LOS SISTEMAS INTEGRADOS DE GESTIÓN BIBLIOTECARIA

F. DE MOYA ANEGÓN

COLECCIÓN ESTUDIOS

LOS SISTEMAS INTEGRADOS DE GESTIÓN BIBLIOTECARIA

FÉLIX DE MOYA ANEGÓN

LOS SISTEMAS INTEGRADOS DE GESTIÓN BIBLIOTECARIA

Estructuras de datos y recuperación de información

FABAD

Moya Anegón, Félix de

Los sistemas integrados de gestión bibliotecaria : estructuras de datos y recuperación de información / Félix de Moya Anegón. — Madrid : ANABAD, cop. 1995. 227 p. — (Colección Estudios).

Bibliografía: p. 217-227.
Depósito legal: M-37333-1994
ISBN: 84-88716-15-X

1. Bibliotecas-Automatización. 2. Sistemas de información. I. Título. II. Serie: Colección Estudios (Asociación Española de Archiveros, Bibliotecarios, Museólogos y Documentalistas (España))

025 : 681.31

Cubierta: Graphica, S. A.

Realización: Editorial La Muralla, S. A. Constancia, 33. 28022 Madrid.

ISBN: 84-88716-15-X
Depósito legal: M-37333-1994
Printed in Spain. Impreso por Grafur, S. A. (Madrid).

A Luis, que me mira desde su quietud.

This hardly an indictment of a discipline [information science] that has existed for less than half a century and that finds its roots in very practical problems. It would, in fact, be surprising well-developed theories at a stage where the primary activity is gathering facts. Our facts have been gathered by persons with training in diverse disciplines and with diverse points of view. This is reflected in the borrowing of theories and models from other sources.

BERT R. BOYCE y DONALD H. KRAFT

One general point about theories and models [in information retrieval] should be made. While any retrieval system must be based on some theory of retrieval, such implicit theories are extremely difficult to extract or analyse. Even some explicitly formulated theories are formulated in such general terms, with such loose connection between the theory and the system desing, that they are still very difficult to evaluate. So the works [...] have a bias towards mathematical models, not because mathematics per se is necessarily a Good Thing, but because the setting up of a mathematical model generally presupposes a careful formal analysis of the problem and specification of the assumptions and explicit formulation of the way in which the model depends on the assumptions.

S. E. ROBERTSON

One of the attractive aspects of working with file structures is that designing them involves consideration of only a handful of basic concepts, yet there seems to be an almost infinite number of ways to combine and vary these concepts. Working on a storage and retrieval problem is always an opportunity to make something new. Along with the excitement of making new things, however, there is the pleasure that comes from working with fundamental conceptual tools that you understand well and have used many times before.

MICHAEL J. FOLK y BILL ZOELLICK

ÍNDICE

I

INTRODUCCIÓN

Las palabras contenidas en las tres citas iniciales pueden servir muy bien para introducir este trabajo. En ellas se alude a tres aspectos importantes de la investigación en el campo de la Documentación que de forma implícita están presentes en estas páginas. Por una parte, este trabajo se ocupa en gran medida de la constatación de hechos, por esta razón entiendo que resulta poco especulativo y menos teórico. Esto me ha obligado a seguir bastante al pie de la letra el planteamiento de la segunda cita, según el cual la débil conexión existente entre las teorías y los sistemas que se diseñan en nuestra disciplina hace necesaria la utilización de modelos matemáticos que nos aseguren, al menos, la corrección formal de los análisis realizados y de las conclusiones extraídas. Por último, la aplicación de estos principios metodológicos al análisis de las estructuras de la información gestionadas mediante sistemas automáticos, no sólo me ha permitido hacer cosas nuevas, sino que, al mismo tiempo, he podido experimentar con el uso de herramientas que manejadas adecuadamente son de un «poder» incalculable.

El objeto de este trabajo es doble:

A) El análisis de los diferentes modelos de estructuras de datos utilizados por los Sistemas Integrados de Gestión Bibliotecaria (SIGB) para determinar la relación existente entre dichas estructuras y las prestaciones, en términos de recuperación de información, de las aplicaciones desarrolladas siguiendo estos modelos.

B) Estudiar algunas de las técnicas avanzadas de recuperación de información que aún no han sido implementadas en los SIGB con el fin de proponer una estructura de datos alternativa a las anteriores que permita la ejecución de procesos automáticos de recuperación que mejoren las clásicas técnicas booleanas.

Para lograr este doble objetivo me he servido de los modelos de análisis convencionales ligados al diseño de sistemas, de los modelos físicos y

matemáticos más extendidos en el campo de la recuperación de información y de las técnicas de programación habituales en los estudios de gestión de la información.

Este trabajo tiene su origen en la tesis doctoral que realicé entre 1989 y 1993, por lo que tiene la misma estructura que el original. Por tanto está claramente dividido en dos partes, muy desiguales en su extensión pero similares en su tiempo de realización. Los capítulos que van del primero al quinto son básicamente el resultado del estudio y análisis de la bibliografía consultada, mientras que el capítulo sexto combina un intenso análisis de investigaciones muy recientes con el tratamiento de datos reales. El resultado pretende ser una detallada descripción del estado de la cuestión de los fundamentos estructurales de los SIGB, y una modesta aportación por lo que afecta a las técnicas de recuperación de información empleadas por este tipo de aplicaciones.

Por lo que afecta a la bibliografía utilizada, es importante reseñar que he tratado de utilizar en todo momento las publicaciones más recientes, tanto en artículos de revistas como en monografías. Los títulos de revistas más utilizados han sido *Annual Review of Information Science and Technology* (ARIST), *Journal of American Society for Information Science* (JASIS), *Information Processing and Management, Journal of Documentation* y las distintas publicaciones de la «Association for computing Machinery» (ACM). En cuanto a las monografías son siempre trabajos muy específicos y relacionados con aspectos parciales de las técnicas de recuperación de información en sentido amplio o en los entornos bibliotecarios.

