	1	190
29: * 3	1	87	72: * 0	1	148	115: * 5	1	191
30: * 4	3	88	73: * 5	1	149	116: * 0	1	192
31: * 5	1	91	74: * 0	1	150	117: * 0	1	193
32: * 6	1	92	75: * 1	1	151	118: * 1	1	194
33: * 0	1	93	76: * 3	1	152	119: * 0	1	195
34: * 0	1	94	77: * 9	1	153	120: * 0	1	196
35: * 1	2	95	78: * 0	1	154	121: * 0	1	197
36: * 4	1	97	79: * 0	1	155	122: * 0	1	198
37: * 9	1	98	80: * 2	1	156	123: * 1	1	199
38: * 0	7	99	81: * 2	1	157	124: * 0	1	200
39: * 2	1	106	82: * 8	1	158	125: * 0	1	201
40: * 3	1	107	83: * 9	1	159	126: * 1	1	202
41: * 4	1	108	84: * 0	1	160	127: * 2	1	203
42: * 9	1	109	85: * 0	1	161	128: * >	1	204
43: * 9	6	110	86: * 1	1	162	129: * O	1	205

Figura 3. Tabla en la que se representa el autómata.

130: * N	2 206	173: * >	1 250	216: * >	1 290
131: * C	1 208	174: * >	1 251	217: * T	1 291
132: * >	1 209	175: * >	1 252	218: * P	1 292
133: * >	1 210	176: * >	1 253	219: * I	1 293
134: * A	1 211	177: * >	1 254	220: * E	1 294
135: * N	1 212	178: * >	1 255	221: * R	1 295
136: * P	1 213	179: * >	1 256	222: * S	1 296
137: * I	1 214	180: * >	1 257	223: * O	1 297
138: * U	1 215	181: * >	1 258	224: * R	1 298
139: * D	1 216	182: * >	1 259	225: 0	
140: * S	1 217	183: * >	1 260	226: #	0 6
141: * I	1 218	184: * >	1 261	227: 0	
142: * B	1 219	185: * >	1 262	228: 0	
143: * D	1 220	186: * >	1 263	229: 0	
144: * O	1 221	187: * >	1 264	230: 0	
145: * I	1 222	188: * >	1 265	231: 0	
146: * B	1 223	189: * >	1 266	232: 0	
147: * P	1 224	190: * >	1 267	233: 0	
148: * >	1 225	191: * >	1 268	234: 0	
149: * >	1 226	192: * >	1 269	235: 0	
150: * >	1 227	193: * >	1 270	236: 0	
151: * >	1 228	194: * >	1 271	237: #	7 26
152: * >	1 229	195: * >	1 272	238: #	7 26
153: * >	1 230	196: * >	1 273	239: #	7 26
154: * >	1 231	197: * >	1 274	240: #	7 26
155: * >	1 232	198: * >	1 275	241: #	34 29
156: * >	1 233	199: * >	1 276	242: #	34 29
157: * >	1 234	200: * >	1 277	243: #	34 29
158: * >	1 235	201: * >	1 278	244: 0	
159: * >	1 236	202: * >	1 279	245: 0	
160: * >	1 237	203: * >	1 280	246: 0	
161: * >	1 238	204: 0		247: 0	
162: * >	1 239	205: * >	1 281	248: 0	
163: * >	1 240	206: * F	1 282	249: 0	
164: * >	1 241	207: * T	1 283	250: 0	
165: * >	1 242	208: * O	1 284	251: 0	
166: * >	1 243	209: 0		252: 0	
167: * >	1 244	210: 0		253: 0	
168: * >	1 245	211: * P	1 285	254: 0	
169: * >	1 246	212: * >	1 286	255: 0	
170: * >	1 247	213: * U	1 287	256: 0	
171: * >	1 248	214: * O	1 288	257: 0	
172: * >	1 249	215: * S	1 289	258: 0	

Figura 3. (Continuación)

259: 0	281: 0	303: * > 1 315
260: 0	282: * > 1 299	304: * A 1 316
261: 0	283: * > 1 300	305: 0
262: 0	284: * N 1 301	306: 0
263: 0	285: * > 1 302	307: 0
264: 0	286: \$	308: * > 1 317
265: 0	287: * B 1 303	309: * > 1 318
266: 0	288: * M 1 304	310: 0
267: 0	289: * > 1 305	311: 0
268: 0	290: 0	312: * > 1 319
269: 0	291: * > 1 306	313: * D 1 320
270: 0	292: * > 1 307	314: * > 1 321
271: 0	293: * B 1 308	315: 0
272: 0	294: * S 1 309	316: * > 1 322
273: 0	295: * > 1 310	317: 0
274: 0	296: * > 1 311	318: 0
275: 0	297: * F 1 312	319: 0
276: 0	298: * O 1 313	320: * > 1 323
277: 0	299: 0	321: 0
278: 0	300: 0	322: 0
279: 0	301: * M 1 314	323: 0
280: 0	302: 0	

Figura 3. (Conclusión)

En los registros correspondientes a los estados del autómata está codificada la información necesaria para obtener la función de transición del autómata; cada uno de estos registros está formado por cuatro campos: en el primero, al que denotaremos como "tipo_nodo", está el *; en el segundo, denotado "etiqueta_arco", se almacena el carácter que debe ser comparado con el carácter actual de la cadena de entrada para decidir si se realiza o no la transición al estado correspondiente a este registro; en el tercero, "hijos", se guarda el número de hijos que tiene el nodo correspondiente al registro; y, en el cuarto, "primer_hijo", está el apuntador al registro a partir del cual están secuencialmente almacenados los hijos del nodo en cuestión. En cambio, cuando el registro corresponde a un nodo agregado en él estará codificada la información necesaria para una adecuada aplicación de la acción semántica. La figura 3 es la tabla que representa al autómata para el reconocimiento de las etiquetas MARC sin códigos de subcampos. En ella se han codificado, a modo de ejemplo, tres operadores semánticos con sus parámetros ("0", "#" y "\$").

En la figura 4 se muestra el algoritmo que reproduce la función de transición del autómata a partir de la información almacenada en la tabla. El estado en el que se encuentra el autómata se representa mediante un apuntador "estado_actual" a los registros de la tabla. El primer registro de la tabla corresponde al estado inicial del autómata que es donde se inicia el análisis.

lisis para cada etiqueta MARC. Para decidir a qué estado debe hacerse la transición para el siguiente carácter de la cadena de entrada, es necesario revisar, de acuerdo con el orden secuencial preestablecido, las etiquetas de los arcos que llevan del estado actual a cada uno de sus hijos. Estas etiquetas están almacenadas precisamente en los registros correspondientes a dichos hijos, por lo que la variable “estado__actual” está siempre un nivel hacia adelante del estado actual real del autómata. En el momento en que el carácter de la cadena de entrada coincide con una de las etiquetas de los hijos del estado actual, se realiza la transición a ese hijo, la cual consiste en la decodificación del número de hijos del estado actual en la variable “hijo”, la cual controlará el número de revisiones de las etiquetas. Éste es el único momento en el que la variable “estado__actual” apunta exactamente al estado actual real del autómata. Inmediatamente después de esta transición se vuelve a “ver hacia adelante” con la variable “estado__actual”. Ocasionalmente, después de haber revisado todas las etiquetas de los arcos que llevan de un estado a sus hijos —es decir, la variable “hijo” ha sido decremen-tada hasta cero— no habrá coincidido el carácter de entrada con ninguna de ellas, lo cual significará haber detectado un error en la información.

```

error ← FALSO
estado__actual ← 1
hijo ← tabla[estado__actual].hijos
c ← obtener carácter de entrada
estado__actual ← tabla[estado__actual].primer__hijo
MIENTRAS tabla[estado__actual].tipo__nodo = estado del autómata
    Y NO error( ) EFECTUA
{
    SI c = tabla[estado__actual].etiqueta__arco ENTONCES
    {
        hijo ← tabla[estado__actual].hijos
        c ← obtener carácter de entrada
        estado__actual ← tabla[estado__actual].primer__hijo
    }
    SI NO
    {
        estado__actual ← estado__actual + 1
        hijo ← hijo - 1
        SI hijo = 0 ENTONCES error ← CIERTO
    }
}

```

Figura 4. Algoritmo que reproduce la función de transición del autómata.

El hecho de que la variable “estado__actual” esté siempre “viendo hacia adelante” permite que se identifique que el autómata está en un estado

final porque el tipo de nodo al que apunta dicha variable tiene el código de un operador semántico. Cuando el autómata llega a un estado final, es decir, acepta una etiqueta MARC, se aplica el operador semántico —cuyo código está en el nodo agregado a ese estado final— a la información de entrada que es en realidad el contenido de la etiqueta identificada.

En la tabla de la figura 3 en la que se representa el autómata y sus nodos agregados fueron codificados los operadores semánticos necesarios para construir un subconjunto de la información catalográfica en formato fijo, que incluya el número de control Colmex (etiqueta 035), el autor (etiquetas 100, 110, 111 y 130) y el título (etiquetas 240, 241 y 245).

Para dejar claro cómo se usa la tabla para el reconocimiento e identificación de etiquetas MARC, así como para tomar la decisión de cuál operador semántico utilizar, a continuación se presentan los registros de la tabla que se van seleccionando durante el procesamiento del siguiente ejemplo:

...245) The allied occupation of Japan) ...

<i>estado__actual</i>	<i>c</i>	<i>Resto de la cadena de entrada</i>
1: * < 19 2 2		..45) The allied occupation of Japan) ...
2: * 0 6 21 2		..45) The allied occupation of Japan) ...
3: * 1 3 27 2		..45) The allied occupation of Japan) ...
4: * 2 3 30 2		..45) The allied occupation of Japan) ...
30: * 4 3 88 4		...5) The allied occupation of Japan) ...
88: * 0 1 164 5	) The allied occupation of Japan) ...
89: * 1 1 165 5	) The allied occupation of Japan) ...
90: * 5 1 166 5	) The allied occupation of Japan) ...
166: *) 1 243)	 The allied occupation of Japan) ...
243: # 34 29 T	he allied occupation of Japan) ...

Al alcanzar el nodo agregado 243 se inicia la aplicación del operador semántico “#” de escritura en formato fijo, y el resultado de esto es poner la información del título “The allied occupation of Japan” en los caracteres del 34 al 62 del registro de salida.

El reconocimiento del carácter separador “)” después del título produce que se reinicie la operación del autómata.

Por el orden en que la información entra al proceso de análisis, ya debió haberse hecho, para la unidad de información que nos sirve de ejemplo, el reconocimiento y la transformación de los campos de número de control Colmex y de autor, lo cual ha producido que en el registro de salida ya esté escrita adecuadamente dicha información.

A continuación se presenta el resultado del procesamiento dirigido por la tabla de la figura 3 para los cuatro ejemplos de unidades de información del capítulo anterior:

001707 International Organization	Congress volume
001717 Manual de disciplina	The manual of discipline
001943 Ward, Robert Edward	The allied occupation of Japan
001957 Sanders, Henry Arthur	The Old Testament manuscripts

Los operadores semánticos son pequeños procedimientos, subrutinas o secciones del programa de análisis mediante las que se efectúa la transformación o selección de la información. Estos operadores deben ser programados de acuerdo al objetivo particular del sistema.

En el ejemplo presentado se codificaron tres tipos de operadores semánticos:

1) El operador semántico de escritura en formato fijo “#” con sus dos parámetros “carácter inicial” y “longitud máxima” permite seleccionar los campos que se incluirán en el subconjunto, y, al mismo tiempo, copiar el contenido del campo identificado al registro de salida desde la posición indicada por el primer parámetro; en caso de que la longitud del contenido exceda al valor del segundo parámetro, el resto del contenido será omitido.

2) El operador semántico de omisión “0” permite la exclusión de los campos que no se desea incluir en el subconjunto.

3) El operador de fin de ficha “\$” hace las reinicializaciones apropiadas.

Una vez construida la tabla que representa al autómata que reconoce el lenguaje MARC, es muy simple codificar, en los nodos agregados, los operadores semánticos seleccionados, para las etiquetas que interese. La posibilidad de cambiar los operadores seleccionados, o asignarlos a otras etiquetas, produce, de hecho, que un mismo programa se comporte de formas variadas.

Como se dijo en su momento, la tabla de la figura 3 fue construida sin incluir los códigos de subcampo de MARC, con el propósito de dar claridad a la explicación sobre la construcción del autómata y el análisis léxico. Cuando se quiere analizar la información catalogada bajo el esquema estructural de MARC considerando en forma discriminada los subcampos de cada etiqueta, se construye para cada etiqueta —adicionalmente al autómata de reconocimiento de las etiquetas MARC descrito anteriormente— un autómata de reconocimiento de los códigos de subcampos que corresponden a dicha etiqueta. Cada autómata adicional se construye con la metodología descrita para el autómata principal. Su tabla correspondiente se incluye en la tabla principal ligada a través de un *registro de conexión* que está en lugar del nodo agregado al nodo final de reconocimiento de la etiqueta dada. En cada uno de estos registros se codifica el apuntador que lleva al estado inicial del autómata correspondiente. En los registros de conexión se pueden codificar además los parámetros que sean comunes a todo el conjunto de operadores semánticos asociados a los códigos de subcampo de la etiqueta dada. El análisis de los subcampos se efectúa usando el autómata adicional correspondiente —con el mismo algoritmo de la figura 4— hasta encontrar el separador que marca el fin del contenido de la etiqueta, con lo que se regresa al autómata principal para seguir el análisis.

El autómata principal y los autómatas adicionales se implantan con estructuras similares y se utilizan con el mismo algoritmo, de aquí que sea natural representarlos en una misma tabla utilizando los registros de conexión. Usaremos el término *autómatas conectados* para referirnos a los autómatas que se usan en conjunto con la modalidad de representación descrita.

Con los operadores semánticos de exclusión de campos y de transformación a formato fijo que acabamos de explicar puede crearse la estructura de archivos con la información catalográfica que proporcione los mismos resultados que dan los catálogos por autor y por título usados normalmente en una biblioteca. Describiremos brevemente a continuación, también a manera de ejemplo, la forma en que se puede crear dicha organización de información. Usando los operadores semánticos exactamente como lo describimos en el ejemplo sobre el análisis léxico, se construye un archivo en formato fijo que contiene en cada registro la información del campo de autor, la del campo de título, y además el número de control Colmex para cada unidad de información. Este archivo se carga en cualquier manejador de bases de datos. Por otro lado, se crea un archivo cuyos registros guardan el número de control Colmex para cada unidad de información y los apuntadores al inicio y fin de dicha unidad de información completa en formato variable. Este archivo es creado por supuesto con un programa que utiliza la misma metodología de análisis léxico. Mediante las operaciones del manejador de bases de datos se realizan las selecciones de los registros que cumplan las condiciones impuestas por el usuario a la información de autor y de título. También, mediante operaciones del manejador, se obtienen los apuntadores al inicio y fin de los registros seleccionados, y dichos apuntadores son descargados de la base de datos. Posteriormente, un programa muy sencillo toma los apuntadores de registros seleccionados y recupera, del archivo en formato variable, la información completa de las unidades seleccionadas.

En el anexo A se incluyen el programa que construye la tabla en la que se representan autómatas (*Arbol.c*), el programa que codifica el contenido de los registros de conexión y de los nodos agregados (*Semantic.c*), el programa que transforma a formato fijo subconjuntos de la información catalográfica (*Indices.c*), el programa que crea los apuntadores para la recuperación de las fichas en formato variable (*Creapun.c*), el programa que recupera dichas fichas (*Recupera.c*), así como algunas de las tablas que utilizan.

A continuación queremos discutir las similitudes que existen entre lo que hemos denominado autómatas conectados y operadores semánticos, y el modelo de redes de transición aumentadas (ATN). Para esto usaremos algunos de los conceptos y argumentos que presenta W. A. Woods en su artículo "Transition Network Grammars for Natural Language Analysis" (Woods, 1970).

Una red de transición recursiva es una gráfica dirigida con estados y arcos etiquetados. . . Esencialmente se parece a un diagrama de transiciones de un autómata finito no determinístico excepto porque las etiquetas de los arcos pueden

ser nombres de estados además de símbolos terminales.

Podríamos decir que los registros de conexión de nuestro formalismo corresponden a arcos etiquetados con el nombre de un estado.

La interpretación de un arco etiquetado con un nombre de estado es que el estado al que llega el arco será metido a una pila y el control se transferirá (sin avanzar la cadena de entrada) al estado que etiqueta al arco.

Efectivamente el apuntador al registro de conexión es conservado en una variable especial con el propósito de regresar a ese "estado" cada vez que se reinicia el reconocimiento de un subcampo para una etiqueta MARC específica.

Los nombres de estados que pueden aparecer en los arcos en este modelo son esencialmente nombres de construcciones que se pueden encontrar como "frases" de la cadena de entrada. El efecto de un arco etiquetado como estado es que la transición que representa puede llevarse a cabo si una construcción del tipo indicado se encuentra como "frase" en el punto apropiado de la cadena de entrada.

En nuestro caso las frases o constituyentes que debemos encontrar son los subcampos de cada etiqueta MARC.

Para el problema que discutimos en esta obra, es decir, para el reconocimiento y análisis de las fichas catalográficas en lenguaje MARC no es necesaria la aplicación recursiva de los autómatas principal o agregados debido a lo cual la pila queda reducida a una variable y la representación de la red de transición puede ser simplemente la tabla que hemos descrito.

Woods resalta algunas de las características de las ATN:

Las principales características que una gramática transformacional agrega a las gramáticas libres de contexto son la capacidad de mover, copiar o borrar fragmentos de la estructura de la oración, y efectuar estas acciones sobre constituyentes que generalmente dependen del contexto en el que ocurren. Nosotros podemos agregar facilidades equivalentes al modelo de redes de transición agregando a cada arco de la red . . . un conjunto de acciones para construir estructura que se ejecutarán si se sigue el arco. Llamamos a esta versión del modelo redes de transición aumentadas.

Las redes de transición aumentadas van construyendo una descripción estructural parcial de la oración conforme van de estado a estado a través de la red. Las piezas de esta descripción parcial son almacenadas en registros. . . Las acciones para construir estructuras especificadas en los arcos cambian el contenido de estos registros en función de contenidos previos, de contenidos de otros registros, el símbolo de entrada, y/o los resultados de operaciones de nivel inferior. . . las subestructuras eventualmente se incorporarán a estructuras mayores. . .

Podríamos intentar establecer el siguiente paralelismo: la cadena de en-

trada a nuestro proceso corresponde a la estructura superficial de una ficha MARC, y el producto que resultará del análisis y transformación de una estructura superficial mediante nuestros operadores semánticos corresponde a una especie de estructura profunda. Aquí los operadores semánticos corresponden al conjunto de acciones para construir estructura que se efectuarán durante el análisis cuando se realizan las transferencias entre nodos de la ATN.

En otro intento de establecer un paralelismo la ficha MARC sería una conjunción de estructuras superficial y profunda, donde los contenidos de los campos corresponden a la superficial y el conjunto ordenado de etiquetas y códigos de subcampos MARC que aparecen en una ficha dada son la estructura profunda. Aquí deberemos entender el procesamiento descrito en este libro como un proceso de traducción. En él se usa la tabla —que puede verse como ATN— para interpretar el lenguaje fuente, y nuestros operadores semánticos se utilizan para caracterizar al lenguaje objeto.

CONCLUSIONES

En este trabajo se presenta una solución al problema de recuperación bibliográfica desde un punto de vista descriptivo, debido a lo cual muchas de las afirmaciones están basadas en ejemplos particulares. Sin embargo, la solución descrita tiene intrínseca una metodología computacional para el manejo de información documental que incluye: la acomodación de un esquema estructural a los datos por analizar, el uso de un formato variable con identificadores y separadores de campos para almacenar la información, la aplicación del análisis léxico dirigido por una tabla en la que se representan autómatas conectados, y el diseño y programación de operadores semánticos.

El método de programación descrito permite obtener productos muy especializados de manera rápida, debido a que ya han sido programados tanto el algoritmo para construir la tabla que representa a los autómatas —a partir de simples descripciones de las etiquetas del esquema estructural—, como el algoritmo de utilización de esa tabla para efectuar el análisis léxico. Al abordar cada nueva aplicación de este método de programación, es necesario solamente programar los operadores semánticos. En este libro se han ejemplificado algunos de los operadores semánticos más simples, sin embargo, con operadores más complejos, este mismo método ha sido utilizado para elaborar algunos de los productos requeridos por la BDCV; por ejemplo, los juegos de tarjetas catalográficas con alta calidad de impresión junto con su correspondiente juego de etiquetas para el marcaje del libro; la lista de obras catalogadas que está formada por los datos de la referencia bibliográfica y en la cual los descriptores (particularmente encabezamientos de materia) se utilizan para conformar un índice a las referencias; bibliografías especializadas; y registros MARC propiamente dichos los cuales se han importado al paquete MICRO CDS/ISIS.

La interrelación entre el sistema descrito y algún manejador de bases de datos permite trabajar con un alto grado de eficiencia. Por un lado, se aprovechan las ventajas de la recuperación selectiva que ofrece el manejador de bases de datos. Y, por otro lado, una vez hecha la selección de un subconjunto de unidades de información, es posible aplicarle alguno de los programas para emisión de productos especializados como los mencionados en el párrafo anterior.

Las herramientas aquí descritas, tanto la metodología como la implementación misma del sistema, poseen un alto grado de adaptabilidad. Cuando un investigador especializado obtiene nueva información que complementa a la ya existente pero que no había sido prevista en su esquema estructural, basta que defina el esquema estructural aumentado, y que, utilizando los criterios

metodológicos descritos, agregue a las unidades de información pertinentes sus nuevos datos. A nivel del trabajo computacional definir el esquema estructural aumentado equivale a agregar las etiquetas faltantes a la descripción del esquema original, y aplicar los programas de construcción de la tabla en la que se representan los autómatas y de codificación de los operadores semánticos. Los programas ya existentes pueden ser usados sin modificación.

La posibilidad de aumentar información, sobre todo resúmenes en lengua natural —probablemente mal estructurados dentro de sí mismos, aunque bien estructurados como componentes de las unidades de información completas— y manipularla en conjunto mediante las herramientas computacionales descritas en este libro, permite al investigador encontrar las características que él representará mediante descriptores con objeto de dar a su información mejores formas de organización.

En el procedimiento de recuperación bibliográfica descrito en el último capítulo, los programas con los que se ejemplifica la metodología que ha sido objeto de esta obra están efectuando la función de una operación generalizada de proyección de un manejador de bases de datos relacional. Es por esto que el trabajo descrito en este libro puede considerarse como un método computacional complementario a los manejadores de bases de datos.

A N E X O S

ANEXO A

PROGRAMAS Y ALGUNAS DE LAS TABLAS QUE UTILIZAN

1. Arbol.c	49
2. Lista de descripciones para reconocimiento de MARC	54
3. Tabla en la que se representa a los autómatas conectados	57
4. Semantic.c	66
5. Indices.c	72
6. Creapun.c	80
7. Recupera.c	88

/*

Programa: Arbol.c**Objetivo:** Construir tablas que representan autómatas simples o conectados.

Datos de entrada: Una lista ordenada de descripciones formadas por las etiquetas, que reconocerá el o los autómatas que se van a representar, asociadas a códigos de operadores semánticos con sus parámetros. En el caso de autómatas simples, cada descripción está formada por la etiqueta, una coma, y el operador con sus parámetros; por ejemplo "FIN>,\$". En el caso de autómatas conectados habrá tantas descripciones para cada etiqueta como posibles cadenas se deseen reconocer mediante el autómata conectado al autómata principal. Los caracteres "<" que van a continuación de la etiqueta se utilizan para crear el registro de conexión entre los autómatas. Por ejemplo:

260)<\$a,#06000

260)<\$b,#02000

260)<\$c,#02000

Datos de salida: La tabla que representa al autómata simple o a los autómatas conectados.

El programa se ejecuta de la siguiente manera:

```
arbol.ex <arch-des> <arch-sal>
```

*/

```
#include <stdio.h>
#define LONDES 30 /* longitud máxima de una descripción */
#define NUMDES 850 /* máximo número de descripciones, debe ser
por lo menos uno más que el número real */
#define MAXNOD 1300 /* máximo número de nodos en el árbol */
#define LOPTER 7 /* máxima longitud de los operadores terminales,
debe ser mayor o igual a 2 */
```

```
FILE *fopen(), *fpent, *fpsal;
```

```
main (argc,argv) /* construye árboles */
```

```
int argc;
```

```
char *argv[ ];
```

```
{
```

```
char des[NUMDES][LONDES]; /* tabla con descripciones */
```

```
int totdes; /* total de descripciones */
```

```
char arbol[MAXNOD][LOPTER]; /* tabla de operadores y
caracteres */
```

```

int hijos[MAXNOD]; /* tabla de contadores de hijos */
int apun[MAXNOD]; /* tabla con apuntadores */
int par[NUMDES]; /* particiones que controlan */
int paraux[NUMDES]; /* particiones que se crean */
int numpar; /* número de particiones */
int c, l, n; /* del carácter en las descripciones: contenido, horizontal
y vertical */

int i, j, k;
int cd; /* índice horizontal en las descripciones */
int cn; /* apuntador al nodo actual */
int niv; /* índice vertical en las descripciones */
int p; /* índice en las particiones */
int cnh; /* apuntador secuencial a hijos */
int aptmp; /* temporal para crear apuntadores */
int fin; /* 0 cuando en la partición hay ceros */
char oporci; /* operador de % = nodo de base */

```

/* LECTURA E IMPRESIÓN DE LAS DESCRIPCIONES */

```

fpent = fopen(argv[1], "r");
fpsal = fopen(argv[2], "w");

for (i=0; i< NUMDES; ++i)
for (j=0; j< LONDES; ++j)
    des[i][j] = '\0';
l=0;
totdes=0;
while ((c = getc(fpent)) != EOF)
    if (c != '\n') { des[totdes][l] = c; ++l; }
    else { ++totdes; l=0; } ;
printf("Número de descripciones = %04d\n", totdes);

for (i=0; i< totdes; ++i)
{
    for (j=0; j< LONDES; ++j)
        printf("%c", des[i][j]);
    printf("\n");
};
printf("\n");

```

/* CONSTRUCCIÓN DE LA ARBORESCENCIA */

```

for (i=0; i< NUMDES; ++i) { par[i]=0; paraux[i]=0; } ;
for (i=0; i< MAXNOD; ++i)
{

```

```

    for (j=0; j<LOPTER; ++j) arbol[i][j] = '\0';
    hijos[i]=0; apun[i]=0;
};
fin = 1;
numpar = 1;
par[numpar] = totdes;
niv = 0;
n = 0;
p = 1;
cd = 0;
for (j=0; j< NUMDES; ++j) paraux[j] = 1;
cnh = 1;
arbol[1][0] = '*';
arbol[1][1] = '<';
cn = 2;
while (fin == 1)
{
    for (j = 1; j <= numpar; ++j)
    {
        if (par[j] == 0) paraux[p] = 0;
        else
        {
            while (n < par[j] - 1)
            {
                printf("%c %c\n", des[niv][cd], des[niv + 1][cd]);
                if (des[niv][cd] == des[niv + 1][cd])
                    ++ paraux[p];
                else
                {
                    if (des[niv][cd] == ',')
                    {
                        for (k=0; k<LOPTER; ++k)
                            arbol[cn][k] = des[niv][k + 1 + cd];
                        paraux[p] = 0;
                    }
                    else
                    {
                        arbol[cn][0] = '*';
                        arbol[cn][1] = des[niv][cd];
                    }
                }
                ++ hijos[cnh];
                printf("\n%3d: %c %c %2d", cnh, arbol[cnh][0],
                    arbol[cnh][1], hijos[cnh]);
                printf("\n%3d: %c %c %2d", cn, arbol[cn][0],
                    arbol[cn][1], hijos[cn]);
                printf("\n");
            }
        }
    }
}

```

```

        ++cn;
        ++p;
    };
    ++niv;
    ++n;
};
if (des[niv][cd] == ',')
{
    for (k=0; k< LOPTER; ++k)
        arbol[cn][k]=des[niv][k+1+cd];
    paraux[p]=0;
}
else
{
    arbol[cn][0]='*';
    arbol[cn][1]=des[niv][cd];
};
++hijos[cnh];
printf("\n%3d: %c %c %2d", cnh, arbol[cnh][0],
    arbol[cnh][1], hijos[cnh]);
printf("\n%3d: %c %c %2d", cn, arbol[cn][0],
    arbol[cn][1], hijos[cn]);
printf("\n");
++cn;
++cnh;
while (arbol[cnh][0] != '*' && cnh< cn) ++cnh;
};
++p;
++niv;
n=0;
};
niv=0;
++cd;
numpar=p-1;
fin=0;
for (k=1; k<= numpar; ++k)
if (paraux[k] != 0) fin=1;
for (j=0; j< NUMDES; ++j) par[j]=paraux[j];
for (j=0; j< NUMDES; ++j) paraux[j]=1;
printf("\n");
for (j=0; j< NUMDES; ++j) printf("%2d", j);
printf("\n");
for (j=0; j< numpar; ++j) printf("%2d", par[j]);
printf("\n");
p=1;
};

```

```
printf("\nNúmero de nodos %04d\n", cn);
```

```
/* 

|                          |
|--------------------------|
| CÁLCULO DE LOS APUNTADES |
|--------------------------|

 */
```

```
apun[1] = 2;
aptmp = apun[1] + hijos[1];
for (j = 2; j < cn; ++j)
    if (arbol[j][0] == '*')
    {
        apun[j] = aptmp;
        aptmp = apun[j] + hijos[j];
    };
```

```
/* 

|                     |
|---------------------|
| IMPRESIÓN DEL ÁRBOL |
|---------------------|

 */
```

```
oporci = '%';
for (j = 1; j < cn; ++j)
{
    switch(arbol[j][0])
    {
        case '*':
            if (arbol[j][1] == '!')
                fprintf(fpsal, "%c%02d%04d", oporci, hijos[j], apun[j]);
            else
                fprintf(fpsal, "%c%c%02d%04d", arbol[j][0], arbol[j][1],
                    hijos[j], apun[j]);
            break;
        default:
            k = 0;
            while (arbol[j][k] != '\0' && k < LOPTER)
                fprintf(fpsal, "%c", arbol[j][k++]);
            break;
    };
    fprintf(fpsal, "\n");
};
};
```

Descripciones para crear la tabla de reconocimiento de
etiquetas y subcampos de MARC

020) !< \$a,0	110) !< \$w,0	245) !< \$c,0	410) !< \$w,0
035) !< \$a,0	111) !< \$a,0	245) !< \$h,0	410) !< \$x,0
040) !< \$a,0	111) !< \$b,0	245) !< \$n,0	411) !< \$a,0
040) !< \$b,0	111) !< \$c,0	245) !< \$p,0	411) !< \$b,0
040) !< \$c,0	111) !< \$d,0	250) !< \$a,0	411) !< \$c,0
040) !< \$d,0	111) !< \$e,0	250) !< \$b,0	411) !< \$d,0
041) !< \$a,0	111) !< \$f,0	260) !< \$a,0	411) !< \$e,0
043) !< \$a,0	111) !< \$k,0	260) !< \$b,0	411) !< \$f,0
049) !< \$a,0	111) !< \$l,0	260) !< \$c,0	411) !< \$k,0
050) !< \$a,0	111) !< \$n,0	300) !< \$a,0	411) !< \$l,0
050) !< \$b,0	111) !< \$p,0	300) !< \$b,0	411) !< \$n,0
080) !< \$a,0	111) !< \$q,0	300) !< \$c,0	411) !< \$p,0
082) !< \$a,0	111) !< \$w,0	300) !< \$e,0	411) !< \$q,0
092) !< \$a,0	130) !< \$a,0	300) !< \$f,0	411) !< \$t,0
092) !< \$b,0	130) !< \$d,0	300) !< \$g,0	411) !< \$v,0
098) !< \$a,0	130) !< \$f,0	400) !< \$a,0	411) !< \$w,0
099) !< \$a,0	130) !< \$g,0	400) !< \$b,0	411) !< \$x,0
099) !< \$b,0	130) !< \$h,0	400) !< \$c,0	440) !< \$a,0
100) !< \$a,0	130) !< \$k,0	400) !< \$d,0	440) !< \$n,0
100) !< \$b,0	130) !< \$l,0	400) !< \$e,0	440) !< \$p,0
100) !< \$c,0	130) !< \$n,0	400) !< \$f,0	440) !< \$v,0
100) !< \$d,0	130) !< \$p,0	400) !< \$k,0	440) !< \$x,0
100) !< \$e,0	130) !< \$s,0	400) !< \$l,0	490) !< \$a,0
100) !< \$f,0	130) !< \$t,0	400) !< \$n,0	490) !< \$l,0
100) !< \$k,0	130) !< \$w,0	400) !< \$p,0	490) !< \$n,0
100) !< \$l,0	240) !< \$a,0	400) !< \$q,0	490) !< \$v,0
100) !< \$n,0	240) !< \$d,0	400) !< \$t,0	490) !< \$x,0
100) !< \$p,0	240) !< \$f,0	400) !< \$v,0	500) !< \$a,0
100) !< \$q,0	240) !< \$g,0	400) !< \$w,0	501) !< \$a,0
100) !< \$t,0	240) !< \$h,0	400) !< \$x,0	502) !< \$a,0
100) !< \$w,0	240) !< \$k,0	410) !< \$a,0	502) !< \$b,0
110) !< \$a,0	240) !< \$l,0	410) !< \$b,0	503) !< \$a,0
110) !< \$b,0	240) !< \$m,0	410) !< \$c,0	504) !< \$a,0
110) !< \$c,0	240) !< \$n,0	410) !< \$d,0	505) !< \$a,0
110) !< \$d,0	240) !< \$p,0	410) !< \$e,0	507) !< \$a,0
110) !< \$e,0	240) !< \$r,0	410) !< \$f,0	520) !< \$a,0
110) !< \$f,0	240) !< \$s,0	410) !< \$g,0	533) !< \$a,0
110) !< \$g,0	240) !< \$w,0	410) !< \$k,0	533) !< \$b,0
110) !< \$k,0	241) !< \$a,0	410) !< \$l,0	533) !< \$c,0
110) !< \$l,0	241) !< \$b,0	410) !< \$n,0	546) !< \$a,0
110) !< \$n,0	241) !< \$c,0	410) !< \$p,0	590) !< \$a,0
110) !< \$p,0	245) !< \$a,0	410) !< \$t,0	690) !< \$a,0
110) !< \$t,0	245) !< \$b,0	410) !< \$v,0	690) !< \$b,0

Descripciones para crear la tabla de reconocimiento de
etiquetas y subcampos de MARC