Para terminar con esta introducción quiero resaltar que, como ocurre siempre en un trabajo de estas características, no hubiera podido llegar a este punto en su realización de no haber contado con la colaboración de diversas personas. En unos casos porque han tenido una participación directa en su desarrollo, como la directora del mismo, la Dra. Mercedes Caridad, que me ha apoyado y aconsejado y cuya actitud hacia mí me ha proporcionado la confianza necesaria para llegar al final. En otros casos porque me han soportado en los larguísimos meses de convivencia fantasmal mientras no tenía otro «partner» que el ordenador, como mi familia, y en especial Belén, a quien tanto he martirizado y espero poder compensar algún día. Y, por fin, a aquellos de mis colegas que han aguantado mis «discursos» sobre el tema o incluso se han prestado a leer los originales, como los profesores Pedro Hípola, Javier López y Concha García, así como a mi hermano Manolo y a Pedro Corral, con quienes discutí, a veces inoportunamente, los problemas matemáticos y físicos más vidriosos del trabajo. A todos ellos mi gratitud más sincera.

II

ASPECTOS METODOLÓGICOS

El desarrollo de este capítulo pretende poner de manifiesto, con el mayor detalle posible, la metodología utilizada para la realización del presente trabajo de investigación. Esta metodología ha sido desarrollada a partir, fundamentalmente, pero no de manera exclusiva, de algunas de las conclusiones del grupo ANSI/X3/SPARC [Tsichritzis 78] y de ISO [Iso 82], así como de las aportaciones del grupo Galacsi [Galacsi 86]. Partiendo de conceptos ligados al diseño de Sistemas de Información Automáticos (SIA a partir de ahora) expuestos en estos documentos, he descrito una metodología cuyo esquema general está basado en estos modelos, pero que, en cada una de sus fases, incorpora variantes apoyadas en otros autores.

Las variantes introducidas sobre la base bibliográfica elegida han tenido que ser incluidas como consecuencia de ciertas disfunciones que se produjeron a lo largo del desarrollo del propio trabajo. En cuanto a la selección de varias fuentes metodológicas diré que no es fácil encontrar una metodología de diseño y análisis de SIA que pueda ser utilizada para el desarrollo completo de un sistema tan complejo como el de una biblioteca. Las fuentes elegidas adolecen también de ese mismo defecto, pero complementando unas con otras creo haber empleado un método completo y satisfactorio, no extrapolable por estar demasiado ligado a las especificidades de la información a la que se aplica, pero eficaz en este caso.

Esta metodología que a continuación expongo esta dividida en tres fases, que coinciden, fundamentalmente, con las fases típicas del desarrollo de un SIA tal y como se describen en los documentos antes citados.

Primera fase: Análisis de la organización. Pretende describir la organización social de la que me ocupo, la biblioteca, y más específicamente de los flujos informativos que se producen en su seno. Todo esto con el fin de poder desarrollar un modelo de descripción que pueda ser utilizado en fases posteriores. Sin pretender teorizar excesivamente, sino tratando de definir una estructura que sea lo más funcional posible. Téngase en cuenta que en todo momento, se deberán utilizar como punto de referencia las

características orgánicas y funcionales de la biblioteca con el fin de que las conclusiones de este análisis organizativo nos permitan elaborar, de manera muy concreta, un Sistema Integrado de Gestión Bibliotecaria que, al fin y al cabo, es el objetivo último de este proyecto. Quizá si este objetivo hubiera sido establecido a otro nivel, probablemente, en esta primera fase de desarrollo del trabajo, hubiera sido posible también proponer un modelo de descripción organizativa más teórico, más en la línea de los autores que han especulado con la teoría de la biblioteconomía.

Segunda fase: Análisis funcional. Se recurre en esta fase a modelos conocidos, básicamente de dos tipos, el modelo de datos y el modelo de procesos. El objetivo es establecer un esquema conceptual que integre datos y procesos y que constituya un punto de referencia sólido y al margen de cualquier hardware o software, por muy básico que sea éste. Es decir, el análisis funcional pretende ser, utilizando la terminología informática más al uso, hardware and software independent pero, al mismo tiempo, ha de ser un análisis ligado a los modelos de gestión bibliotecaria en uso en la actualidad, es decir, que los utilice como punto de partida. Y esto sin olvidar que el objetivo final de este análisis funcional es hacer una propuesta que mejore, en alguna medida, las soluciones en uso hoy día.

Tercera fase: Análisis orgánico. Consiste en hacer una adaptación específica de la solución funcional propuesta para la gestión de cualquier tipo de biblioteca. Es preciso señalar aquí que como el objetivo de este trabajo no es el desarrollo de una solución informática completa, de esta última fase sólo se han incluido algunas referencias en el último capítulo del trabajo. Esta fase se divide en cinco subfases:

1. Organización de los datos en forma de tablas y/o registros.
2. Organización de los procesos, definiéndolos, por una parte, y asignándolos a alguno de los tres grupos clásicos: procesos batch, procesos transaccionales y procesos interactivos.
3. Soluciones técnicas informáticas específicas, especialmente por lo que se refiere a la elección del software básico necesario.
4. Interface hombre/máquina, tanto por lo que se refiere al hardware, como por lo que afecta al software.
5. Programación.

A continuación desarrollaré cada una de las fases metodológicas con intención de detallar los pasos que he ido dando a lo largo del trabajo y en qué medida esos pasos se ajustan al método someramente descrito hasta ahora.

II.A. Análisis de la biblioteca y de su sistema de organización

El punto de partida de este análisis es la llamada estructura organizacional de una biblioteca que, para nosotros, utilizando uno de los modelos clásicos de análisis de organizaciones, el modelo entidad-relación [Chen 77, Yourdon 89], estará formado por un conjunto de entidades y relaciones que serán los elementos constitutivos de esta estructura.

Las entidades pueden ser de naturaleza muy diversa: personales, institucionales, documentales e incluso existirán lugares físicos como entidades. Esta naturaleza diversa de las entidades tiene implicaciones decisivas en la configuración de la estructura organizacional, especialmente porque, en el caso de la biblioteca como organización, su objeto, como gestora de grandes cantidades de materiales informativos, le confiere un carácter de institución ocupada en la prestación de servicios de información, que condicionará grandemente la estructuración de su SIA.