690) !< \$x,0	693) !< \$p,0	700) !< \$l,0	730) !< \$f,0
690) !< \$y,0	693) !< \$r,0	700) !< \$m,0	730) !< \$g,0
690) !< \$z,0	693) !< \$s,0	700) !< \$n,0	730) !< \$h,0
691) !< \$a,0	693) !< \$t,0	700) !< \$o,0	730) !< \$k,0
691) !< \$b,0	693) !< \$w,0	700) !< \$p,0	730) !< \$l,0
691) !< \$x,0	693) !< \$x,0	700) !< \$q,0	730) !< \$m,0
691) !< \$y,0	693) !< \$y,0	700) !< \$t,0	730) !< \$n,0
691) !< \$z,0	693) !< \$z,0	700) !< \$w,0	730) !< \$o,0
692) !< \$a,0	694) !< \$a,0	710) !< \$a,0	730) !< \$p,0
692) !< \$b,0	694) !< \$b,0	710) !< \$b,0	730) !< \$r,0
692) !< \$c,0	694) !< \$c,0	710) !< \$c,0	730) !< \$s,0
692) !< \$d,0	694) !< \$d,0	710) !< \$d,0	730) !< \$t,0
692) !< \$e,0	694) !< \$e,0	710) !< \$e,0	730) !< \$w,0
692) !< \$f,0	694) !< \$g,0	710) !< \$f,0	740) !< \$a,0
692) !< \$g,0	694) !< \$h,0	710) !< \$g,0	740) !< \$b,0
692) !< \$h,0	694) !< \$q,0	710) !< \$h,0	740) !< \$c,0
692) !< \$k,0	694) !< \$t,0	710) !< \$k,0	740) !< \$h,0
692) !< \$l,0	694) !< \$w,0	710) !< \$l,0	740) !< \$n,0
692) !< \$m,0	694) !< \$x,0	710) !< \$m,0	740) !< \$p,0
692) !< \$n,0	694) !< \$y,0	710) !< \$n,0	800) !< \$a,0
692) !< \$o,0	694) !< \$z,0	710) !< \$o,0	800) !< \$b,0
692) !< \$p,0	695) !< \$a,0	710) !< \$p,0	800) !< \$c,0
692) !< \$q,0	695) !< \$d,0	710) !< \$r,0	800) !< \$d,0
692) !< \$r,0	695) !< \$f,0	710) !< \$s,0	800) !< \$e,0
692) !< \$s,0	695) !< \$h,0	710) !< \$t,0	800) !< \$f,0
692) !< \$t,0	695) !< \$k,0	710) !< \$w,0	800) !< \$h,0
692) !< \$w,0	695) !< \$l,0	711) !< \$a,0	800) !< \$k,0
692) !< \$x,0	695) !< \$n,0	711) !< \$b,0	800) !< \$l,0
692) !< \$y,0	695) !< \$p,0	711) !< \$c,0	800) !< \$m,0
692) !< \$z,0	695) !< \$s,0	711) !< \$d,0	800) !< \$n,0
693) !< \$a,0	695) !< \$t,0	711) !< \$e,0	800) !< \$p,0
693) !< \$b,0	695) !< \$w,0	711) !< \$f,0	800) !< \$q,0
693) !< \$c,0	695) !< \$x,0	711) !< \$h,0	800) !< \$s,0
693) !< \$d,0	695) !< \$y,0	711) !< \$k,0	800) !< \$t,0
693) !< \$e,0	695) !< \$z,0	711) !< \$l,0	800) !< \$v,0
693) !< \$f,0	700) !< \$a,0	711) !< \$n,0	810) !< \$a,0
693) !< \$g,0	700) !< \$b,0	711) !< \$p,0	810) !< \$b,0
693) !< \$h,0	700) !< \$c,0	711) !< \$q,0	810) !< \$c,0
693) !< \$k,0	700) !< \$d,0	711) !< \$s,0	810) !< \$d,0
693) !< \$l,0	700) !< \$e,0	711) !< \$t,0	810) !< \$e,0
693) !< \$m,0	700) !< \$f,0	711) !< \$w,0	810) !< \$f,0
693) !< \$n,0	700) !< \$h,0	730) !< \$a,0	810) !< \$g,0
693) !< \$o,0	700) !< \$k,0	730) !< \$d,0	810) !< \$h,0

Descripciones para crear la tabla de reconocimiento de
etiquetas y subcampos de MARC

810>!<\$k,0	811>!<\$h,0	830>!<\$n,0	F1>,0
810>!<\$l,0	811>!<\$k,0	830>!<\$o,0	F2>,0
810>!<\$m,0	811>!<\$l,0	830>!<\$p,0	FCAP>,0
810>!<\$n,0	811>!<\$n,0	830>!<\$r,0	FIN>,\$
810>!<\$p,0	811>!<\$p,0	830>!<\$s,0	FSPUB>,0
810>!<\$r,0	811>!<\$q,0	830>!<\$t,0	IDIOMA>,0
810>!<\$s,0	811>!<\$t,0	830>!<\$v,0	ILUS>,0
810>!<\$t,0	811>!<\$v,0	840>!<\$a,0	IND>,0
810>!<\$v,0	830>!<\$a,0	840>!<\$v,0	LEST>,0
811>!<\$a,0	830>!<\$d,0	901>!<\$a,0	LTIP>,0
811>!<\$b,0	830>!<\$f,0	902>!<\$a,0	LNBIB>,0
811>!<\$c,0	830>!<\$g,0	AP>,0	LNDES>,0
811>!<\$d,0	830>!<\$h,0	BIO>,0	LFOR>,0
811>!<\$e,0	830>!<\$k,0	CONF>,0	PAIS>,0
811>!<\$f,0	830>!<\$l,0	CONT>,0	PUBOF>,0
811>!<\$g,0	830>!<\$m,0	EDCONM>,0	REPROD>,0

Tabla en la que se representa a los autómatas conectados

1: * <	19	2	45: * 1	2 117	89: * 1	1 165
2: * 0	6	21	46: * 3	1 119	90: * 5	1 166
3: * 1	3	27	47: * 4	1 120	91: * 0	1 167
4: * 2	3	30	48: * 0	1 121	92: * 0	1 168
5: * 3	1	33	49: * 1	2 122	93: * 0	1 169
6: * 4	4	34	50: * 3	1 124	94: * 0	1 170
7: * 5	5	38	51: * 4	1 125	95: * 0	1 171
8: * 6	1	43	52: * 0	2 126	96: * 1	1 172
9: * 7	4	44	53: * P	1 128	97: * 0	1 173
10: * 8	4	48	54: * I	1 129	98: * 0	1 174
11: * 9	1	52	55: * O	1 130	99: * 0	1 175
12: * A	1	53	56: * D	1 131	100: * 1	1 176
13: * B	1	54	57: * 1	1 132	101: * 2	1 177
14: * C	1	55	58: * 2	1 133	102: * 3	1 178
15: * E	1	56	59: * C	1 134	103: * 4	1 179
16: * F	5	57	60: * I	1 135	104: * 5	1 180
17: * I	3	62	61: * S	1 136	105: * 7	1 181
18: * L	4	65	62: * D	1 137	106: * 0	1 182
19: * P	2	69	63: * L	1 138	107: * 3	1 183
20: * R	1	71	64: * N	1 139	108: * 6	1 184
21: * 2	1	72	65: * E	1 140	109: * 0	1 185
22: * 3	1	73	66: * T	1 141	110: * 0	1 186
23: * 4	4	74	67: * N	2 142	111: * 1	1 187
24: * 5	1	78	68: * F	1 144	112: * 2	1 188
25: * 8	2	79	69: * A	1 145	113: * 3	1 189
26: * 9	3	81	70: * U	1 146	114: * 4	1 190
27: * 0	1	84	71: * E	1 147	115: * 5	1 191
28: * 1	2	85	72: * 0	1 148	116: * 0	1 192
29: * 3	1	87	73: * 5	1 149	117: * 0	1 193
30: * 4	3	88	74: * 0	1 150	118: * 1	1 194
31: * 5	1	91	75: * 1	1 151	119: * 0	1 195
32: * 6	1	92	76: * 3	1 152	120: * 0	1 196
33: * 0	1	93	77: * 9	1 153	121: * 0	1 197
34: * 0	1	94	78: * 0	1 154	122: * 0	1 198
35: * 1	2	95	79: * 0	1 155	123: * 1	1 199
36: * 4	1	97	80: * 2	1 156	124: * 0	1 200
37: * 9	1	98	81: * 2	1 157	125: * 0	1 201
38: * 0	7	99	82: * 8	1 158	126: * 1	1 202
39: * 2	1	106	83: * 9	1 159	127: * 2	1 203
40: * 3	1	107	84: * 0	1 160	128: * >	1 204
41: * 4	1	108	85: * 0	1 161	129: * O	1 205
42: * 9	1	109	86: * 1	1 162	130: * N	2 206
43: * 9	6	110	87: * 0	1 163	131: * C	1 208
44: * 0	1	116	88: * 0	1 164	132: * >	1 209

Tabla en la que se representa a los autómatas conectados

133: * >	1 210	178: * >	1 255	223: * O	1 297
134: * A	1 211	179: * >	1 256	224: * R	1 298
135: * N	1 212	180: * >	1 257	225: %	299
136: * P	1 213	181: * >	1 258	226: %	300
137: * I	1 214	182: * >	1 259	227: %	301
138: * U	1 215	183: * >	1 260	228: %	302
139: * D	1 216	184: * >	1 261	229: %	303
140: * S	1 217	185: * >	1 262	230: %	304
141: * I	1 218	186: * >	1 263	231: %	305
142: * B	1 219	187: * >	1 264	232: %	306
143: * D	1 220	188: * >	1 265	233: %	307
144: * O	1 221	189: * >	1 266	234: %	308
145: * I	1 222	190: * >	1 267	235: %	309
146: * B	1 223	191: * >	1 268	236: %	310
147: * P	1 224	192: * >	1 269	237: %	311
148: * >	1 225	193: * >	1 270	238: %	312
149: * >	1 226	194: * >	1 271	239: %	313
150: * >	1 227	195: * >	1 272	240: %	314
151: * >	1 228	196: * >	1 273	241: %	315
152: * >	1 229	197: * >	1 274	242: %	316
153: * >	1 230	198: * >	1 275	243: %	317
154: * >	1 231	199: * >	1 276	244: %	318
155: * >	1 232	200: * >	1 277	245: %	319
156: * >	1 233	201: * >	1 278	246: %	320
157: * >	1 234	202: * >	1 279	247: %	321
158: * >	1 235	203: * >	1 280	248: %	322
159: * >	1 236	204: 0		249: %	323
160: * >	1 237	205: * >	1 281	250: %	324
161: * >	1 238	206: * F	1 282	251: %	325
162: * >	1 239	207: * T	1 283	252: %	326
163: * >	1 240	208: * O	1 284	253: %	327
164: * >	1 241	209: 0		254: %	328
165: * >	1 242	210: 0		255: %	329
166: * >	1 243	211: * P	1 285	256: %	330
167: * >	1 244	212: * >	1 286	257: %	331
168: * >	1 245	213: * U	1 287	258: %	332
169: * >	1 246	214: * O	1 288	259: %	333
170: * >	1 247	215: * S	1 289	260: %	334
171: * >	1 248	216: * >	1 290	261: %	335
172: * >	1 249	217: * T	1 291	262: %	336
173: * >	1 250	218: * P	1 292	263: %	337
174: * >	1 251	219: * I	1 293	264: %	338
175: * >	1 252	220: * E	1 294	265: %	339
176: * >	1 253	221: * R	1 295	266: %	340
177: * >	1 254	222: * S	1 296	267: %	341

Tabla en la que se representa a los autómatas conectados

268: 7/0 342	313: * < 1 384	358: 0
269: 7/0 343	314: * < 1 385	359: * > 1 427
270: 7/0 344	315: * < 1 386	360: * A 1 428
271: 7/0 345	316: * < 1 387	361: 0
272: 7/0 346	317: * < 1 388	362: 0
273: 7/0 347	318: * < 1 389	363: 0
274: 7/0 348	319: * < 1 390	364: * > 1 429
275: 7/0 349	320: * < 1 391	365: * > 1 430
276: 7/0 350	321: * < 1 392	366: 0
277: 7/0 351	322: * < 1 393	367: 0
278: 7/0 352	323: * < 1 394	368: * > 1 431
279: 7/0 353	324: * < 1 395	369: * D 1 432
280: 7/0 354	325: * < 1 396	370: * \$ 1 433
281: 0	326: * < 1 397	371: * \$ 1 434
282: * > 1 355	327: * < 1 398	372: * \$ 4 435
283: * > 1 356	328: * < 1 399	373: * \$ 1 439
284: * N 1 357	329: * < 1 400	374: * \$ 1 440
285: * > 1 358	330: * < 1 401	375: * \$ 1 441
286: \$	331: * < 1 402	376: * \$ 2 442
287: * B 1 359	332: * < 1 403	377: * \$ 1 444
288: * M 1 360	333: * < 1 404	378: * \$ 1 445
289: * > 1 361	334: * < 1 405	379: * \$ 2 446
290: 0	335: * < 1 406	380: * \$ 1 448
291: * > 1 362	336: * < 1 407	381: * \$ 2 449
292: * > 1 363	337: * < 1 408	382: * \$ 13 451
293: * B 1 364	338: * < 1 409	383: * \$ 13 464
294: * S 1 365	339: * < 1 410	384: * \$ 12 477
295: * > 1 366	340: * < 1 411	385: * \$ 12 489
296: * > 1 367	341: * < 1 412	386: * \$ 13 501
297: * F 1 368	342: * < 1 413	387: * \$ 3 514
298: * O 1 369	343: * < 1 414	388: * \$ 6 517
299: * < 1 370	344: * < 1 415	389: * \$ 2 523
300: * < 1 371	345: * < 1 416	390: * \$ 3 525
301: * < 1 372	346: * < 1 417	391: * \$ 6 528
302: * < 1 373	347: * < 1 418	392: * \$ 15 534
303: * < 1 374	348: * < 1 419	393: * \$ 15 549
304: * < 1 375	349: * < 1 420	394: * \$ 15 564
305: * < 1 376	350: * < 1 421	395: * \$ 5 579
306: * < 1 377	351: * < 1 422	396: * \$ 5 584
307: * < 1 378	352: * < 1 423	397: * \$ 1 589
308: * < 1 379	353: * < 1 424	398: * \$ 1 590
309: * < 1 380	354: * < 1 425	399: * \$ 2 591
310: * < 1 381	355: 0	400: * \$ 1 593
311: * < 1 382	356: 0	401: * \$ 1 594
312: * < 1 383	357: * M 1 426	402: * \$ 1 595

Tabla en la que se representa a los autómatas conectados

403: * \$	1 596	448: * a	1 838	493: * h	1 883
404: * \$	1 597	449: * a	1 839	494: * k	1 884
405: * \$	3 598	450: * b	1 840	495: * l	1 885
406: * \$	1 601	451: * a	1 841	496: * n	1 886
407: * \$	1 602	452: * b	1 842	497: * p	1 887
408: * \$	5 603	453: * c	1 843	498: * s	1 888
409: * \$	5 608	454: * d	1 844	499: * t	1 889
410: * \$	22 613	455: * e	1 845	500: * w	1 890
411: * \$	21 635	456: * f	1 846	501: * a	1 891
412: * \$	13 656	457: * k	1 847	502: * d	1 892
413: * \$	14 669	458: * l	1 848	503: * f	1 893
414: * \$	16 683	459: * n	1 849	504: * g	1 894
415: * \$	18 699	460: * p	1 850	505: * h	1 895
416: * \$	15 717	461: * q	1 851	506: * k	1 896
417: * \$	15 732	462: * t	1 852	507: * l	1 897
418: * \$	6 747	463: * w	1 853	508: * m	1 898
419: * \$	16 753	464: * a	1 854	509: * n	1 899
420: * \$	17 769	465: * b	1 855	510: * p	1 900
421: * \$	15 786	466: * c	1 856	511: * r	1 901
422: * \$	15 801	467: * d	1 857	512: * s	1 902
423: * \$	2 816	468: * e	1 858	513: * w	1 903
424: * \$	1 818	469: * f	1 859	514: * a	1 904
425: * \$	1 819	470: * g	1 860	515: * b	1 905
426: * >	1 820	471: * k	1 861	516: * c	1 906
427: 0		472: * l	1 862	517: * a	1 907
428: * >	1 821	473: * n	1 863	518: * b	1 908
429: 0		474: * p	1 864	519: * c	1 909
430: 0		475: * t	1 865	520: * h	1 910
431: 0		476: * w	1 866	521: * n	1 911
432: * >	1 822	477: * a	1 867	522: * p	1 912
433: * a	1 823	478: * b	1 868	523: * a	1 913
434: * a	1 824	479: * c	1 869	524: * b	1 914
435: * a	1 825	480: * d	1 870	525: * a	1 915
436: * b	1 826	481: * e	1 871	526: * b	1 916
437: * c	1 827	482: * f	1 872	527: * c	1 917
438: * d	1 828	483: * k	1 873	528: * a	1 918
439: * a	1 829	484: * l	1 874	529: * b	1 919
440: * a	1 830	485: * n	1 875	530: * c	1 920
441: * a	1 831	486: * p	1 876	531: * e	1 921
442: * a	1 832	487: * q	1 877	532: * f	1 922
443: * b	1 833	488: * w	1 878	533: * g	1 923
444: * a	1 834	489: * a	1 879	534: * a	1 924
445: * a	1 835	490: * d	1 880	535: * b	1 925
446: * a	1 836	491: * f	1 881	536: * c	1 926
447: * b	1 837	492: * g	1 882	537: * d	1 927

Tabla en la que se representa a los autómatas conectados

538: * e 1 928	583: * x 1 973	628: * r 1 1018
539: * f 1 929	584: * a 1 974	629: * s 1 1019
540: * k 1 930	585: * l 1 975	630: * t 1 1020
541: * l 1 931	586: * n 1 976	631: * w 1 1021
542: * n 1 932	587: * v 1 977	632: * x 1 1022
543: * p 1 933	588: * x 1 978	633: * y 1 1023
544: * q 1 934	589: * a 1 979	634: * z 1 1024
545: * t 1 935	590: * a 1 980	635: * a 1 1025
546: * v 1 936	591: * a 1 981	636: * b 1 1026
547: * w 1 937	592: * b 1 982	637: * c 1 1027
548: * x 1 938	593: * a 1 983	638: * d 1 1028
549: * a 1 939	594: * a 1 984	639: * e 1 1029
550: * b 1 940	595: * a 1 985	640: * f 1 1030
551: * c 1 941	596: * a 1 986	641: * g 1 1031
552: * d 1 942	597: * a 1 987	642: * h 1 1032
553: * e 1 943	598: * a 1 988	643: * k 1 1033
554: * f 1 944	599: * b 1 989	644: * l 1 1034
555: * g 1 945	600: * c 1 990	645: * m 1 1035
556: * k 1 946	601: * a 1 991	646: * n 1 1036
557: * l 1 947	602: * a 1 992	647: * o 1 1037
558: * n 1 948	603: * a 1 993	648: * p 1 1038
559: * p 1 949	604: * b 1 994	649: * r 1 1039
560: * t 1 950	605: * x 1 995	650: * s 1 1040
561: * v 1 951	606: * y 1 996	651: * t 1 1041
562: * w 1 952	607: * z 1 997	652: * w 1 1042
563: * x 1 953	608: * a 1 998	653: * x 1 1043
564: * a 1 954	609: * b 1 999	654: * y 1 1044
565: * b 1 955	610: * x 1 1000	655: * z 1 1045
566: * c 1 956	611: * y 1 1001	656: * a 1 1046
567: * d 1 957	612: * z 1 1002	657: * b 1 1047
568: * e 1 958	613: * a 1 1003	658: * c 1 1048
569: * f 1 959	614: * b 1 1004	659: * d 1 1049
570: * k 1 960	615: * c 1 1005	660: * e 1 1050
571: * l 1 961	616: * d 1 1006	661: * g 1 1051
572: * n 1 962	617: * e 1 1007	662: * h 1 1052
573: * p 1 963	618: * f 1 1008	663: * q 1 1053
574: * q 1 964	619: * g 1 1009	664: * t 1 1054
575: * t 1 965	620: * h 1 1010	665: * w 1 1055
576: * v 1 966	621: * k 1 1011	666: * x 1 1056
577: * w 1 967	622: * l 1 1012	667: * y 1 1057
578: * x 1 968	623: * m 1 1013	668: * z 1 1058
579: * a 1 969	624: * n 1 1014	669: * a 1 1059
580: * n 1 970	625: * o 1 1015	670: * d 1 1060
581: * p 1 971	626: * p 1 1016	671: * f 1 1061
582: * v 1 972	627: * q 1 1017	672: * h 1 1062

Tabla en la que se representa a los autómatas conectados

673: * k	1 1063	718: * b	1 1108	763: * n	1 1153
674: * l	1 1064	719: * c	1 1109	764: * p	1 1154
675: * n	1 1065	720: * d	1 1110	765: * q	1 1155
676: * p	1 1066	721: * e	1 1111	766: * s	1 1156
677: * s	1 1067	722: * f	1 1112	767: * t	1 1157
678: * t	1 1068	723: * h	1 1113	768: * v	1 1158
679: * w	1 1069	724: * k	1 1114	769: * a	1 1159
680: * x	1 1070	725: * l	1 1115	770: * b	1 1160
681: * y	1 1071	726: * n	1 1116	771: * c	1 1161
682: * z	1 1072	727: * p	1 1117	772: * d	1 1162
683: * a	1 1073	728: * q	1 1118	773: * e	1 1163
684: * b	1 1074	729: * s	1 1119	774: * f	1 1164
685: * c	1 1075	730: * t	1 1120	775: * g	1 1165
686: * d	1 1076	731: * w	1 1121	776: * h	1 1166
687: * e	1 1077	732: * a	1 1122	777: * k	1 1167
688: * f	1 1078	733: * d	1 1123	778: * l	1 1168
689: * h	1 1079	734: * f	1 1124	779: * m	1 1169
690: * k	1 1080	735: * g	1 1125	780: * n	1 1170
691: * l	1 1081	736: * h	1 1126	781: * p	1 1171
692: * m	1 1082	737: * k	1 1127	782: * r	1 1172
693: * n	1 1083	738: * l	1 1128	783: * s	1 1173
694: * o	1 1084	739: * m	1 1129	784: * t	1 1174
695: * p	1 1085	740: * n	1 1130	785: * v	1 1175
696: * q	1 1086	741: * o	1 1131	786: * a	1 1176
697: * t	1 1087	742: * p	1 1132	787: * b	1 1177
698: * w	1 1088	743: * r	1 1133	788: * c	1 1178
699: * a	1 1089	744: * s	1 1134	789: * d	1 1179
700: * b	1 1090	745: * t	1 1135	790: * e	1 1180
701: * c	1 1091	746: * w	1 1136	791: * f	1 1181
702: * d	1 1092	747: * a	1 1137	792: * g	1 1182
703: * e	1 1093	748: * b	1 1138	793: * h	1 1183
704: * f	1 1094	749: * c	1 1139	794: * k	1 1184
705: * g	1 1095	750: * h	1 1140	795: * l	1 1185
706: * h	1 1096	751: * n	1 1141	796: * n	1 1186
707: * k	1 1097	752: * p	1 1142	797: * p	1 1187
708: * l	1 1098	753: * a	1 1143	798: * q	1 1188
709: * m	1 1099	754: * b	1 1144	799: * t	1 1189
710: * n	1 1100	755: * c	1 1145	800: * v	1 1190
711: * o	1 1101	756: * d	1 1146	801: * a	1 1191
712: * p	1 1102	757: * e	1 1147	802: * d	1 1192
713: * r	1 1103	758: * f	1 1148	803: * f	1 1193
714: * s	1 1104	759: * h	1 1149	804: * g	1 1194
715: * t	1 1105	760: * k	1 1150	805: * h	1 1195
716: * w	1 1106	761: * l	1 1151	806: * k	1 1196
717: * a	1 1107	762: * m	1 1152	807: * l	1 1197

Tabla en la que se representa a los autómatas conectados

808: * m 1 1198	853: 0	898: 0
809: * n 1 1199	854: 0	899: 0
810: * o 1 1200	855: 0	900: 0
811: * p 1 1201	856: 0	901: 0
812: * r 1 1202	857: 0	902: 0
813: * s 1 1203	858: 0	903: 0
814: * t 1 1204	859: 0	904: 0
815: * v 1 1205	860: 0	905: 0
816: * a 1 1206	861: 0	906: 0
817: * v 1 1207	862: 0	907: 0
818: * a 1 1208	863: 0	908: 0
819: * a 1 1209	864: 0	909: 0
820: 0	865: 0	910: 0
821: 0	866: 0	911: 0
822: 0	867: 0	912: 0
823: 0	868: 0	913: 0
824: 0	869: 0	914: 0
825: 0	870: 0	915: 0
826: 0	871: 0	916: 0
827: 0	872: 0	917: 0
828: 0	873: 0	918: 0
829: 0	874: 0	919: 0
830: 0	875: 0	920: 0
831: 0	876: 0	921: 0
832: 0	877: 0	922: 0
833: 0	878: 0	923: 0
834: 0	879: 0	924: 0
835: 0	880: 0	925: 0
836: 0	881: 0	926: 0
837: 0	882: 0	927: 0
838: 0	883: 0	928: 0
839: 0	884: 0	929: 0
840: 0	885: 0	930: 0
841: 0	886: 0	931: 0
842: 0	887: 0	932: 0
843: 0	888: 0	933: 0
844: 0	889: 0	934: 0
845: 0	890: 0	935: 0
846: 0	891: 0	936: 0
847: 0	892: 0	937: 0
848: 0	893: 0	938: 0
849: 0	894: 0	939: 0
850: 0	895: 0	940: 0
851: 0	896: 0	941: 0
852: 0	897: 0	942: 0

Tabla en la que se representa a los autómatas conectados

943: 0	988: 0	1033: 0
944: 0	989: 0	1034: 0
945: 0	990: 0	1035: 0
946: 0	991: 0	1036: 0
947: 0	992: 0	1037: 0
948: 0	993: 0	1038: 0
949: 0	994: 0	1039: 0
950: 0	995: 0	1040: 0
951: 0	996: 0	1041: 0
952: 0	997: 0	1042: 0
953: 0	998: 0	1043: 0
954: 0	999: 0	1044: 0
955: 0	1000: 0	1045: 0
956: 0	1001: 0	1046: 0
957: 0	1002: 0	1047: 0
958: 0	1003: 0	1048: 0
959: 0	1004: 0	1049: 0
960: 0	1005: 0	1050: 0
961: 0	1006: 0	1051: 0
962: 0	1007: 0	1052: 0
963: 0	1008: 0	1053: 0
964: 0	1009: 0	1054: 0
965: 0	1010: 0	1055: 0
966: 0	1011: 0	1056: 0
967: 0	1012: 0	1057: 0
968: 0	1013: 0	1058: 0
969: 0	1014: 0	1059: 0
970: 0	1015: 0	1060: 0
971: 0	1016: 0	1061: 0
972: 0	1017: 0	1062: 0
973: 0	1018: 0	1063: 0
974: 0	1019: 0	1064: 0
975: 0	1020: 0	1065: 0
976: 0	1021: 0	1066: 0
977: 0	1022: 0	1067: 0
978: 0	1023: 0	1068: 0
979: 0	1024: 0	1069: 0
980: 0	1025: 0	1070: 0
981: 0	1026: 0	1071: 0
982: 0	1027: 0	1072: 0
983: 0	1028: 0	1073: 0
984: 0	1029: 0	1074: 0
985: 0	1030: 0	1075: 0
986: 0	1031: 0	1076: 0
987: 0	1032: 0	1077: 0

Tabla en la que se representa a los autómatas conectados

1078: 0	1122: 0	1166: 0
1079: 0	1123: 0	1167: 0
1080: 0	1124: 0	1168: 0
1081: 0	1125: 0	1169: 0
1082: 0	1126: 0	1170: 0
1083: 0	1127: 0	1171: 0
1084: 0	1128: 0	1172: 0
1085: 0	1129: 0	1173: 0
1086: 0	1130: 0	1174: 0
1087: 0	1131: 0	1175: 0
1088: 0	1132: 0	1176: 0
1089: 0	1133: 0	1177: 0
1090: 0	1134: 0	1178: 0
1091: 0	1135: 0	1179: 0
1092: 0	1136: 0	1180: 0
1093: 0	1137: 0	1181: 0
1094: 0	1138: 0	1182: 0
1095: 0	1139: 0	1183: 0
1096: 0	1140: 0	1184: 0
1097: 0	1141: 0	1185: 0
1098: 0	1142: 0	1186: 0
1099: 0	1143: 0	1187: 0
1100: 0	1144: 0	1188: 0
1101: 0	1145: 0	1189: 0
1102: 0	1146: 0	1190: 0
1103: 0	1147: 0	1191: 0
1104: 0	1148: 0	1192: 0
1105: 0	1149: 0	1193: 0
1106: 0	1150: 0	1194: 0
1107: 0	1151: 0	1195: 0
1108: 0	1152: 0	1196: 0
1109: 0	1153: 0	1197: 0
1110: 0	1154: 0	1198: 0
1111: 0	1155: 0	1199: 0
1112: 0	1156: 0	1200: 0
1113: 0	1157: 0	1201: 0
1114: 0	1158: 0	1202: 0
1115: 0	1159: 0	1203: 0
1116: 0	1160: 0	1204: 0
1117: 0	1161: 0	1205: 0
1118: 0	1162: 0	1206: 0
1119: 0	1163: 0	1207: 0
1120: 0	1164: 0	1208: 0
1121: 0	1165: 0	1209: 0

```

/* Programa:      Semantic.c
Objetivo:        Con este programa se codifica el contenido de los
                  registros de conexión y de los registros agregados
                  de la tabla en la que se representan los autómatas.

Datos de Entrada: Un archivo con la tabla en que se representan los
                  autómatas, en la cual se codificó en todos los re-
                  gistros agregados el operador semántico "0", que
                  por convención se utiliza para omisión de conteni-
                  do.

                  Un archivo con las etiquetas para las cuales se de-
                  sea codificar un operador diferente al "0", y los
                  operadores con sus parámetros correspondientes.
                  Por ejemplo:

                  035> 038075
                  $a # 030281
                  ;
                  100> 050312
                  $a&020011
                  $b # 030011
                  ;

```

El programa agregará en el registro de conexión de la etiqueta correspondiente la información codificada a continuación de dicha etiqueta. Por ejemplo, en el correspondiente a la etiqueta 035> agregará 038075. En las siguientes líneas (hasta encontrar un ";") están los operadores que se desean codificar en los registros agregados a los nodos terminales correspondientes a los subcampos de la etiqueta; por ejemplo, en el registro agregado al nodo \$a de la etiqueta 100> se pondrá &020011. El ";" nos indica el fin de los subcampos de la etiqueta anterior.

Datos de salida: La tabla con los operadores y parámetros.

Método: Se lee la tabla y se entra a una iteración mientras haya descripciones por leer. Se rastrea cada descripción utilizando el algoritmo de análisis léxico y al alcanzar un registro de conexión o un registro agregado se guardan los nuevos operadores y sus parámetros en un arreglo de cadenas paralelo a la tabla.

El programa se ejecuta de la siguiente manera:
 semantic.ex < arch-des > < arbol-sal >

```

#include < stdio.h >
#define MAXNOD 1300
FILE *fopen( ),*fp,*fpd,*fpas;

```

*/

```

struct nodo { /* La Tabla */
    char arbol[2]; /* operadores y caracteres por comparar */
    int hijos; /* número de hijos */
    int apun; /* apuntador a primer hijo */
    char desc[30]; /* para guardar lo que sobra de la descripción */
    } ar[MAXNOD];
char dl[35]; /* descripción que leo */
int cn; /* nodo actual */
int hijo; /* número de hijos actual */
int base; /* base del subárbol */
int c;
int ap; /* apuntador a descripción en uso */
int fin,sigue;
int i;

main(argc,argv)
int argc;
char *argv[ ];
{
    if (argc! = 3)
    {
        printf("Faltan o sobran nombres de archivos\n");
        printf("Dar el nombre de la descripción y del árbol a
            generar\n");
    }
    else
    {
        if((fpd = fopen(* ++ argv, "r")) = = NULL)
            printf("No existe ese archivo para la descripción\n");
        else
        {
            fpas = fopen(* ++ argv, "w");
            fp = fopen("arbolpuro", "r");
            for(i = 0; i < MAXNOD; ++i)
                ar[i].arbol[0] = ' ';
        }
    }
}

```

/* LECTURA DEL ÁRBOL PURO */

```

cn = 1;
while (fscanf(fp, "%1s", &c) != EOF)
{
    switch (c)
    {
        case '*':
            ar[cn].arbol[0] = c;
    }
}

```

```

fscanf(fp, "%c%2d%4d", &ar[cn].arbol[1], &ar[cn].hijos,
&ar[cn].apun);
break;
case '$':
case '0':
    ar[cn].arbol[0] = c;
    break;
case '%':
    ar[cn].arbol[0] = c;
    fscanf(fp, "%4d", &ar[cn].apun);
    break;
} /* del switch */
++cn;
} /* de while */

```

```

/* 

|                                                                    |
|--------------------------------------------------------------------|
| LECTURA DE DESCRIPCIONES Y SUSTITUCIÓN<br>DE OPERADORES SEMÁNTICOS |
|--------------------------------------------------------------------|

 */

```

```

fin = 1;
c = '<';
cn = 1;
hijo = ar[1].hijos;
while((fscanf(fp, "%s", &dl) != EOF) && fin)
{
    ap = (-1);
    sigue = 1;
    if (dl[0] == ';')
    {
        cn = 1;
        c = '<';
        hijo = ar[1].hijos;
    }
    else
    while (sigue) /* sigue mientras estamos en etiqueta (disyun) */
    switch (ar[cn].arbol[0])
    {
        case '*': disyun( );
                break;
        case '%': subarbol( );
                break;
        case '0': copia( );
                break;
    }; /* del switch */
} /* del while */

```

```

/* ESCRITURA DEL ÁRBOL MODIFICADO */
cn = 1;
while ((c = ar[cn].arbol[0]) != ' ')
{
    switch (c)
    {
        case '*':
            fprintf(fpas, "%c%c%02d%04d\n", ar[cn].arbol[0],
                ar[cn].arbol[1], ar[cn].hijos, ar[cn].apun);
            break;
        case '$':
        case '0':
            fprintf(fpas, "%c\n", ar[cn].arbol[0]);
            break;
        case '%':
            fprintf(fpas, "%c%04d%s\n", ar[cn].arbol[0], ar[cn].apun,
                ar[cn].desc);
            break;
        default:
            fprintf(fpas, "%s\n", ar[cn].desc);
    } /* del switch */
    ++cn;
} /* de while */
for (i = 1; i < MAXNOD; ++i)
    printf("= %s = \n", ar[i].desc);

        close(fpd);
        close(fp)
        close(fpas);
    }
}

```

```

/* OPERADORES SEMÁNTICOS */

```

```

/* DISYUNCIÓN */
disyun( )
{
    while (hijo > 0)
    {
        if ((c == '\n') || (c == EOF))
        {

```

```

printf("Error en descripción, terminé mal \n");
sigue = 0;
fin = 0;
return;
}
else
{
    if (c == ar[cn].arbol[1]) /* carácter actual igual a carácter
                               del nodo */
    {
        hijo = ar[cn].hijos;
        cn = ar[cn].apun;
        c = dl[++ap]; /* obtengo el siguiente carácter de la
                       descripción */
        return;
    }
    else
    {
        --hijo;
        ++cn;
        if (hijo == 0)
        {
            printf("Error en descripción, terminé mal c = %c
                   cn = %d \n", c, cn);
            sigue = 0;
            fin = 0;
            return;
        }
    }
} /* del else */
} /* del while */
} /* de disyun() */