En cuanto a las características de las relaciones, también éstas lo son de naturaleza muy diversa. Las relaciones están muy condicionadas por sus fines, así como por las entidades que las conforman. Así, por ejemplo, es indudable que la relación que se establece entre un documento y el usuario que se lleva en préstamo dicho documento es una relación muy distinta de la que se establece entre ese mismo documento y su proveedor, por lo que, como se verá después, las operaciones ligadas a ambas relaciones serán de naturaleza muy distinta.

Es importante señalar que las relaciones entre las entidades, de cualquier organización, de manera abstracta, pueden ser de grado dos o mayor que dos, pero de entre todas las posibles relaciones que se pueden establecer entre entidades de una organización a nosotros nos interesan sólo aquellas que tienen un carácter permanente, en la medida en que esas relaciones con carácter permanente son valorables desde el punto de vista de la actividad de la organización y, por tanto, formalizables, como tales relaciones junto con los procesos que están ligados a ellas. Por lo que afecta a las relaciones con carácter eventual, sólo decir que, aunque podrían ser tenidas en cuenta si lo que pretendiéramos fuera el desarrollo de un producto comercial, sin embargo, dado que lo que se pretende es determinar los aspectos esenciales de la organización bibliotecaria, con el fin de poder gestionar mediante procedimientos automáticos los diferentes flujos informativos que de ella surjan, no me parece, por tanto, necesario tomar en consideración este tipo de relaciones.

Serán, por tanto, entidades pertenecientes a la estructura organizacional bibliotecaria todas aquellas que mantengan algunas relaciones con otras entidades en el curso del desarrollo de las actividades del conjunto de la organización. Esto quiere decir que cualquier entidad que, en prin-

cipio, parezca extraña a la organización misma por el hecho de mantener alguna relación con entidades identificables con la organización, serán automáticamente consideradas como entidades de la propia organización.

Tanto las entidades como sus relaciones se definen también en función de sus propiedades, lo que tendrá, como veremos, mucha importancia de cara a la definición del SIA, puesto que la indefinición de algunas propiedades imposibilita las relaciones y es la suma de ambas lo que configura, si no completamente, sí en gran medida, dicho sistema. Así mismo, las entidades, relaciones y por supuesto sus propiedades son sujetos dinámicos dentro de la estructura organizacional. Lo que convierte a ésta en algo en constante evolución. Esto tendrá una gran influencia posterior en el análisis funcional, porque al realizar dicho análisis y predefinir el modelo de gestión del conjunto del sistema, habrá que tener en cuenta la dinamicidad del mismo. Si esto no se tuviera en cuenta, tendríamos un modelo de gestión que no permitiría su adaptación a los cambios que la propia estructura organizacional va experimentando. Por ejemplo, la aparición de nuevos elementos o nuevas propiedades de los elementos existentes que se van incorporando a la estructura organizacional deben tener su reflejo lo más pronto posible en el SIA con el fin de que esos nuevos elementos puedan ser controlados por el sistema. De no ser así su desarrollo se irá alejando de la estructura organizacional, será cada vez menos su reflejo y, por consiguiente, no servirá a los fines de la organización.

La organización está sujeta a cambios como consecuencia de su permanente interacción con el resto de los elementos que conforman el cuerpo social. Así, en el caso de las bibliotecas es especialmente claro, pues la necesidad de adaptar el funcionamiento de la biblioteca para que pueda prestar los servicios que de ella se exigen la obliga permanentemente a alterar el conjunto de las entidades que la conforman o de las propiedades que son constitutivas de esas entidades y, por supuesto, también del tipo de relación que se establece entre estas entidades. Algunos ejemplos de esto podremos ver en su momento.

Esta interacción es inherente a la consideración de organización que tiene la biblioteca, entendida como un agregado de entidades relacionadas en constante actividad. La actividad a la que está sujeta toda organización hay que entenderla como un flujo constante de hechos independientes y sucesivos, que pueden ser clasificados en razón de su interdependencia funcional y de su ubicación en el "mapa" de la estructura organizacional.

Es importante reflexionar sobre la independencia de estos hechos porque si no fueran considerados como independientes su formalización en el momento de la constitución del SIA sería difícil y esto habría hecho bastante complicada la realización de este proyecto. Es necesario proceder a

analizar la organización, proceder a la disección no sólo de los elementos constitutivos —entidades y relaciones— de la organización sino también la sucesión de actividades, muchas veces muy ligadas unas con otras, que se producen en el seno de un biblioteca. Por eso, en esta fase del trabajo es importante identificar esas actividades y al mismo tiempo definir su especificidad, para separar cada una de las demás. Toda esta actividad que se desarrolla en el seno de la biblioteca está permanentemente controlada en cualquier organización por un conjunto de medios técnicos y humanos cuya actividad se denomina gestión, en este caso, gestión bibliotecaria.

Este control es necesario para evitar disfunciones en la actividad de la organización que la aparten de sus fines. Hasta tal punto importa la actividad de gestión que los medios que la realizan forman por sí solos el llamado SIA de la biblioteca. En este punto quizá sea importante comentar que para realizar la descripción de la organización bibliotecaria, aparte del proceso que vengo describiendo aquí, como la organización bibliotecaria es muy compleja, con múltiples funciones, he procedido a hacer una disección previa que podríamos llamar departamental, de tal forma que han quedado inicialmente los procesos en grupos para que resulte más operativa la aplicación de esta metodología. Este aspecto será desarrollado más ampliamente al comienzo del capítulo siguiente.

Volviendo al SIA, es importante resaltar que algunos autores separan lo que se llama el sistema de decisiones de una organización del SIA propiamente dicho. Deliberadamente, yo no establezco esta distinción y considero que de acuerdo con el desarrollo de la tecnología actual es difícil separar la identificación de un problema de su resolución, puesto que en la identificación del problema está implícita muchas veces su solución. Al mismo tiempo es muy corriente hoy que en las organizaciones sean precisamente los responsables del SIA los propios ejecutivos de las organizaciones. Yo definiría, por tanto, en sentido amplio, el SIA en base a sus elementos constituyentes y a su objeto. Sus integrantes, como es sabido, son un conjunto de medios y métodos que tienen por objeto el tratamiento de todo tipo de informaciones que fluyen en una estructura organizacional. Ni que decir tiene que los medios que se ponen en juego en un sistema de información son medios humanos, técnicos o materiales que no voy a enumerar aquí porque este trabajo no se ocupa de los medios de los SIA y menos aún de los medios de la organización bibliotecaria.