```

```

/* _____
   SUBÁRBOL _____ */

```

```

subarbol( )
{
    sigue = 0;
    i = 0;
    while((ar[cn].dsc[i++] = dl[ap++]) != '\0')
        ; /* copiamos el resto de la línea */
    base = ar[cn].apun;
    c = '<';
    cn = ar[cn].apun;
    hijo = ar[cn].hijos;
    return;
}

```

```
}  
  
/* _____  
   COPIA  
   _____ */  
copia()  
{  
    sigue = 0;  
    i = 0;  
    ar[cn].arbol[0] = dl[ap];  
    while((ar[cn].desc[i++] = dl[ap++]) != '\0')  
        ; /* copiamos el resto de la línea */  
    c = '<';  
    cn = base;  
    hijo = ar[cn].hijos;  
    return;  
}
```

/*	Programa:	Indices.c
	Objetivo:	Este programa transforma a formato fijo subconjuntos de la información catalográfica.
	Datos de entrada:	El archivo de datos y la tabla en la que se representan los autómatas finitos conectados en la cual se han codificado los operadores semánticos elegidos en los registros agregados a los estados finales correspondientes.
	Datos de salida:	El índice generado.
	Método:	El programa guiado por la tabla para el reconocimiento del lenguaje MARC efectúa el análisis léxico de las fichas catalográficas y, cada vez que identifica una etiqueta o un código de subcampo aplica al contenido el operador semántico seleccionado.
		Descripción de los operadores semánticos:
		* Estado del autómata: este operador se utiliza para hacer el análisis léxico y permite detectar si hay algún error en las etiquetas de una unidad de información.
		%o Registro de conexión: este operador se aplica cuando se ha reconocido una etiqueta. Decodifica de la tabla los parámetros comunes a todos los subcampos de dicha etiqueta, específicamente el inicio absoluto y la longitud máxima en el registro de formato fijo en donde quedará el contenido del campo correspondiente a la etiqueta identificada. Controla las repeticiones de etiquetas. Hace la conexión con el autómata que reconoce los subcampos de la etiqueta.
		0 Omite: este operador se encuentra en registros agregados a estados finales. Lee —sin utilizar— la información hasta encontrar cualquiera de los caracteres “>” o “\$” que marcan el final del contenido.
		# Rastrea: este operador obtiene la información y la copia al registro de salida. Decodifica de la tabla el inicio y la longitud máxima relativos a la parte del registro de formato fijo correspondiente a la etiqueta, en donde se copia el contenido del subcampo identificado. Si la información es más larga omite el sobrante.

\$ Fin de ficha: escribe el registro en formato fijo al archivo de salida.

El programa se ejecuta de la siguiente manera:
indices.ex < arch-ent > < arbol > < arch-sal >

```

*/
#include < stdio.h >
#define MAXNOD 1300
#define TAMBUF 1000

FILE *fopen( ),*fpar,*fpd,*fpas;

struct nodo { /* La Tabla */
    char arbol[2]; /* operadores y caracteres por comparar */
    int hijos; /* número de hijos */
    int apun; /* apuntador a primer hijo */
    int par1,par2; /* parámetros para otros operadores */
    int rep; /* número de veces que ha aparecido la etq. */
} ar[MAXNOD];

int cn; /* nodo actual */
int hijo; /* número de hijos actual */
int base; /* base del subárbol */
int c;
int nofin;
int inietq,tametq; /* inicio y tamaño de etiqueta */
char buf[TAMBUF]; /* buffer de salida */
int i;

extern int finsubc( );
extern int finetq( );

main(argc,argv)
int argc;
char *argv[ ];
{
    if (argc!= 4)
    {
        printf("Faltan o sobran nombres de archivos \n");
        printf("Dar el nombre del archivo de datos, el árbol y el
            índice \n");
    }
    else
    {
        if ((fpd = fopen(*++ argv,"r"))= = NULL)
            printf("No existe ese archivo de datos \n");
        else
        {

```

```

if ((fpar = fopen(*++argv, "r")) == NULL)
    printf("No existe ese árbol\n");
else
    {
        fpas = fopen(*++argv, "w");
        for (i=0; i<MAXNOD; ++i)
            ar[i].arbol[0] = ' ';
    }

```

```

/* LECTURA DEL ÁRBOL */

```

```

cn = 1;
while (fscanf(fpar, "%1s", &c) != EOF)
    {
        switch (c)
            {
                case '*':
                    ar[cn].arbol[0] = c;
                    fscanf(fpar, "%c%2d%4d", &ar[cn].arbol[1], &ar[cn].hijos,
                        &ar[cn].apun);
                    break;
                case '$':
                    ar[cn].arbol[0] = c;
                    break;
                case '0':
                    ar[cn].arbol[0] = c;
                    break;
                case '%': /* operador, nueva base, inicio de etiqueta,
                            tamaño de etiqueta */
                    ar[cn].arbol[0] = c;
                    fscanf(fpar, "%4d%3d%3d%2d\n", &ar[cn].apun,
                        &ar[cn].par1, &ar[cn].par2, &ar[cn].hijos);
                    break;
                case '#': /* operador, inicio dentro de etiqueta, máximo */
                    ar[cn].arbol[0] = c;
                    fscanf(fpar, "%3d%3d", &ar[cn].par1, &ar[cn].par2);
                    break;
            } /* del switch */

        ++cn;
    } /* de while */

```

```

/* ESCRITURA DEL ÁRBOL */

```

```

cn = 1;
while (cn <= MAXNOD)
    {
        switch (ar[cn].arbol[0])
            {

```

```

case '*':
    printf(“%d:%c%c%c%2d%4d\n”, cn, ar[cn].arbol[0],
        ar[cn].arbol[1], ar[cn].hijos, ar[cn].apun);
    break;
case '$':
case '0':
    printf(“%d:%c\n”, cn, ar[cn].arbol[0]);
    break;
case '%':
    /* operador, nueva base, inicio de etiqueta, tamaño de
       etiqueta */
    printf(“%d:%c%4d%3d%3d%2d\n”, cn, ar[cn].arbol[0],
        ar[cn].apun, ar[cn].par1, ar[cn].par2, ar[cn].hijos);
    break;
case '#':
    /* operador, inicio dentro de etiqueta, máximo */
    printf(“%d:%c%3d%3d\n”, cn, ar[cn].arbol[0],
        ar[cn].par1, ar[cn].par2);
    break;
} /* del switch */
++cn;
} /* de while */

```

```

/* 

|                                                      |
|------------------------------------------------------|
| RASTREO DE LA INFORMACIÓN<br>Y GENERACIÓN DEL ÍNDICE |
|------------------------------------------------------|

 */

```

```

for(i=0; i< TAMBUF; ++i) buf[i]=' ';
for(i=0; i< MAXNOD; ++i) ar[cn].rep=0;
nofin = 1;
finetq( );
while (nofin)
{
    switch (ar[cn].arbol[0])
    {
        case '*':
            disyun( );
            break;
        case '%':
            subarbol( );
            break;
        case '0':
            omite( );
            break;
        case '#':
            rastrea( );
            break;
    }
}

```

```

    case '$':
        finfi();
        break;
    }; /* del switch */
} /* del while */

        close(fpd);
        close(fpar);
        close(fpas);
    }
}
}
printf("Fin, terminé bien\n");
} /* fin de main */

/* 

|                       |
|-----------------------|
| OPERADORES SEMÁNTICOS |
|-----------------------|

 */

/* DISYUNCIÓN */
disyun( )
{
    while (hijo > 0)
    {
        if (c == '\n') c = getc(fpd);
        if (c == EOF)
        {
            printf("Error en los datos, terminé mal, fin de archivo
            inesperado\n");
            nofin = 0;
            return;
        }
        else
        {
            if (c == ar[cn].arbol[1]) /* carácter actual igual a carácter del
            nodo */
            {
                hijo = ar[cn].hijos;
                cn = ar[cn].apun;
                c = getc(fpd); /* obtengo el siguiente carácter de los datos
                */
            }
            return;
        }
        else
        {
            --hijo;
        }
    }
}

```

```

    ++cn;
    if (hijo == 0)
    {
        printf("Error en datos, terminé mal c = %c
              cn = %d\n",c,cn);
        nofin = 0;
        return;
    }
}
} /* del else */
} /* del while */
} /* de disyun( ) */

/* -----
   SUBÁRBOL ----- */
subarbol( )
{
    if (ar[cn].rep == ar[cn].hijos) /* más repeticiones de las
                                    permitidas */
    {
        printf("Más repeticiones de las permitidas\n");
        while (((c = getc(fpd)) != '\n') && (c != EOF))
            ; /* saltamos toda la etiqueta */
        c = getc(fpd);
        finetq( );
    }
    else
    {
        base = ar[cn].apun;
        getc(fpd); getc(fpd); /* salto indicadores */
        inietq = ar[cn].par1;
        tametq = ar[cn].par2;
        inietq + = (ar[cn].rep*tametq);
        ar[cn].rep + = 1;
        cn = base;
        c = '<';
        hijo = ar[cn].hijos;
    }
    return;
}

/* -----
   OMITE ----- */
omite( )
{
    int f;
    f = 1;

```

```

while (f)
  switch(c)
  {
  case 'y':
    f = finetq( );
    break;
  case '$':
    f = finsubc( );
    break;
  case EOF:
    f = 0;
    nofin = 0;
    printf("Error, fin de archivo inesperado \n");
    break;
  default:
    c =getc(fpd);
    break;
  };
return;
}

/* _____
   RASTREA _____ */
rastrea( )
{
  int col,max,f;
  /* inicio relativo dentro de etiqueta */
  col = inietq + ar[cn].par1;
  max = col + ar[cn].par2;
  f = 1;
  while (f)
  {
    switch(c)
    {
    case 'y':
      f = finetq( );
      break;
    case '$':
      f = finsubc( );
      break;
    case EOF:
      printf("Error, fin de archivo inesperado \n");
      f = 0;
      nofin = 0;
      break;
    default:

```

```

if (col > max)
    while((c != ' ') && (c != '$') && (c != EOF))
        c = getc(fpd);
    else
    {
        buf[col++] = c;
        c = getc(fpd);
    }
    break;
} /* de switch */
} /* del while */
}

/* -----
   FIN DE FICHA */
finfi() /* escribimos el registro */
{
    for (i=0; i< TAMBUF; ++i) fprintf(fpas, "%c", buf[i]);
    fprintf(fpas, "\n");
    for (i=0; i< TAMBUF; ++i) buf[i] = ' ';
    for (i=1; i< MAXNOD; ++i) ar[i].rep = 0;
    finetq();
    return;
}

/* -----
   RUTINAS AUXILIARES */
int finctq()
{
    cn = 1;
    c = '<';
    hijo = ar[cn].hijos;
    return(0);
}
int finsubc()
{
    cn = ar[base].apun; /* ya estamos en el nodo de $ */
    hijo = ar[cn].hijos;
    return(0);
}

```

/*	<p>Programa: Creapun.c</p> <p>Objetivo: Este programa genera un archivo con los números de control Colmex de las fichas y sus respectivos apuntadores al principio y al final de la ficha.</p> <p>Datos de entrada: El archivo de datos que se quiere indexar. Además, un archivo llamado "arbolnum" que contiene la tabla en la que se representan los autómatas finitos conectados en la cual se han codificado los operadores semánticos elegidos en los registros agregados a los estados finales correspondientes.</p> <p>Datos de salida: Un archivo con los números de ficha y los apuntadores.</p> <p>Método: El programa guiado por la tabla para el reconocimiento del lenguaje MARC efectúa el análisis léxico de las fichas catalográficas y, cada vez que identifica una etiqueta o un código de subcampo aplica al contenido un operador semántico. Se lleva control del avance en la lectura de la ficha para obtener apuntadores a su inicio y su fin en el archivo de datos. Mediante esos apuntadores se efectuará la posterior recuperación de la ficha.</p> <p>Descripción de los operadores semánticos:</p> <ul style="list-style-type: none"> * Estado del autómata: este operador se utiliza para hacer el análisis léxico y permite detectar si hay algún error en las etiquetas de una unidad de información. % Registro de conexión: este operador se aplica cuando se ha reconocido una etiqueta. Hace la conexión con el autómata que reconoce los subcampos de la etiqueta. O Omite: este operador se encuentra en todos los registros agregados a estados finales excepto en el correspondiente a la etiqueta FIN) y al subcampo \$a de la etiqueta 035). Lee —sin utilizar— la información hasta encontrar cualquiera de los caracteres ">" o "\$" que marcan el final del contenido. # Rastrea: este operador se encuentra codificado únicamente en el registro agregado al estado final que reconoce el subcampo \$a de la etiqueta 035). Copia al registro de salida el contenido de este subcampo.
----	---

§ Fin de ficha: escribe en el archivo de salida el número de control Colmex y los apuntadores al inicio y al fin de la ficha en el archivo de datos.

El programa se ejecuta de la siguiente manera:
creapun.ex <arch-ent> <arch-sal>

```

#include <stdio.h>
#define MAXNOD 1300
#define TAMBUF 7

FILE *fopen( ), *fpar, *fpent, *fpsal;

struct nodo    {    /* La Tabla */
    char arbol[2]; /* operadores y caracteres por comparar */
    int hijos;     /* número de hijos */
    int apun;      /* apuntador a primer hijo */
                } ar[MAXNOD];

int cn;          /* nodo actual */
int hijo;        /* número de hijos actual */
int base;       /* base del subárbol */
int c;
int nofin;
int i;
long int inif, contcar;
char bufsal[TAMBUF];
char arch[11];

extern int finsubc( );
extern int finetq( );
extern copcad( );

main(argc, argv)
int argc;
char *argv[ ];
{
    if (argc != 3)
    {
        printf("Faltan o sobran nombres de archivos \n");
        printf("Dar el nombre del archivo de entrada y el de
            salida \n");
    }
    else
    {
        if ((fpent = fopen(argv[1], "r")) == NULL)
            printf("No existe ese archivo de entrada \n");
        else
        {

```

```

fpsal = fopen(argv[2], "w");
fpar = fopen("arbolnum", "r");
copcad(arch, argv[1]);
for (i = 0; i < MAXNOD; ++i)
    ar[i].arbol[0] = ' ';

```

```

/* LECTURA DEL ÁRBOL */

```

```

cn = 1;
while (fscanf(fpar, "%1s", &c) != EOF)
{
    switch (c)
    {
        case '*':
            ar[cn].arbol[0] = c;
            fscanf(fpar, "%c%2d%4d", &ar[cn].arbol[1], &ar[cn].hijos,
                &ar[cn].apun);
            break;
        case '$':
            ar[cn].arbol[0] = c;
            break;
        case '0':
            ar[cn].arbol[0] = c;
            break;
        case '%':
            /* operador, nueva base, número de hijos */
            ar[cn].arbol[0] = c;
            fscanf(fpar, "%4d%2d\n", &ar[cn].apun, &ar[cn].hijos);
            break;
        case '#':
            /* operador */
            ar[cn].arbol[0] = c;
            break;
    } /* del switch */
    ++cn;
} /* del while */

```

```

/* ESCRITURA DEL ÁRBOL */

```

```

cn = 1;
while (cn <= MAXNOD)
{
    switch (ar[cn].arbol[0])
    {
        case '*':

```

```

printf(“%d:%c%c%c%2d%4d\n”, cn, ar[cn].arbol[0],
      ar[cn].arbol[1], ar[cn].hijos, ar[cn].apun);
break;

case ‘$’:
case ‘0’:
case ‘#’:
    printf(“%d:%c\n”, cn, ar[cn].arbol[0]);
    break;
case ‘%’:
    /*operador, nueva base, inicio y tamaño de etiqueta*/
    printf(“%d:%c%c%4d%2d\n”, cn, ar[cn].arbol[0], ar[cn].apun,
          ar[cn].hijos);
    break;
case ‘#’:
    /* operador, inicio dentro de etiqueta, máximo */
    printf(“%d:%c\n”, cn, ar[cn].arbol[0]);
    break;
} /* del switch */
++cn;
} /* del while */

```

```

/* 

|                           |
|---------------------------|
| RASTREO DE LA INFORMACIÓN |
|---------------------------|

 */

```

```

nofin = 1;
contcar = 0;
inif = 0;
for (i=0; i< TAMBUF; ++i) bufasal[i] = ‘ ’;
finctq( );
while (nofin)
{
    switch (ar[cn].arbol[0])
    {
        case ‘*’:
            disyun( );
            break;
        case ‘%’:
            subarbol( );
            break;
        case ‘0’:
            omite( );
            break;
        case ‘#’:
            rastrea( );
            break;
        case ‘$’:

```

```

    finfi();
    break;
}; /* del switch */
} /* del while */

```

```

        close(fpent);
        close(fpar);
        close(fpsal);
    }
}
printf("Fin, terminé bien\n");
} /* fin de main */

```

```

/* OPERADORES SEMÁNTICOS */

```

```

/* DISYUNCIÓN */

```

```

disyun()
{
    while (hijo > 0)
    {
        if (c == '\n') { c = getc(fpent); ++contcar; }
        if (c == EOF)
        {
            nofin = 0;
            return;
        }
        else
        {
            if (c == ar[cn].arbol[1]) /* carácter actual igual a carácter
                del nodo */
            {
                hijo = ar[cn].hijos;
                cn = ar[cn].apun;
                c = getc(fpent);
                ++contcar; /* obtengo el siguiente carácter de los
                    datos */
                return;
            }
            else
            {
                --hijo;
                ++cn;
                if (hijo == 0)

```

```

    {
        printf("Error en los datos, terminé mal c=%c
              cn=%d\n",c,cn);
        nofin=0;
        return;
    }
} /* del else */
} /* del while */
} /* de disyun( ) */

```

```

/* -----
   SUBÁRBOL ----- */

```

```

subarbol( )
{
    base = ar[cn].apun;
    c = getc(fpent); ++contcar;
    c = getc(fpent); ++contcar; /* salto indicadores */
    cn = base;
    c = '<';
    hijo = ar[cn].hijos;
    return;
}

```

```

/* -----
   OMITE ----- */

```

```

omite( )
{
    int f;
    f = 1;
    while (f)
        switch (c)
        {
            case '>':
                f = finetq( );
                break;
            case '$':
                f = finsubc( );
                break;
            case EOF:
                f = 0;
                nofin = 0;
                printf("Error, fin de archivo inesperado\n");
                break;
            default:
                c = getc(fpent); ++contcar;
        }
}

```

```

        break;
    };
    return;
}

/* -----
   RASTREA */
rastrea( )
{
    int f,ap=0;
    f = 1;
    while (f)
    {
        switch (c)
        {
            case '>':
                f = finetq( );
                break;
            case '$':
                f = finsubc( );
                break;
            case EOF:
                printf("Error, fin de archivo inesperado\n");
                f = 0;
                nofin = 0;
                break;
            default:
                bufstal[ap++] = c;
                c = getc(fpent);
                ++contcar;
                break;
        } /* del switch */
    } /* del while */
}

/* -----
   FIN DE FICHA */
finfi( ) /* escribimos el registro */
{
    for (i = 1; i < TAMBUF; ++i) fprintf(fpsal, "%c", bufstal[i]);
    /* ojo escribo desde 1 porque no quiero el blanco */
    fprintf(fpsal, "%09ld %09ld %0s\n", inif, contcar-1, arch);
    inif = contcar;
    for (i = 0; i < TAMBUF; ++i) bufstal[i] = ' ';
    finetq( );
    return;
}

```

```

}
/* -----
                RUTINAS AUXILIARES
----- */
int finetq()
{
    cn = 1;
    c = '<';
    hijo = ar[cn].hijos;
    return(0);
}
int finsubc()
{
    cn = ar[base].apun; /* ya estamos en el nodo de $ */
    hijo = ar[cn].hijos;
    return(0);
}
copcad(s,t) /* copia la cadena t a la cadena s */
char *s, *t;
{
    while (*s++ = *t++);
}

```

```

/* Programa:          Recupera.c
Objetivo:            Con este programa se recupera la información
                    completa de las fichas seleccionadas.

Datos de entrada:   El archivo original con los datos completos. Ade-
                    más, el archivo con los apuntadores al inicio y al
                    fin de las fichas que se desean recuperar.

Datos de salida:    El archivo con la información completa.

El programa se ejecuta de la siguiente manera:
                    recupera.ex <arch-ent> <arch-sal> < <arch-apun>
*/

```

```

#include <stdio.h>
main(argc,argv)
int argc;
char *argv[ ];
{
    int fp1;
    FILE *fopen( ), *fp2;
    int pri,i,n,m;
    long int posi,posf;
    char c,erro;
    char buf[1536];

    fp1 = open(argv[1],0); /* archivo de datos */
    fp2 = fopen(argv[2],"w"); /* archivo de salida con las fichas
                               sin error */

    pri = 0;
    while ((erro = getchar( )) != EOF)
    {
        if (erro == '1' && pri == 0)
        {
            close(fp2);
            fp2 = fopen("malas","w");
            pri = 1;
        }
        /* putchar(erro); */
        posi = posf = 0;
        for (i = 0; i < 1535; ++i) buf[i] = ' ';
        for (i = 1; i < 60; ++i)
            { c = getchar( ); /* putchar(c); */ };
        while ((c = getchar( )) != ' ') posi = posi*10 + (c-'0');
        while ((c = getchar( )) != '\n') posf = posf*10 + (c-'0');
        /* printf("%ld %ld",posi,posf); */
    }
}

```

```
lseek(fp1, posi, 0);
m = posf - posi + 1;
/* printf("m: %5d", m); */
n = read(fp1, buf, m);
if (n != m)
{
    printf("\nERROR AL LEER CARACTERES\n");
    printf("n: %5d , %5d\n", n, (posf - posi + 1));
    for (i = 0; i < n; ++i) printf("%c", buf[i]);
    printf("\n\n");
};
for (i = 0; i < n; ++i) putc(buf[i], fp2);
};
}
```


ANEXO B

EJEMPLOS DE ALGUNOS PRODUCTOS ELABORADOS CON PROGRAMAS QUE SIGUEN LA METODOLOGÍA DE PROGRAMACIÓN DESCRITA EN ESTE LIBRO E INCLUYEN OPERADORES SEMÁNTICOS COMPLEJOS

1. Juego de tarjetas catalográficas	93
2. Etiquetas para el marcaje del libro	94
3. Lista de obras catalogadas	95
4. Índice de la lista de obras catalogadas	96
5. Registros MARC	97

Juego de tarjetas catalográficas

f 320.98 W9276 no.54		PUB
	Dunkerley, James. The crisis in Bolivia [by]...and Rolando Morales. Notre Dame, Ind., Helen Kellogg Institute for International Studies, University of Notre Dame, 1985. 55 p. 28 cm. (Working paper, 54)	
	1. Bolivia-Politica y gobierno-1982- Rolando, coaut. II. titulo. III. Serie.	I. Morales,
	○	
JVS/jvg		COLMEX/031850

f 320.98 W9276 no.54	The crisis in Bolivia	
	Dunkerley, James. The crisis in Bolivia [by]...and Rolando Morales. Notre Dame, Ind., Helen Kellogg Institute for International Studies, University of Notre Dame, 1985. 55 p. 28 cm. (Working paper, 54)	
	1. Bolivia-Politica y gobierno-1982- Rolando, coaut. II. titulo. III. Serie.	I. Morales,
	○	
JVS/jvg		COLMEX/031850

f 320.98 W9276 no.54	BOLIVIA-POLITICA Y GOBIERNO- 1982-	
	Dunkerley, James. The crisis in Bolivia [by]...and Rolando Morales. Notre Dame, Ind., Helen Kellogg Institute for International Studies, University of Notre Dame, 1985. 55 p. 28 cm. (Working paper, 54)	
	1. Bolivia-Politica y gobierno-1982- Rolando, coaut. II. titulo. III. Serie.	I. Morales,
	○	
JVS/jvg		COLMEX/031850

I. 320.98 W9276 no. 54	Morales, Rolando, coaut.
	Dunckerley, James. The crisis in Bolivia [by]... and Rolando Morales. Notre Dame, Ind., Helen Kellogg Institute for International Studies, University of Notre Dame, 1985. 55 p. 28 cm. (Working paper, 54)
	I. Bolivia-Política y gobierno-1982- Rolando, coaut. II. Título. III. Serie. . I. Morales,
JVS/iva	COLMEX/031850

Etiquetas para el marcaje del libro

295644
346.1/M6113n/ Baja California México Nacimientos...

295644
346.1/M6113n/ Baja California México Nacimientos...

346.1 M6113n Baja Califo

Lista de obras catalogadas

- (059) 051
K277
1979
no 3
sobr
Klare, Michel T., 1942-
South Africa's U.S. weapons connections.
(Washington) Institute for Policy Studies
(1979?)
(4) p. illus. 28 cm.
Sobretiro de: The Nation, July 28-August 4,
1979.
A la cabeza de la portada: Rne corporate
sunrunners.
COLMEI/050793
- (060) f
070.48
P446
COLMEI/009184
El periodismo rural en Africa. Por Paul Ansah (y
otros). Paris, Organización de las
Naciones Unidas para la Educación, la
Ciencia y la Cultura, 1981.
31 p. 27 cm.
- (061) f
070.48
R948
COLMEI/053122
Rural journalism in Africa. By Paul Ansah (et
al.) Paris, UNESCO (1981)
35 p. 27 cm. (Reports and papers on mass
communication, 88)
- (062) 079.6
R3131n
COLMEI/052995
Harris, Phil.
La información sobre Africa austral; cómo
informan desde el sur de Africa las agencias
occidentales de noticias (Traducción de Homero
Arista) (Barcelona) (Serbal) (1984)
188 p. 19 cm. (Colección de temas
africanos, 19)
Coed. con UNESCO.
- (063) 079.6
R9747p
COLMEI/050898
Ruth Sloan Associates, Washington, D.C.
The press in Africa. Edited by Helen Kitchen.
Washington, 1985.
56 p. 29 cm.
- (064) f
079.6511
L627n
COLMEI/052318
Leselbaum, Charles.
Notes sur "El Correo Espanol" de Orán.
Paris, Hispaniques, 1975.
19 p. 24 cm.
Sobretiro de: Mélanges offerts à Charles
Vincent Aubrun.
- (065) 092.1
824
1963
v.86
COLMEI/052717
Wilson, John Albert, 1899-
La cultura egipcia. Tr. de Florentino M.
Torner. México, Fondo de Cultura
Económica (1963)
483 p. 17.5 cm. (Breviarios del Fondo de
Cultura Económica, 85)
- (066) 092.1
824
v.165
COLMEI/052719
Pauline, Denise.
Las esculturas del Africa negra (Tr. de
Francisco González Aránguiz) México, Fondo
de Cultura Económica (1962)
171 p. illus. 17.5 cm. (Breviarios del
Fondo de Cultura Económica, 165)
- (067) 092.1
824
v.86
COLMEI/052716
Wilson, John Albert, 1899-
La cultura egipcia (Tr. de Florentino M.
Torner) México, Fondo de Cultura Económica
(1963)
483 p. illus. 17.5 cm. (Breviarios del
Fondo de Cultura Económica, 86)
- (068) 092.1
63
v.1604
COLMEI/050214
Déjeux, Jean.
La littérature Algérienne contemporaine.
Paris, Universitaires de France (1975)
126 p. 18 cm. (Que sais-je?, 1604)
- (069) 092.1
45
v.241
COLMEI/052721
Labouret, Henri, 1878-
Histoire des noirs d'Afrique. Paris, Presses
Universitaires de France, 1950.
126 p. mapa. 17.5 cm. (Que sais-je, 241)
- (070) 092.1
63
v.241a
COLMEI/052722
Labouret, Henri, 1878-
L'Afrique précoloniale. Paris, Presses
Universitaires de France, 1959.
126 p. mapas. 17.5 cm. (Que Sais-Je? -
Le point des connaissances actuelles, No. 241)
Bibliografía: p. 1125-1126.
- (071) 092.1
63
v.318
COLMEI/052723
Klein, Jacques.
La Tunisie. Paris, Presses Universitaires de
France, 1949.
129 p. mapas, diagrs. 17.5 cm. (Que
sais-Je? - Le point des connaissances actuelles,
318)
- (072) 092.1
63
v.4
COLMEI/052720
Julien, Charles-André, 1891-
Histoire de l'Afrique. Ed. refondu et mise a
jour. Paris, Presses Universitaires de France,
1963.
136 p. mapas. 17.5 cm. (Que Sais-Je? -
Le point des connaissances actuelles, 4)
Bibliografía: p. 134-135.

Índice de la lista de obras catalogadas

INDICE

- Teherán. Conferencia, 1943) 2762
 Teledifusión - África) 1818
 Textiles - África Oriental - Etiopía) 1936
 Teridofitas - África Ecuatorial Francesa) 3239
 Tesis y disertaciones académicas - Canadá -
 Bibliografía) 0031
 Tesis y disertaciones académicas - EE. UU. -
 Bibliografía) 0031
 Textiles - Industria y telas - India) 1826
 Texto, Libros de - África del Sur) 1766
 Thiers, Adolphe, 1797-1877) 0824
 Thonga (Tribu)) 3390
 Tibbu (Pueblo africano)) 3282
 Tierra - Tenencia, Derecho Vivienda - África del Sur -
 Botswana) 1247
 Tierra - Tenencia, Derecho de - África Central -
 Camerún) 1251
 Tierra - Tenencia, Derecho de - África Occidental -
 Ghana) 1400
 Tierra - Tenencia, Derecho de - África Occidental -
 Nigeria) 1241
 Tierra - Tenencia, Derecho de - África Oriental -
 Etiopía) 1250, 1252
 Tierra - Tenencia, Derecho de - África Oriental) 3300
 Tierra - Tenencia, Derecho de - África del Sur) 3349
 Tierra - Tenencia, Derecho de - África, Sur del Sahara) 1253
 Tierra - Tenencia, Derecho de - África) 1254, 1256
 Tierra - Tenencia, Derecho de - Babilonia) 2754
 Tierra - Tenencia, Derecho de - Tanganyika) 1705
 Tierra - Tenencia, Derecho de - África) 1255
 Tierra - Tenencia, derecho de - África Oriental -
 Etiopía) 1246
 Tierra - Tenencia, Derecho de - África Oriental -
 Etiopía) 1249
 Tierra, Tenencia - África del Norte - Egipto) 1264
 Tierra, Tenencia - Asia - Iraq) 1264
 Tierra, Tenencia - Asia - Siria) 1264
 Tierra, Tenencia - Derecho de - Cercano Oriente) 1264
 Tierra, Tenencia - Oriente (Cercano Oriente)) 1264
 Tierra, Tenencia - Países árabes) 1264
 Tierra, Tenencia, Derecho de - África del Norte -
 Marruecos) 1365
 Tierras Áridas - África Occidental) 1261
 Tizbaktu - Historia) 3157
 Tizbaktu - Vida social y costumbres) 2544
 Tizbaktu, Mall - Historia) 3154
 Tizbaktu, Sudán - Historia) 3163
 Tingad, Argelia - Descripción) 2075
 Timur, el Grande, 1336-1405) 2280
 Togo - Civilización) 2566
 Togo - Lengua) 3241
 Togo - Religión) 1982
 Togoland Francés - Historia) 3209
 Tolomeo, reyes de Egipto) 2719
 tong King - Historia) 3397
 Totemismo) 2252
 Touré, S' ekou, Pres. Guinea, 1922-) 0580
 Touré, Sékou, pres. Guinea - 1922-) 2539
 Touré, Sékou, Pres. Guinea, 1922-) 2766
 Trabajadores agrícolas) 1137
 Trabajadores agrícola - África del Norte - Liberia) 1369
 Trabajadores agrícolas - África Central) 1148
 Trabajadores agrícolas - África Occidental - Historia) 0001, 0002
 Trabajadores agrícolas - África Occidental) 1148
 Trabajadores agrícolas - África Oriental - Tanzania) 1088
 Trabajadores agrícolas - África del Norte - Argelia) 1147
 Trabajadores agrícolas - África del Norte - Liberia) 1368
 Trabajadores agrícolas - África del Norte - Sudán) 1356
 Trabajadores agrícolas - África del Norte - Tónex) 1147
 Trabajadores agrícolas - África, Sur del Sahara) 0711, 0712
 Trabajadores extranjeros - África Occidental Francesa -
 Costa de Marfil) 1155, 2529, 2536
 Trabajadores extranjeros - Europa - Francia) 1136
 Trabajadores extranjeros - Europa - Gran Bretaña) 1156
 Trabajadores extranjeros - Europa) 1154
 Trabajadores extranjeros africanos - Hispanoamérica -
 Jamaica) 1233
 Trabajadores extranjeros - E. U.) 1157
 Trabajadores y trabajadores - África Oriental - Tanzania
) 1089
 Trabajadores, Representación ante la gerencia - África
 del Norte - Argelia) 1016
 Trabajo - Leyes y Legislación - África) 1166
 Trabajo - Movilidad - África del Norte - Argelia) 1112
 Trabajo - Oferta - África Central - Angola - Historia) 0187
 Trabajo - Oferta - África Central - Camerún) 1120
 Trabajo - Oferta - África Oriental - Kenia) 1135
 Trabajo - Oferta - África Oriental - Tanzania) 0735
 Trabajo - Oferta - África del Norte - Argelia) 1136
 Trabajo - Oferta - África del Norte - Egipto) 1134, 1200
 Trabajo - Oferta - África del Norte - Sudán) 0177
 Trabajo - Oferta - África del Sur - Zambia) 1698
 Trabajo - Oferta - África) 1121, 1698
 Trabajo y trabajadores - África - Accra) 1130
 Trabajo y trabajadores - África - Alto Volta) 0402
 Trabajo y trabajadores - África - Congresos) 0394
 Trabajo y trabajadores - África - Discursos, ensayos,
 conferencias) 1160
 Trabajo y trabajadores - África Central - Camerún) 1120, 1126
 Trabajo y trabajadores - África Central - Rodesia del
 Norte) 1129

Registros MARC

007030000022001690004500008004100000035001300041040001500054
 092002200069100002200091245008800113260006300201300003500264
 440008600299690010700385710002800492901001300520>840830 1974
 us a eng >bb \$a 011101>bb \$a CM \$c CM>bb
 \$a 301.35 \$b H281p>10 \$a Harik, Iliya F->14 \$a The politica
 l mobilization of peasants \$b a study of an Egyptian
 community>0b \$a Bloomington, Ind. \$b Indiana University Pres
 s \$c [c1974]>bb \$a x, 309 p. \$b illus. \$c 24 cm.>b0 \$a Intern
 ational Development Research Center \$p Studies in Dev
 elopment \$v 8>b0 \$a Aldeas \$z Africa del Norte \$z Egipto>b0
 \$a Participacio@'n poli@'tica \$z Africa del Norte \$z Egipto
 >10 \$a Indiana \$b University>bb \$a 125930>&