Si tratamos de hacer una definición más estricta, más ligada al tema del presente trabajo, que es la creación de un Sistema de Información Automatizada para las bibliotecas, entonces tendríamos que identificar sistema de información con métodos automáticos de tratamiento de diversas formas mecanizables de información en las bibliotecas. De alguna forma lo que se pretende con esta definición es considerar el sistema de informa-

ción automatizada como un subconjunto del sistema de información real. En este sentido, es importante hacer algunas precisiones en torno a la definición, pues nos ayudarán a delimitar el marco de desarrollo de este trabajo y de esta forma se encontrará justificación a alguna de las restricciones implícitas que el propio trabajo contiene. Téngase en cuenta que los parámetros de un Sistema de Información son tan diversos que pueden dar lugar a una gran variedad de enfoques en su análisis [Chaumier 86, Salton 75].

En primer lugar, me gustaría definir lo que entiendo por actividad mecanizable en esta definición. Considero que cualquier actividad lo es en la medida en que es representable algorítmicamente. Es decir, que es controlable mediante procedimientos informáticos. Esto quiere decir que el ámbito de este trabajo se reduce a la gestión de las informaciones que son procesables informáticamente en una biblioteca.

Otro concepto de la definición que me gustaría acotar es el de métodos automáticos. Los métodos automáticos son, fundamentalmente, métodos informáticos y, evidentemente, esto supone que no hablaré aquí de nada que tenga que ver con procesos cuyo tratamiento mediante ordenadores no sea posible.

En cuanto al término tratamiento, entiendo básicamente por tratamiento tres tipos de procesos: almacenamiento, consulta y modificación de la información, con todas sus variantes. Aunque algunos autores desarrollan toda una tipología de procesos genéricos que se pueden realizar con la información, en este trabajo, como opinan otros, consideraré sólo estos tres tipos de tratamientos porque entiendo que en definitiva todos los demás se reducen a éstos.

En cuanto a los tipos de información es importante establecer una distinción que condicionará, en gran medida, algunos de los contenidos de este trabajo. Básicamente todos los autores coinciden en que las dos formas generales de información en un sistema automatizado son lo que se ha dado en llamar información estructurada e información no estructurada. La información estructurada es una información representable tabularmente, en la que cada elemento definido en el sistema tabular representa un «item informativo». Por explicarlo de una manera sencilla, información estructurada es aquella cuya estructura coincide, esencialmente, con la estructura del sistema definido. Por el contrario, información no estructurada es una información que no se presta bien a los tratamientos algorítmicos, puesto que la ausencia de una estructura predefinida dificulta la definición de procesos desligados de los contenidos informativos específicos. Sólo la fragmentación en las unidades de significado más simples permite su tratamiento. Por tanto, en el caso de la información no estructurada, podemos decir que la estructura definida en

el sistema no coincide con la estructura de elementos informativos que contiene dicha información.

De acuerdo con todo esto la creación de un sistema de información automatizado se puede realizar en 4 grandes fases:

- En primer lugar, es necesario crear una imagen de los elementos constitutivos de la estructura organizacional.
- En segundo lugar, hay que crear una imagen de las actividades que realiza la organización; en este caso, la biblioteca.
- En tercer lugar, hay que definir los procesos que se realizarán con la información estructurada.
- Finalmente, definir los procesos que se realizarán con la información no estructurada.

En cuanto a la primera fase, la representación formal de las entidades y relaciones que forman la estructura organizacional, hay que realizarla teniendo en cuenta dos limitaciones importantes. Por una parte, sólo las entidades y relaciones formalizables para fines de control automático de la gestión informativa serán tenidas en cuenta. Pues, como dije anteriormente, el SIA en realidad lo que persigue es realizar un control de las actividades que se realizan en la biblioteca y, por consiguiente, aquellas entidades y relaciones que no tengan ninguna relevancia de cara al control de la gestión informativa, no deben formar parte del sistema de información.

En cuanto a la segunda restricción, es de carácter subjetivo, evidentemente, pero es necesario consignarla en la explicación del método, ya que solamente serán tenidas en cuenta aquellas entidades y relaciones que hayan sido percibidas por quien hace el análisis de la organización. Evidentemente, cualquier sistema de información, en la medida que es una representación formalizada de una realidad, tiene un cierto carácter subjetivo, que sólo puede ser superado por el estudio de la mayor cantidad de visiones subjetivas posibles. En este sentido, teniendo en cuenta las dos restricciones que acabo de enunciar, se puede decir que el sistema resulta ser una visión simplificada de la realidad de la estructura organizacional bibliotecaria y como tal sistema no aspira tampoco a otra cosa más que a ser una visión parcial. En cualquier caso, lo que verdaderamente importa no es si refleja o no completamente la realidad organizacional, sino si con ese sistema se puede realizar la función de control informativo antes mencionada.

Como veremos a continuación, la formalización necesaria para la creación de las imágenes de las entidades y sus relaciones implica la definición de propiedades implícitas de esas entidades y de esas relaciones. Algunas

veces, las propiedades acompañan a la simple denominación de la entidad, pero otras propiedades no son tan evidentes por sí mismas. Estas propiedades, entre otras cosas, como veremos, implican la clasificación de las informaciones como estructuradas o como no estructuradas, lo que tendrá consecuencias importantes en fases posteriores del desarrollo del presente trabajo. Cada entidad se define, como acabo de decir, por un conjunto de propiedades que tienen diferentes valores. Algunas de esas propiedades solamente tienen un valor descriptivo-diferenciador, mientras que en otros casos, la propiedad desempeña una función vital en la realización de ciertas actividades o en la conformación de las relaciones. Este diferente carácter de las propiedades debe ser tenido en cuenta en esta fase, pues, en definitiva, una correcta definición de las propiedades nos permitirá hacer un análisis funcional y orgánico correcto. Si las propiedades no estuvieran bien definidas, podríamos hacer un correcto análisis funcional, pero difícilmente orgánico, e incluso podría haber errores en el análisis funcional.