006220000022001570004500008004100000035001300041040001500054
 092002200069100003300091245005600124260010900180300002300289
 440005900312691008000371901001300451>840830 1961 sp
 >bb \$a 011102>bb \$a CM \$c CM>bb \$a 301.35 \$
 b S676p>10 \$a Soares, Amadeu de Castilho>10 \$a Poli@'tica de
 bem-estar rural em Angola \$b Ensaio>0b \$a Lisboa \$b Junta d
 e Investigaco@~es do Ultramar. Centro de Estudos Poli
 @'ticos e Sociais \$c 1961>bb \$a 278 p. \$c 25 cm.>b0 \$a Estud
 os de Ciencias Poli@'ticas e Sociais \$v No. 49>b0 \$a Angola
 \$x Condiciones rurales>b0 \$a Africa Central \$x Condiciones
 rurales>bb \$a 093308>&

006440000022001690004500008004100000035001300041040001500054
 049001200069092002200081100003600103245006000139260007800199
 300002300277546002300300690013800323901001300461>840830 1975
 ua ara >bb \$a 011103>bb \$a CM \$c CM>bb
 \$a ARABE>bb \$a 301.36 \$b A122a>10 \$a [Al-'Abba@~di@~, 'Abd
 Alla@~h>10 \$a Anna@~t wa-nama@~di^@y al-mudun al-kubra fi-l-
 Suda@~n>0b \$a El Cairo \$b Instituto de Investigaciones y Est
 udios Arabes \$c 1975]>bb \$a 216 p. \$c 24 cm.>bb \$a Texto
 en a@'rabe>b0 \$a Ciudades y pueblos \$x Proyectos y urbaniza
 cio@'n \$z Africa del Norte \$z Suda@'n>b0 \$a Urbanizacio@'n \$
 z Africa del Norte \$z Suda@'n>bb \$a 159710>&

007560000022001690004500008004100000035001300041040001500054
 092002100069100003000090245004500120260013300165300002300298
 6900005500321691015700376710004000533901001300573>840830 1958
 uk eng >bb \$a 011104>bb \$a CM \$c CM>bb
 \$a 301.36 \$b E64p>10 \$a Epstein, Arnold Leonard>10 \$a Polit
 ics in an urban African community>0b \$a [Manchester] \$b Publ
 ished on the Rhodeslivingstone Institute Northern Rho
 desia by Manchester University Press \$c [1958]>bb \$a 254 p.
 \$c 22 cm.>b0 \$a Ciudades y pueblos \$z Rodesia del Norte \$x C
 asos>b0 \$a Luanshya, Rodesia del Norte >b0 \$a Rodesia del No
 rte \$z Razas nativas>b0 \$a Africa Central \$x Razas nativas>b

ANEXO C

**FORMATO MARCOLMEX
BIBLIOTECA "DANIEL COSÍO VILLEGAS"
EL COLEGIO DE MÉXICO**

Documento de circulación interna

Líder y directorio

<i>Posición dentro del líder o directorio</i>	<i>Código del contenido</i>	<i>Descripción del campo y del código del contenido</i>
Líder		
00-04		Longitud lógica del registro
05		Estatus del registro
	c	Registro revisado o corregido
	d	Registro dado de baja
	n	Registro nuevo
06		Tipo de registro
	a	Lenguaje impreso
	b	Lenguaje manuscrito
	c	Música impresa
	d	Música manuscrita
	e	Mapa impreso
	f	Mapa manuscrito
	g	Audiovisuales (excepto i, j, n, o)
	i	Grabaciones de sonido no musical
	j	Grabaciones de sonido musical
	n	Materiales audiovisuales de enseñanza
	o	Juegos audiovisuales (Kits)
07		Nivel bibliográfico
	a	Analítica
	c	Colección
	m	Monografía
	s	Publicación seriada (serials)
08-09	b	Blancos
10	2	Conteo de los indicadores de campo
11	2	Conteo de los indicadores de subcampo
12-16		Base de la dirección de cada registro
17		Nivel de descripción bibliográfica del registro
	b	Registro completo (inspección física)
	1	Registro parcial (tomado de fichas)
18		Forma de la catalogación descriptiva
	b	Antes de AACR1 revisadas
	a	AACR2
	i	ACCRI revisadas (ISBD completo)
	p	AACR1 revisadas (ISBD parcial)
19	b	Blanco
20	4	Número de caracteres de la longitud del campo

<i>Posición dentro del líder</i>	<i>Código del contenido</i>	<i>Descripción del campo y del código del contenido</i>
21	5	Número de caracteres de la posición del primer carácter en cada campo
22-23	00	
Directorio		
00-02		Número de etiqueta
03-06		Longitud del campo
07-11		Posición del primer carácter del campo

Campos fijos

<i>Etiqueta COLMEX</i>	<i>Posición dentro del campo</i>	<i>Código</i>	<i>Descripción del campo y del código del contenido</i>
FCAP	00-05		Fecha de entrada al archivo
FSPUB	06		Código de tipo de fecha de publicación
		c	Dos fechas: real y de copyright
		d	Fecha detallada con año, mes y día
		m	Dos fechas: de inicio y de terminación
		n	Fechas de publicación desconocidas
		q	Fecha dudosa
		r	Dos fechas: original y de reimpresión
		s	Fecha única (o probable)
F1	07-10		Fecha 1
F2	11-14		Fecha 2
PAIS	15-17		Código de país de publicación
ILUS	18-21		Código de ilustraciones
		b	Sin ilustraciones
		a	Ilustraciones
		b	Mapas
		h	Fascículos
		o	Fotografías
	22		Código intelectual
		b	No aplica
REPROD	23		Forma de la reproducción
		b	No es reproducción
		a	Microfilm
		b	Microficha
CONT	24-27		Código de contenido
		b	No especificado
		t	Reportes técnicos
		b	Bibliografías
		c	Catálogos
		i	Índices
		l	Legislación
		a	Resúmenes
		d	Diccionarios
		e	Enciclopedias
		r	Directorios
		y	Anuarios
		s	Estadísticas
		f	Manuales

<i>Etiqueta COLMEX</i>	<i>Posición dentro del campo</i>	<i>Código</i>	<i>Descripción del campo y del código del contenido</i>
PUBOF	28	o	Reseñas
		p	Textos programados
			Código de publicación oficial
		b	No oficial
		f	Federal/Nacional
		i	Organismos internacionales
		l	Municipio o ciudades
		o	Nivel no determinado
		s	Estatal
		u	De origen dudoso
CONF	29	z	Otros (oficiales)
			Indicador de conferencias
		0	No es una publicación de conferencias
EDCONM	30	l	Sí es una publicación de conferencias
			Indicador de Edición conmemorativa (festschrift)
IND	31	0	No es una edición conmemorativa
		l	Sí es una edición conmemorativa
AP	32		Indicador de asiento principal
		0	No es parte del registro
		l	Sí es parte del registro
BIO	33		Indicador de ficción
	34		Código biográfico
IDIOMA	35-37 38	b	No es material biográfico
		a	Autobiografía
		b	Biografía individual
		c	Biografía colectiva
		d	Contiene información biográfica
			Código de idioma
			Código de registros modificados
39	b	Registro idéntico a la tarjeta	
	r	El registro está romanizado y la tarjeta está parcialmente romanizada	
	o	El registro está romanizado y la tarjeta también	
	z	El registro no contiene los diacríticos	
		Fuente de catalogación	
	d	No es LC	

Campos de longitud variable

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
020	b b	\$a	Número Internacional Normalizado del Libro (ISBN)
035	b b	\$a	No. Control COLMEX
040	b b	\$a	Fuentes de la Catalogación
		\$b	Fuente original de catalogación
		\$c	Idioma de catalogación
		\$d	Agencia que introduce el registro
041	b		Agencia modificadora
	0		Idiomas
	1		Varios idiomas
		\$a	Traducción
043	b b	\$a	Idioma del texto o de su traducción
049	b b	\$a	Código del área geográfica
050	b b	\$a	Registro de colecciones
		\$a	Clasificación de la Biblioteca del Congreso de EUA (LCC)
		\$a	Número clasificador asignado por la Bib. del Congreso EUA
		\$b	Número de autor
080	b b	\$a	Clasificación Decimal Universal
082	b b	\$a	Clasificación Decimal Dewey (DDC) asignada por Bib. Congreso EUA
		\$a	Número clasificador asignado
092	b b	\$a	Clasificación Decimal Dewey (DDC) Colmex
		\$a	Número clasificador
		\$b	Número de autor
098	b b	\$a	Otros sistemas de clasificación
		\$a	Sistema de Naciones Unidas
099	b b	\$a	Clasificación de texto libre
		\$a	Primer elemento
		\$b	Segundo elemento
100			Asiento principal: Autor personal
	0		Nombre en orden directo
	1		Apellido simple
	2		Apellido compuesto
	3		Nombre de familia
	0		El asiento principal no es asiento de materia

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
	1		El asiento principal sí es asiento de materia
		\$a	Nombre del autor
		\$b	Numeración usada como elemento asociado al nombre
		\$c	Títulos u otras designaciones asociadas al nombre
		\$d	Fechas de nac. y/o muerte
		\$e	Términos que describen la relación de autoría
		\$f	Fecha de la obra
		\$k	Subasiento de forma
		\$l	Idioma de la obra
		\$n	Número de la parte o sección de la obra
		\$p	Nombre de la parte o sección de la obra
		\$q	Forma compelta del nombre (como adición)
		\$t	Título usado como subasiento del nombre
		\$w	Forma del asiento verificado según AACR2
110	0		Asiento principal: Autor corporativo Apellido [(por inversión) requerido sólo para asientos pre-AACR2]
	1		Lugar (jurisdicción)
	2		Nombre en orden directo
	0		El asiento principal no es asiento de materia
	1		El asiento principal sí es asiento de materia
		\$a	Nombre del autor corporativo
		\$b	Nombre de la entidad subordinada
		\$c	Lugar (Conferencia)
		\$d	Fecha (Conferencia)
		\$e	Términos que describen la relación de autoría
		\$f	Fecha de la obra
		\$g	Información adicional

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$k	Subasiento de forma
		\$l	Idioma de la obra
		\$n	Número de la parte o sección
		\$p	Nombre la parte o sección de la obra
		\$t	Título de la obra
		\$w	Forma del asiento verificada según AACR2
111			Asiento principal: Conferencia o Reunión
	0		Apellido [(por inversión) requerido sólo para asientos pre-AACR2]
	1		Lugar o lugar más nombre (requerido sólo para asientos pre-AACR2)
	2		Nombre en orden directo
	0		El asiento principal no es asiento de materia
	1		El asiento principal sí es asiento de materia
		\$a	Nombre de la conferencia, reunión, etc. (o, elemento geográfico)
		\$b	Número de la conferencia o reunión (utilizado exclusivamente para asientos pre-AACR2)
		\$c	Lugar de realización de la conferencia o reunión
		\$d	Fecha de la conferencia o reunión
		\$e	Nombre de la entidad subordinada a la conferencia o reunión
		\$f	Fecha de la obra
		\$k	Subasiento de forma
		\$l	Idioma
		\$n	Número de la conferencia o reunión; parte o sección
		\$p	Nombre de la parte o sección
		\$q	Nombre de la conferencia o reunión a continuación del elemento geográfico

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
130	0-9	\$w	Forma del asiento verificada según AACR2
			Asiento principal: Título uniforme
			Número de caracteres que no se tomarán en cuenta para la ordenación alfabética del asiento
		0	El asiento principal no es asiento de materia
		1	El asiento principal es asiento de materia
		\$a	Título o elemento básico del asiento de título
		\$d	Fecha de celebración de tratado
		\$f	Fecha de la obra
		\$g	Información adicional
		\$h	Designación general de material
		\$k	Subasiento de forma
		\$l	Idioma
		\$n	Número de la parte o sección de la obra
		\$p	Nombre de la parte o sección de la obra
\$s	Versión		
\$t	Título		
\$w	Forma del asiento verificada según AACR2		
240	b 0-9		Título uniforme
			Número de caracteres que no se tomarán en cuenta para la ordenación alfabética del asiento
		\$a	Título uniforme
		\$d	Fecha de celebración del tratado
		\$f	Fecha de edición de la obra
		\$g	Información adicional
		\$h	Designación general de material
		\$k	Subencabezamiento de forma
		\$l	Idioma
		\$m	Medio de ejecución
		\$n	Número de la parte o sección
\$p	Nombre de la parte o sección		

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$r	Clave
		\$s	Versión
		\$w	Formas de asiento verificado según AACR2
241	1		Título romanizado (Colmex) Sí se requiere asiento secundario de título
	0-9		Número de caracteres que no se tomarán en cuenta para la ordenación alfabética del asiento
		\$a	Título
		\$b	Subtítulo
		\$c	Mención de responsabilidad
245	0		Título propiamente dicho No se requiere asiento secundario de título
	1		Sí se requiere asiento secundario de título
	0-9		Número de caracteres que no se tomarán en cuenta para la ordenación alfabética del asiento
		\$a	Título propiamente dicho
		\$b	Subtítulo; otra información acerca del título
		\$c	Mención de responsabilidad
		\$h	Designación general de material (DMG)
		\$n	Número de la parte o sección
		\$p	Nombre de la parte o sección
250	b b		Mención de edición
		\$a	Mención de edición propiamente dicha
		\$b	Mención de responsabilidad relacionada con la edición
260	b		Pie de imprenta
	0		El editor no es el asiento principal
	1		El editor es el asiento principal
		\$a	Lugar de publicación
		\$b	Nombre del editor
		\$c	Fecha de publicación
300	b b		Descripción física
		\$a	Número de páginas o volúmenes

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$b	Mención de ilustración
		\$c	Tamaño
		\$e	Material acompañante
		\$f	Tipo de unidad
		\$g	Tamaño de unidad
400			Mención de serie: Autor personal/Título (sólo pre-AACR2)
	0		Nombre en orden directo
	1		Apellido simple
	2		Apellido compuesto
	3		Nombre de familia
	0		El elemento principal del asiento de la serie no está representado por un pronombre
	1		El elemento principal del asiento de la serie sí está representado por un pronombre ("su")
		\$a	Nombre del autor
		\$b	Numeración usada como elemento asociado al nombre
		\$c	Títulos u otras designaciones asociadas al nombre
		\$d	Fechas de nac. y/o muerte
		\$e	Términos que describen la relación de autoría
		\$f	Fecha de la obra
		\$k	Subasiento de forma
		\$l	Idioma de la obra
		\$n	Número de la parte o sección
		\$p	Nombre de la parte o sección
		\$q	Forma compelta del nombre
		\$t	Título de la serie usado como subasiento del nombre
		\$v	Volumen o número de la serie
		\$x	ISSN
		\$w	Forma del asiento verificado según AACR2
410			Mención de serie: Autor corporativo/Título (pre-AACR2)
	0		Apellido [(por inversión) requerido sólo para asientos pre-AACR2]
	1		Lugar (jurisdicción)

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
	2		Nombre en orden directo
	0		El elemento principal del asiento de la serie no está representado por un pronombre ("su")
	1		El elemento principal del asiento de la serie sí está representado por un pronombre ("su")
		\$a	Nombre del autor corporativo
		\$b	Nombre de la entidad subordinada
		\$c	Lugar (conferencia)
		\$d	Fecha (conferencia)
		\$e	Términos que describen la relación de autoría
		\$f	Fecha de la obra
		\$g	Información adicional
		\$k	Subasiento de forma
		\$l	Idioma de la obra
		\$n	Número de la parte o sección
		\$p	Nombre de la parte o sección
		\$t	Título de la serie usado como subasiento del nombre
		\$v	Volumen o número de la serie
		\$w	Forma del asiento verificado según AACR2
		\$x	ISSN
411			Mención de serie: Conferencia o Reunión/Título
	0		Apellido [(por inversión) requerido sólo para asientos pre-AACR2]
	1		Lugar o lugar más nombre (requerido sólo para asientos pre-AACR2)
	2		Nombre en orden directo
	0		El elemento principal del asiento de la serie no está representado por un pronombre ("su")
	1		El elemento principal del asiento de la serie sí está representado por un pronombre ("su")

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$a	Nombre de la conferencia, reunión, etc. (o, elemento geográfico)
		\$b	Número de la conferencia o reunión (utilizado exclusivamente para asientos pre-AACR2)
		\$c	Lugar de realización de la conferencia o reunión
		\$d	Fecha de la conferencia o reunión
		\$e	Nombre la entidad subordinada a la conferencia o reunión
		\$f	Fecha de la obra
		\$k	Subasiento de forma
		\$l	Idioma de la obra
		\$n	Número de la conferencia o reunión; parte o sección
		\$p	Nombre de la conferencia o reunión a continuación del elemento geográfico
		\$q	Nombre de la conferencia o reunión a continuación del elemento geográfico
		\$t	Título de la serie usado como subasiento del nombre de la conferencia o reunión
		\$v	Volumen o número de la serie
		\$x	ISSN
		\$w	Forma del asiento verificado según AACR2
440	b 0-9		Mención de serie: Título distintivo Número de caracteres que no se tomarán en cuenta, para ordenación alfabética del asiento
		\$a	Título
		\$n	Número de la parte o sección
		\$p	Nombre de la subserie
		\$v	Volumen o número de la serie
		\$x	ISSN