En cuanto a las relaciones, éstas son directamente consecuencia de la actividad organizacional de la biblioteca. Aunque en teoría cualquier relación es posible, sólo tienen sentido y, por tanto, serán reflejadas en el sistema, aquellas que se dan como consecuencia del desarrollo de las funciones que permitirán cubrir los objetivos de la organización.

Esto quiere decir que las relaciones serán fácilmente identificables al hacer la nómina de las actividades en las que intervienen las entidades consignadas anteriormente. No basta con explicitar la relación en base a la actividad que la ocasiona. Es necesario, además, establecer sus características. Hago hincapié en esta cuestión, pues, como se verá en el desarrollo de esta parte del trabajo, lo que tendré que hacer fundamentalmente es relacionar funciones —actividades, si se quiere— que se realizan en las bibliotecas y, en todo momento, se podrá plantear la necesidad de mencionar una determinada función en lugar de otra. En este sentido, la explicación en la mayoría de los casos, vendrá dada por las restricciones que he mencionado anteriormente, pero también es verdad que en este trabajo no pretendo agotar el tema, sino probar un modelo de gestión de la información como más eficaz para las bibliotecas, de tal manera que, aunque ciertas funciones no aparezcan reflejadas en el trabajo, me parece que la nómina de las que sí han sido probadas es suficientemente significativa como para dar validez al modelo.

El segundo apartado del desarrollo del SIA es la creación de una imagen de las actividades que realiza la organización. Esta representación es factible en la medida en que se puedan aislar las actividades que se realizan en la biblioteca. Este es un proceso tedioso porque es necesario reproducir cada uno de los trabajos que se efectúan en una biblioteca, tratando de dividirlos en procesos lo más elementales posibles.

Este mecanismo obliga a hacer una definición, lo más detallada posible, de los trabajos bibliotecarios divididos en áreas, o, como luego los denominaré, en «funciones básicas». Esta división en áreas es más ficticia que real, pues los propios procesos descritos implican una interrelación constante entre unas áreas y otras. Por ello, esta división tiene más por objeto simplificar la descripción que motivos metodológicos.

Por otra parte, un grupo de actividades importante dentro de cualquier organización es la relacionada con la presentación del estado de las entidades y de las relaciones. Por ejemplo, en una biblioteca pedir información sobre las entidades con información bibliográfica o sobre las entidades relacionadas con los prestatarios de las bibliotecas es una actividad normal. En realidad se trata de solicitudes de información sobre el estado de las entidades. Por ejemplo, solicitar información de préstamos dentro de una biblioteca, en realidad es solicitar información a propósito de las relaciones entre las entidades que conforman el «catálogo» y la entidad «prestatarios».

Por último y muy relacionado con esto, en cualquier organización, y en una biblioteca por supuesto también, es necesario reflejar en todo momento aquellas actividades que tienen que ver con la reconstrucción histórica de las propias actividades realizadas en la biblioteca, de tal manera que la reconstrucción histórica de actividades se convierte, también, en una actividad más.

La tercera fase de la creación del sistema automatizado es la que tiene que ver con la representación del tratamiento de los datos de la información estructurada.

Realmente, una vez definidas las actividades a realizar, hacer el tratamiento de la información estructurada resulta sencillo, aunque es importante hacerlo con una gran precisión, pues esta definición de los tratamientos, si resulta vaga, se convierte en una definición inútil en el momento de su formalización algorítmica o de su representación en el lenguaje informático. Esta necesidad de precisión de los tratamientos nos lleva a hacer una distinción que puede ser muy esclarecedora. Hay ciertos tratamientos que tienen como consecuencia salidas de datos que han sido demandados por un usuario del sistema, pero hay otros tratamientos cuyo resultado es entrada de un tratamiento posterior y, por tanto, los usuarios no perciben la realización de esos tratamientos.

En el caso de una biblioteca, esta distinción es especialmente importante, porque hay cierto tipo de procesos que necesariamente tendrán que ser realizados de forma transparente para el usuario y, sin embargo, son procesos vitales en el funcionamiento del conjunto del sistema. Esto

quiere decir que la identificación de cada uno de los tratamientos y su asignación a un grupo u otro facilitará enormemente su formalización posterior.

Finalmente, el último apartado tiene que ver con el tratamiento de la información no estructurada.

Este tratamiento, una vez que ha sido identificada la información no estructurada, no plantea graves problemas puesto que existen herramientas y soluciones informáticas más que suficientes para hacer el tratamiento de esta información, que tendrán que ser procedimientos no algorítmicos —si se admite esta expresión—. Lo importante, por otra parte, es hacer una definición lo más precisa posible de los objetivos del tratamiento de la información no estructurada que, evidentemente, no coinciden con los objetivos del tratamiento de la información estructurada. En este sentido, hacer una distinción lo más clara posible de unos tratamientos y otros ayudará considerablemente al desarrollo del conjunto del sistema, especialmente en la fase del análisis funcional, donde este tipo de decisiones tendrán una repercusión muy importante de cara a la realización del análisis orgánico.

II.B. Análisis funcional

Una vez analizada la estructura organizacional a partir de los conceptos de entidad y relación, habrá que definir un modelo de gestión automática del conjunto resultante. Esta definición utilizará como punto de partida la estructura que se haya modelado en la fase anterior. Pero, tratando de elaborar un esquema conceptual que integre los datos que forman parte del sistema y sus correspondientes tratamientos. Dicho esquema conceptual será denominado a lo largo del presente capítulo, tanto de esta forma como con el término modelo.

Por otra parte, no creo que sea posible separar en la exposición datos y tratamientos, por lo que serán expuestos sincrónicamente. En mi opinión ciertos tratamientos sólo son posibles con determinado tipo de datos, por lo que resulta evidente que el modelo de datos condiciona muchas veces el tratamiento que se puede hacer de ellos. En este sentido, a diferencia de como lo han hecho otros autores, mi exposición irá describiendo los distintos conjuntos de datos y su organización para cada uno de los modelos junto con los tratamientos posibles en cada modelo.

Al hablar aquí de datos pretendo abarcar el conjunto de informaciones que en la fase anterior han sido definidas como informaciones mecanizables. En este sentido es importante recalcar que sólo analizaré aquellos modelos que han sido diseñados con intención de dar solución a la

gestión íntegra del conjunto de información que fluye en la organización bibliotecaria. Los modelos con aspiraciones parciales no tienen cabida en este trabajo. Ello implica que sólo me ocuparé de los esquemas conceptuales utilizados en el diseño de los llamados Sistemas Integrados de Gestión Bibliotecaria (a partir de ahora SIGB), tratando de llegar al modelo más eficaz en la gestión automática del sistema de información de una biblioteca. Esto me obliga a interrumpir aquí la descripción de la metodología en esta segunda fase para hacer algunas precisiones sobre el concepto de SIGB que yo manejo. Realmente este concepto está basado en la opinión de muchos autores que recientemente han aportado sus puntos de vista sobre esta cuestión [Kemp 88, Clayton 91, Reynolds 89, Tedd 88, etc.].

En mi opinión y recogiendo también la de estos autores, las señas características básicas de un SIGB son:

* En primer lugar que éstos son el resultado de la evolución desde sistemas monofuncionales que se utilizaban hasta finales de los 70 en las bibliotecas y que tenían por objeto resolver el problema de la gestión mecánica de aquellas funciones que suponían un mayor costo en recursos humanos, especialmente en las grandes bibliotecas. Ello supuso la aparición de sistemas de gestión mecánica del préstamo a los que luego se fueron añadiendo otros sistemas (gestión de catálogos, control de publicaciones periódicas...). Cada uno de ellos, en definitiva, resolvía automáticamente la gestión de una función concreta de la gestión bibliotecaria.
* En segundo lugar, los SIGB son aplicaciones multifuncionales de control mecánico de la información que se estructuran de manera muy similar a la forma tradicional en que se organizan las bibliotecas. No es fácil establecer de manera general un grado de funcionalidad tipo para estos sistemas, pues, en cada caso, el diseñador ha resuelto implementar un número específico de funciones mecánicas en el sistema, lo que da como resultado SIGB funcionalmente heterogéneos.
* En tercer lugar, todos los SIGB trabajan contra una sola base de datos para evitar redundancias informativas. Esta cuestión es consecuencia de la forma en que aparecieron estos sistemas, puesto que sus predecesores —los sistemas monofuncionales—tenían como dificultad fundamental la de no entenderse bien entre sí, al haber sido desarrollados independientemente los unos de los otros. Por ello, los SIGB pretenden aportar una solución funcionalmente integrada e informativamente no redundante.

* Por último, hay que entender que los SIGB en sus versiones comerciales pretenden ser programas de «propósitos generales» dentro del mundo bibliotecario, de tal forma que con la misma solución, el suministrador de ese programa pretende resolver el problema de la mecanización de cualquier tipo de biblioteca. Esto las ha convertido en aplicaciones muy adaptables, especialmente las más modernas, hasta tal punto que su proceso de «tayloring» resulta, en ocasiones, complejo por la enorme cantidad de información parametrizable que poseen. Este hecho, al margen de los indudables motivos comerciales que lo sustentan, pone de manifiesto que la evolución de estos sistemas les lleva a intentar ser soluciones de carácter universal.

Quizá las palabras de M. Clayton publicadas recientemente sobre los SIGB puedan resumir con bastante precisión en qué momento de la evolución de la mecanización bibliotecaria se encuentran estos sistemas:

> "Ultimamente los diseñadores de sistemas han comenzado a subrayar las ventajas de un planteamiento integrado para la automatización, es decir, una base de datos común se procesa mediante programas de aplicación que realizan diversas funciones de apoyo técnico, por ejemplo, adquisiciones, catalogación, circulación, control de publicaciones periódicas y catálogo de acceso público en línea. Ha sido posible gracias a la disponibilidad, en la década de los 80, de microordenadores con iguales capacidades y velocidad de ejecución que los de los ordenadores centrales de los años 60 y 70, pero a un coste más reducido. El establecimiento de un sistema integrado en el que todas las funciones comparten una base de datos elimina o aminora la redundancia de la información y por consiguiente disminuye los costes"[Clayton 91, pág. 51].

De acuerdo con este concepto de SIGB el proceso de selección de los modelos elegidos, que paso a explicar, puede ser revelador de algunas de sus características más genéricas y, en cualquier caso, completa esta exposición del método de trabajo.

El estudio pormenorizado de los diferentes programas de mecanización bibliotecaria que existen en el mercado permite determinar los modelos de datos y tratamientos que se vienen utilizando como base de su desarrollo. Esta parte del trabajo me ha supuesto muchas dificultades por la falta de acceso a cierto tipo de información sobre aspectos que no fueran puramente operacionales. Normalmente, los suministradores de estas

aplicaciones no proporcionan mucha información sobre el funcionamiento interno de la aplicación. Es decir, la documentación está muy dirigida al usuario final de la aplicación, pero, sin embargo, aporta mucha menos información sobre temas tales como estructura de ficheros, métodos de acceso a la información utilizados, elementos físicos de los datos, algoritmos básicos de los procesos generales, etc. De tal manera que la documentación que acompaña a cada una de las aplicaciones, en la mayor parte de los casos servía bastante poco a mis propósitos. Por tanto, el proceso de análisis que tuve que seguir fue siempre un proceso de inferencia a partir de efectos operativos, para terminar con el estudio de los datos físicos concretos.

De toda esta parte del estudio, en el presente trabajo sólo queda reflejado el resultado final, es decir, las conclusiones de este esfuerzo de análisis que me han permitido hacer una clasificación de los esquemas conceptuales en que se han basado los distintos desarrollos existentes en el mundo bibliotecario y denominados genéricamente SIGB. Quiero aclarar también que el grado de conciencia que el analista de la aplicación tiene respecto de la elección de un modelo conceptual u otro es muy pequeño, pues, por una parte esa decisión es una decisión informática, no condicionada por las características de la información a procesar, y cada esquema tiene a su vez una clara vocación universalista, lo que implica que pretende ser por sí mismo la clave de la solución de cualquier problema de gestión de información, cuando, en realidad, es consecuencia directa de un determinado nivel de desarrollo tecnológico y es precisamente ésta la causa de su elección en la mayoría de los casos. Esta confiere, además, a la clasificación de los sistemas realizada un marcado carácter histórico, puesto que cada uno de los tipos de clasificación se corresponde con un determinado modelo de solución que, en un período corto, de desarrollo de los sistemas informáticos, en general, se utilizó más.