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
490	b		Mención de serie: Serie no consignada en el registro, o, serie consignada de manera diversa en el registro
	0		Serie no consignada en el registro
	1		Serie consignada de manera diversa en el registro
		\$a	Mención de serie (incluye mención de responsabilidad)
		\$l	Número de la Biblioteca del Congreso de E.U.
		\$n	Número de la Biblioteca Nacional de Canadá
		\$x	ISSN
		\$v	Volumen o número de la serie (la información de este campo no se utiliza para generar asientos secundarios de serie; sin embargo, cada vez que el valor del primer indicador sea "1" habrá una mención de serie correspondiente en 800-840, que dará la serie en la forma que será consignada en el registro)
500	b b	\$a	Nota general
501	b b	\$a	Nota "con"
502	b b	\$a	Nota de tesis académica
		\$a	Mención de tesis y grado académico
		\$b	Institución
503	b b	\$a	Nota de historia bibliográfica
504	b b	\$a	Nota de bibliografías
505	b	\$a	Nota de contenido
	0		Contenido completo
	1		Contenido incompleto
	2		Contenido parcial
507	b b	\$a	Nota de escala
520	b b	\$a	Nota de resumen, alcance, etc.
533	b b	\$a	Nota de fotorreproducción
		\$a	Tipo de reproducción
		\$b	Lugar de reproducción
		\$c	Agencia responsable

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
546	b b	\$a	Nota de idioma
590	b b	\$a	Nota local
690	b		Encabezamientos de materia:
			Tópicos
	0		No validados
	1		Kidder (mecanografiado)
	2		DPT (Kidder manuscrito)
	3		BNM (Escamilla)
	4		UP (Roviera)
	5		Consulta especialistas
	6		Otras fuentes
		\$a	Encabezamiento de materia
		\$b	Elemento geográfico adicional al nombre del lugar
		\$x	Subdivisión general
		\$y	Subdivisión cronológica
		\$z	Subdivisión geográfica
691	b		Encabezamiento de materia
			Geográfico
	0		No validados
	1		Kidder (mecanografiado)
	2		DPT (Kidder manuscrito)
	3		BNM (Escamilla)
	4		UP (Roviera)
	5		Consulta especialistas
	6		Otras fuentes
		\$a	Encabezamiento de materia
		\$b	Elemento geográfico adicional al nombre
		\$x	Subdivisión general
		\$y	Subdivisión cronológica
		\$z	Subdivisión geográfica
692	b		Encabezamiento de materia: Autor
			personal
	0		Nombre en orden directo
	1		Apellido simple
	2		Apellido múltiple
	3		Nombre de familia
		\$a	Nombre del autor personal
		\$b	Numeración usada como elemento del nombre

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$c	Títulos u otras designaciones asociadas al nombre
		\$d	Fecha de nacimiento y/o muerte
		\$e	Términos que describen la relación de autoría
		\$f	Fecha de la obra
		\$g	Información adicional
		\$h	Designación general de material
		\$k	Subasiento de forma
		\$l	Idioma de la obra
		\$m	Medio de ejecución
		\$n	Número de la parte o sección de la obra
		\$o	Mención de arreglo
		\$p	Nombre de la parte o sección
		\$q	Forma completa del nombre (como adición)
		\$r	Clave
		\$s	Versión
		\$t	Título usado como subasiento del nombre
		\$x	Subdivisión general
		\$y	Subdivisión cronológica
		\$z	Subdivisión geográfica
		\$w	Forma del asiento verificado según AACR-2
693	b		Encabezamiento de materia:
	0		Autor corporativo Apellido [(por inversión) requerido sólo para asientos pre-AACR2]
	1		Lugar (jurisdicción) o lugar más nombre (requerido sólo para asientos pre-AACR2)
	2		Nombre en orden directo
		\$a	Nombre del autor corporativo
		\$b	Unidad subordinada
		\$c	Lugar (conferencia)
		\$d	Fecha de la conferencia o año de tratado
		\$e	Términos que describen la relación de autoría
		\$f	Fecha de la obra

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$g	Información adicional
		\$h	Designación general de material
		\$k	Subasiento de forma
		\$l	Idioma
		\$m	Medio de ejecución
		\$n	Número de la parte o sección, o, número de conferencia
		\$o	Mención sobre arreglo
		\$p	Nombre de la parte o sección
		\$r	Clave
		\$s	Versión
		\$t	Título usado como subasiento del nombre
		\$x	Subdivisión general
		\$y	Subdivisión cronológica
		\$z	Subdivisión geográfica
		\$w	Forma del asiento verificado
694	b		Encabezamiento de materia:
	0		Conferencia, reunión, etc. Apellido (Por inversión) requerido sólo para asientos pre-AACR2
	1		Lugar o lugar más nombre (requerido sólo para asientos pre-AACR2)
	2		Nombre en orden directo
		\$a	Nombre de la conferencia, reunión, etc.
		\$b	Número de la conferencia o reunión (utilizado exclusivamente para asientos pre-AACR2)
		\$c	Lugar de realización de la conferencia o reunión
		\$d	Fecha de realización de la conferencia o reunión
		\$e	Nombre de la unidad subordinada
		\$g	Información adicional
		\$h	Designación general de material
		\$q	Nombre de la reunión a continuación de lugar

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$t	Título de la obra usado como subasiento
		\$x	Subdivisión general
		\$y	Subdivisión cronológica
		\$z	Subdivisión geográfica
		\$w	Forma del asiento verificada según AACR2
695	b		Encabezamiento de materia: Título uniforme
	0-9		Número de caracteres que no se tomarán en cuenta para la ordenación alfabética del asiento
		\$a	Título o elemento básico del asiento
		\$d	Fecha de tratado
		\$s	Versión
		\$f	Fecha de la obra
		\$h	Designación general de material
		\$k	Subasiento de forma
		\$l	Idioma de la obra
		\$n	Número de la parte o sección
		\$p	Nombre de la parte o sección
		\$t	Título de la obra
		\$x	Subdivisión general
		\$y	Subdivisión cronológica
		\$z	Subdivisión geográfica
		\$w	Forma del asiento verificada según AACR2
700			Asiento secundario: Autor personal
	0		Nombre en orden directo
	1		Apellido simple
	2		Apellido compuesto
	3		Nombre de familia
	0		Asiento secundario (Autor)
	1		Asiento secundario (Autor Título)
	2		Asiento analítico
		\$a	Nombre del autor
		\$b	Numeración usada como elemento asociado al nombre

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$c	Títulos u otras designaciones asociadas al nombre
		\$d	Fechas de nacimiento y/o muerte
		\$e	Términos que describen la relación de autoría
		\$f	Fecha de la obra
		\$h	Designación general de material
		\$k	Subasiento de forma
		\$l	Idioma de la obra
		\$m	Medio de ejecución
		\$n	Número de la parte o sección
		\$o	Mención sobre arreglo
		\$p	Nombre de la parte o sección
		\$q	Forma completa del nombre (como adición)
		\$t	Título usado como subasiento del nombre
		\$w	Forma de asiento verificado según AACR2
710			Asiento secundario: Autor corporativo
	0		Apellido [(por inversión) requerido sólo para asientos pre-AACR2]
	1		Lugar (jurisdicción) o lugar más nombre (requerido sólo para asientos pre-AACR2)
	2		Nombre en orden directo
	0		Asiento secundario (Autor)
	1		Asiento secundario (Autor-Título)
	2		Asiento analítico
		\$a	Nombre del autor corporativo
		\$b	Unidad subordinada
		\$c	Lugar (conferencia)
		\$d	Fecha de conferencia, o, año de tratado
		\$e	Términos que describen la relación de autoría
		\$f	Fecha de la obra
		\$g	Información adicional
		\$h	Designación general de material
		\$k	Subasiento de forma

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$l	Idioma de la obra
		\$m	Medio de ejecución
		\$n	Número de la parte o sección
		\$p	Nombre de la parte o sección
		\$o	Mención sobre arreglo
		\$r	Clave
		\$s	Versión
		\$t	Título de la obra
		\$w	Forma de asiento verificado según AACR2
711			Asiento secundario: Conferencia, reunión, etc.
	0		Apellido [(por inversión) requerido sólo para asientos pre-AACR2]
	1		Lugar (jurisdicción) o lugar más nombre (requerido sólo para pre-AACR2)
	2		Nombre en orden directo
	0		Asiento secundario (Autor)
	1		Asiento secundario (Autor-Título)
	2		Asiento analítico
		\$a	Nombre de la conferencia, reunión
		\$b	Número de conferencia o reunión
		\$c	Lugar de realización de la conferencia o reunión
		\$d	Fecha de realización de la conferencia o reunión
		\$e	Nombre de la unidad subordinada
		\$f	Fecha de la obra
		\$h	Designación general de material
		\$k	Subasiento de forma
		\$l	Idioma
		\$n	Número de la conferencia o reunión; parte o sección
		\$p	Nombre de la parte o sección
		\$q	Nombre de la reunión a continuación de lugar (requerido sólo para asientos pre-AACR2)
		\$s	Versión

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$t	Título de la obra usado como subasiento
		\$w	Forma del asiento verificado según AACR2
730			Asiento secundario: Título uniforme
	0-9		Número de caracteres que no se tomarán en cuenta en la ordenación alfabética del asiento
	0		Asiento secundario (Autor)
	1		Asiento secundario (Autor-Título)
	2		Asiento analítico
		\$a	Título o elemento básico del asiento
		\$d	Fecha de tratado
		\$f	Fecha de la obra
		\$g	Información adicional
		\$h	Designación general de material
		\$k	Subasiento de forma
		\$l	Idioma de la obra
		\$m	Medio de ejecución
		\$n	Número de la parte o sección
		\$o	Mención sobre arreglo
		\$p	Nombre de la parte o sección
		\$r	Clave
		\$s	Versión
		\$t	Título de la obra
		\$w	Forma del asiento verificado según AACR
740			Asiento secundario: Variante del título
	0-9		Número de caracteres que no se tomarán en cuenta en la ordenación alfabética del asiento
	1		Asiento secundario propiamente dicho
	2		Asiento analítico
		\$a	Variante del título
		\$b	Subtítulo; otra información acerca del título
		\$c	Mención de responsabilidad

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>	
800	b	\$h	Designación general de material	
		\$n	Número de la parte o sección	
		\$p	Nombre de la parte o sección	
			Asiento secundario de serie: Autor personal/Título	
		0	Nombre en orden directo	
		1	Apellido simple	
		2	Apellido compuesto	
		3	Nombre de familia	
			\$a	Nombre del autor
			\$b	Numeración usada como elemento asociado al nombre
			\$c	Títulos u otras designaciones asociadas al nombre
			\$d	Fechas de nacimiento y/o muerte
			\$e	Términos que describen la relación de autoría
			\$f	Fecha de la obra
			\$h	Designación general del material
			\$k	Subasiento de forma
			\$l	Idioma de la obra
			\$m	Medio de ejecución
			\$n	Número de la parte o sección de la obra
			\$p	Nombre de la parte o sección de la obra
			\$q	Forma completa del nombre
			\$s	Versión
			\$t	Título de la serie usado como subasiento del nombre
810	b	\$v	Volumen o número de la serie	
			Asiento secundario de serie: Autor corporativo/Título	
		0	Apellido [(por inversión) requerido sólo para asientos pre-AACR2]	
		1	Lugar (jurisdicción)	
		2	Nombre en orden directo	
			\$a	Nombre del autor corporativo
			\$b	Nombre de la unidad subordinada
			\$c	Lugar (Conferencia)
			\$d	Fecha

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$e	Términos que describen la relación de autoría
		\$f	Fecha de la obra
		\$g	Información adicional
		\$h	Designación gral. de material
		\$k	Subasiento de forma
		\$l	Idioma de la obra
		\$m	Medio de ejecución
		\$n	Número de la parte o sección de la obra
		\$p	Nombre de la parte o sección de la obra
		\$r	Clave
		\$s	Versión
		\$t	Título de la serie usado como subasiento del nombre
811	b	\$v	Volumen o número de la serie
			Asiento secundario de serie: Conferencia o Reunión/Título
	0		Apellido [(por inversión) requerido sólo para asientos pre-AACR2]
	1		Lugar (Jurisdicción)
	2		Nombre en orden directo
		\$a	Nombre de la conferencia, reunión, etc. (o elemento geográfico)
		\$b	Número de la conferencia, reunión (utilizado exclusivamente para asientos pre-AACR2)
		\$c	Lugar de realización de la conferencia o reunión
		\$d	Fecha de la conferencia o reunión
		\$e	Nombre de la entidad subordinada a la conferencia o reunión
		\$f	Fecha de la obra
		\$g	Información adicional
		\$h	Designación gral. de material
		\$k	Subasiento de forma

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
		\$l	Idioma de la obra
		\$n	Número de la parte o sección de la obra
		\$p	Nombre de la parte o sección de la obra
		\$q	Nombre de la reunión a continuación del elemento geográfico
		\$t	Título de la serie usado como subsiguiente del nombre
830	b	\$v	Volumen o número de la serie
	0-9		Asiento secundario de serie: Título uniforme Número de caracteres que no se tomarán en cuenta para la ordenación alfabética del asiento
		\$a	Título o elemento básico del asiento de título
		\$d	Fecha de celebración de tratado
		\$f	Fecha de la obra
		\$g	Información adicional
		\$h	Designación gral. de material
		\$k	Subsiguiente de forma
		\$l	Idioma de la obra
		\$m	Medio de ejecución
		\$n	Número de la parte o sección de la obra
		\$o	Arreglo
		\$p	Nombre de la parte o sección de la obra
		\$r	Clave
		\$s	Versión
		\$t	Título
840	b	\$v	Volumen o número de la serie
	0-9		Asiento secundario serie: Título (Pre-AACR2) Número de caracteres que no se tomarán en cuenta para la ordenación alfabética del asiento
		\$a	Título de la serie

<i>Etiqueta MARC</i>	<i>Valor de indicadores</i>	<i>Código de subcampo</i>	<i>Descripción del campo, del indicador o del código de subcampo</i>
901	♫ ♫	\$v	Volumen o número de la serie
902	♫ ♫	\$a	Número de adquisición
		\$a	Claves adicionales

BIBLIOGRAFÍA

- Aho, A. V. y Corasick, M. J. "Efficient string matching: An aid to bibliographic search" *Comm. ACM* 18:6 pp. 333-340, 1975.
- Aho, A. V. y Ullman, J. D. *Principles of compiler design*, Addison-Wesley, Reading Mass., 1977.
- . *The theory of parsing, traslation, and compiling*, Prentice-Hall, Englewood Cliffs, N. J., 1972.
- Avram, Henriette D. *MARC, its history and implications*, Washington, D. C.: Library of Congress, 1975.
- Bar-Hillel, Yehoshua. "A logician's reaction to theorizing on information search systems" pp. 313-329. En *Language and information. Selected essays on their theory and application*, Addison-Wesley, Reading Mass., 1964.
- . "Theoretical aspects of the mechanization of literature searching" pp. 330-364. En *Language and information. Selected essays on their theory and application*, Addison-Wesley, Reading Mass., 1964.
- El Colegio de México. Biblioteca "Daniel Cosío Villegas". *MARCOLMEX: Adaptación del formato MARC para El Colegio de México* [Documento de circulación interna], 1989.
- Encyclopedia of Library and Information Science*, Marcel Dekker, Inc. N.Y. Vol. 30 pp. 416-463, 1980.
- García Hidalgo, Ma. Isabel. "Metodología" En *Manejo automatizado de acervos documentales* Unidad de Computo de El Colegio de México, accésit CREI DE PLATA 1985.
- García Hidalgo, Ma. Isabel y Freidberg, Laura. "Procedimiento computacional para compatibilizar el sistema de la biblioteca de El Colegio de México con microsis", Segundo Congreso Nacional de Micro CDS/ ISIS, 1989.
- García Hidalgo, Ma. I., Tejeda, C. y Quijano, A. "El proyecto de automatización de la Biblioteca Daniel Cosío Villegas: Adquisiciones y Catalogación y Clasificación", *XV Jornadas Mexicanas de Biblioteconomía*, Tlaxcala, México, 1984.
- Hammer, D. *The Information Age; its development, its impact*, The Scarecrow Press, Inc. Metuchen, N. J. Scarecrow, 1976.
- Heaps, H. S. *Information retrieval; computational and theoretical aspects* Academic Press, San Francisco Calif. 1978.
- Hopcroft, J. E. y Ullman, J. D. *Introduction to automata theory, lenguajes, and computation*, Addison-Wesley, Reading, Mass. 1979.
- Kidder, Ione Marion, comp. *Encabezamientos de materia, castellano-inglés, inglés-castellano*, México: Biblioteca Benjamín Franklin, 1947. Versión mecanografiada.
- Malinconico, S. M. y Fasana, P. J. *The future of the catalog: the library's choices*,

Professional librarian series; N. Y.: Knowledge Industry Publications, 1979.

Wessel, A. *Computer-aided information retrieval*, Melville, Los Angeles Calif. 1975.

Woods, W. A. "Transition Network Grammars for Natural Language Analysis"
Comm. ACM 13:10 pp. 591-606, 1970.

Más allá de los manejadores de bases de datos,
se terminó de imprimir en la sección de
Reproducción de Documentos de El Colegio de México,
en enero de 1991. Se tiraron 300 ejemplares.
Cuidó la edición el Departamento
de Publicaciones de El Colegio de México.



UNIDAD DE CÓMPUTO

En este libro se presenta una solución computacional a un problema de recuperación de información bibliográfica. Esta solución, desarrollada por las autoras del libro, es una aplicación de un método general de tratamiento de información no numérica que surgió a partir de una serie de soluciones a problemas planteados en el ámbito de las ciencias sociales y las humanidades.

La metodología computacional descrita en esta obra incluye los siguientes aspectos: la acomodación de un esquema estructural a los datos que hay que analizar, el uso de un formato de longitud variable con identificadores y separadores de campos para almacenar la información, la aplicación del análisis léxico dirigido por una tabla en la que se representan autómatas conectados, y el diseño y programación de operadores semánticos.

Esta obra incluye además reflexiones sobre la complejidad de los procesos de recuperación de información y los componentes esenciales del sistema de cómputo (que son el corazón del método mencionado). A manera de anexos figuran algunos programas con el propósito de que el lector llegue a manejar el conocimiento necesario para aplicar este método a la resolución de sus propios problemas.