Cuando Kruglinski [Kruglinski 83] hace su clasificación de los modelos de gestión de información utilizados por la informática, considera que en la etapa anterior a la aparición de los sistemas de gestión de bases de datos, el modelo utilizado es el llamado Sistema de Gestión de Ficheros (FMS). Siguiendo esta terminología, he considerado que el esquema de tratamiento de información utilizado, en primer lugar, por los SIGB fue del tipo FMS, en consonancia con el modelo utilizado por los analistas funcionales de aplicaciones a finales de los años 70, época en la que comienza el desarrollo de los primeros sistemas integrados. En este sentido, a nivel del analista funcional, los SIGB son un caso más de aplicación informática compleja y su desarrollo corre parejo con el de otras aplicaciones complejas de gestión de información. Así mismo, como insinué antes, estas aplicaciones son muy sensibles a los cambios tecnológicos en el campo de

la ingeniería del software. Esto explica, por otra parte, que en el transcurso de sólo una década se hayan utilizado tres esquemas conceptuales distintos y se proponga un cuarto para el desarrollo de un solo tipo de aplicación.

Volviendo al primero de los esquemas estudiados, hay que decir que ha sido el más difícil de desentrañar, pues las aplicaciones resultantes adquieren formas muy distintas que hacen olvidar, en ocasiones, el modelo que les dio lugar. Hasta tal punto esto es así que a veces me costó reconocer la existencia del FMS como un modelo único tal y como proponía Kruglinski. Reducir, por tanto, a sus aspectos esenciales en la descripción los contenidos de este modelo, es la clave del desarrollo de esta parte del trabajo, ya que, si esto no se hace así, convertiría el análisis funcional de este modelo en una descripción general de aplicaciones bibliotecarias que empiezan a estar pasadas de moda en su concepción, aunque no en su uso. Por otra parte, resultaba muy difícil además, el análisis de este tipo de modelos, porque por su propia naturaleza, las aplicaciones desarrolladas de acuerdo con el modelo FMS son muy dependientes de un determinado software básico, pues normalmente están muy ligadas al método de acceso del sistema operativo bajo el que han sido desarrolladas.

En cuanto al segundo modelo, se trata de uno de los más conocidos, el llamado modelo relacional (RM/V2). Para estudiarlo he prescindido de las aplicaciones y me he dirigido directamente a sus mentores más reputados: Codd y Date. Partiendo de las definiciones que hace su creador [Codd 70] he tratado de especular sobre las consecuencias de la utilización de dicho modelo en la gestión de un sistema automático de información bibliotecario. La enorme ventaja que supone la existencia de documentación abundante sobre la definición del modelo y sus reglas de funcionamiento facilita mucho las cosas en esta parte del trabajo, aunque, por otra parte, obliga a un rigor no siempre fácil de mantener. En este sentido, el hecho de que el modelo relacional sea en su origen un modelo matemático añade una dificultad más en la exposición, pero, por otra parte, limita los excesos especulativos tan socorridos y tan perniciosos en ocasiones. Al mismo tiempo, el propio rigor del modelo ha sido la causa de la aparición de un cuerpo normativo que he tenido muy en cuenta a la hora de hacer su estudio, ya que en el contexto bibliotecario, la utilización de los llamados "recursos estándar" supone una garantía de cara a la homologación de la información procesada. Es preciso recordar aquí que la homologación de los procesos en el mundo bibliotecario hace imprescindible pensar en estas organizaciones como estructuras abiertas al exterior y cuyos procesos internos de tratamiento de la información deben, siempre que sea posible, normalizarse.

En relación con todo esto, el modelo relacional puede contribuir a resolver problemas que otros modelos ni se plantean. Así, por ejemplo, la visión de un conjunto de estructuras bibliotecarias funcionando sincrónicamente es factible desde la perspectiva del modelo relacional y, sin embargo, no lo es tanto o, por lo menos de forma tan evidente, desde la perspectiva de otros modelos.

El tercero de los modelos en estudio es el llamado IRS (Sistemas de Recuperación de Información), que fue desarrollado con gran éxito y para fines distintos que los puramente bibliotecarios en los años 60. En la actualidad, este modelo, a pesar de su antigüedad, continúa vigente, con sustanciales mejoras respecto de las características de sus orígenes. Como consecuencia de las dificultades que entraña la gestión de información no estructurada por procedimientos automáticos se ha ido mejorando considerablemente este modelo de gestión. Es el menos utilizado de los tres en los SIGB, pero ha sido incluido en este estudio porque resuelve de manera eficaz algunos de los problemas de gestión de la información bibliográfica que los otros sistemas no llegan a resolver. Por esta razón, este modelo desempeña un papel fundamental en la conformación de mi propuesta de modelo alternativo. En la descripción de este esquema, aunque la documentación es abundante, he preferido utilizar un enfoque más funcional que en el caso anterior y baso mi descripción en la estructura de datos que el modelo genera y de la que se sirve, para especular, a continuación, sobre su aplicabilidad en el entorno bibliotecario. La razón de este enfoque es que no he conseguido encontrar en la documentación utilizada por mi una descripción de este modelo que no se base en sus prestaciones, lo que no me ayudó gran cosa a teorizar lo bastante como para poder elaborar una justificación teórica rigurosa del modelo en cuestión.

Normalmente, los autores que se han dedicado a tratar el tema de la recuperación de información textual, dedican sus esfuerzos a analizar procedimientos para mejorar el funcionamiento de los IRS, de tal manera que no es fácil encontrar, fuera de autores como Salton o Van Rijsberger [Salton 68, Rijsberger 79], definiciones del modelo general que puedan ser utilizadas como punto de partida, y aun éstas definen el modelo en base a sus prestaciones.

Aunque es evidente que el objeto del presente trabajo no es tanto desarrollar un modelo original de gestión automática de la información, sino más bien estudiar la conveniencia o no de utilizar un modelo u otro de entre los existentes, es asimismo cierto que el diferente grado de conceptualización al que he llegado en la descripción de los modelos no es sino el reflejo de la visión que los especialistas tienen de cada uno de ellos. Las causas habrá que buscarlas en el proceso de gestación de cada uno de los modelos en cuestión.

Finalmente, el cuarto modelo descrito es el que he llamado modelo híbrido o mixto. Su nombre lo sugiere la forma en que A. Kemp [Kemp 88] llama a un tipo de SIGB cuyo funcionamiento se basa en la combinación de un esquema relacional para el tratamiento de una parte de la información y de un IRS para el tratamiento de otra parte. Kemp lo llama sistema mixto y ya insinúa que existen muy pocos puntos de referencia para analizarlo. Rastreando la bibliografía especializada sobre el tema, no he podido encontrar más de dos o tres referencias a los llamados sistemas mixtos o híbridos, por lo que, en realidad, el trabajo ha consistido principalmente en el estudio pormenorizado de los resultados que podrían obtenerse introduciendo en el desarrollo de un SIGB basado en este modelo mejoras a nivel de la estructura de datos y, sobre todo, en los sistemas de recuperación de información. Como se verá en la descripción de este modelo, se parte de la idea de que las deficiencias encontradas en los esquemas anteriores pueden ser superadas mediante la combinación de sus efectos. Esto hace que la descripción de este modelo se base en parte en la forma de acoplamiento de los esquemas que se pretende combinar, de tal manera que las prestaciones de cada modelo no se vean perjudicadas por el otro. Las referencias encontradas en la literatura de este modelo, como ya he dicho, son mínimas y esto ha hecho que su conformación haya tenido que ser, en gran parte, elaborada por mí. Por eso, incluso la forma en que este modelo es descrita, atiende más a las dificultades de su generación en la tercera fase que a una definición más o menos teórica del mismo.

II.C. Análisis orgánico

La fase del análisis orgánico culminará el proyecto de diseño y ejecución de un SIGB. En torno a esta fase he de decir, en primer lugar, que, en sentido estricto, va más allá del objeto del presente trabajo que, como se recordará, queda cubierto con la fase de análisis funcional, pero me ha parecido necesaria su inclusión porque sin ella no se podrían probar algunos de los efectos que se presumen conseguidos en la fase anterior, así como tampoco se descubrirían problemas no previstos, por lo que se puede atribuir un cierto valor de feed-back sobre el proceso general a esta última fase. En cualquier caso, como queda dicho, el objeto mismo del presente trabajo se ha alcanzado con las conclusiones extraídas en la fase anterior, puesto que con ellas podemos optar por la mejor solución en términos de adecuación de sus prestaciones y de acuerdo con el desarrollo de la tecnología del software en el momento presente. Sin embargo en la informática ocurre con demasiada frecuencia que los modelos definidos

son difícilmente trasladables a código de programación. En este caso esta traslación podría plantear especiales dificultades al exigir, de una parte, la combinación de varios modelos de datos y tratamientos, lo que resulta siempre arriesgado por lo incierto de los resultados, y, por otra parte, al tener que combinar en la misma aplicación recursos informáticos de software básico de distinta procedencia: un DBMS y un IRS con sus correspondientes lenguajes, y todo ello además combinado con desarrollos hechos en un lenguaje de bajo nivel para aquellas rutinas a las que se les exige un rendimiento mayor.

En resumen, son razones fundamentalmente metodológicas las que me llevan a incluir esta tercera fase, pero entiendo que, dada la naturaleza del proyecto, hacer una demostración práctica de la viabilidad de la solución, aunque sólo sea a un primer nivel, resulta no sólo útil sino obligado. Téngase en cuenta así mismo que si el punto de partida de este trabajo es la nómina de necesidades de control de información en un determinado tipo de organización, lo lógico es que éste se cierre con la prueba de la satisfacción de las mismas.

Por otra parte, también tengo que decir que la intención probatoria de esta fase no llegará al extremo de convertir en objeto mismo del trabajo la aplicación desarrollada. Por este motivo en ningún momento trataré de hacer una evaluación formal de la misma, ya que no es sino una de las muchas aplicaciones que podrían hacerse siguiendo el mismo modelo. El desarrollo que presento aquí, en definitiva, no pretende ser el mejor de los posibles, sino simplemente una prueba de la viabilidad del modelo propuesto. Por extensión del argumento habrá que entender, por tanto, que cada una de las opciones propuestas en las sucesivas subfases del análisis orgánico se eligen por razones de oportunidad o de disponibilidad de recursos y, por consiguiente, ni tan siquiera las justificaré en mi exposición, ya que dichas opciones deben ser entendidas en todo momento como simples ejemplos. Por consiguiente, de las distintas subfases que a continuación describo, sólo serán tratadas de manera específica en el último capítulo del presente trabajo algunas de ellas y de forma parcial, las que he considerado imprescindibles para la correcta ponderación de los resultados del estudio:

II.C.1. *Subfase I: La organización de los datos en tablas y registros*

Aquí se ha de realizar una descripción somera de cada una de las tablas y registros que formarían la base de datos que el sistema gestionaría, indicando los componentes de cada tabla o registro y sus características.

Las tablas y registros deben ser la proyección de las entidades y relaciones consignadas en el análisis de la organización (fase I). Es preciso señalar que si la información relativa a entidades y relaciones no es lo bastante concreta y precisa, la elaboración de tablas y registros se convertirá en una tarea penosa no exenta de errores, que tendrán su reflejo en el desarrollo de los procesos que, a su vez, no podrán controlar mecánicamente determinadas actividades orgánicas. Sin embargo, también conviene decir, que si bien en lo que afecta al análisis de la organización sí he pretendido ser exhaustivo en el presente trabajo, no así por lo que afecta a la formalización en tablas y registros de esa estructura organizacional. Mi intención en este caso ha sido comentar únicamente aquellas entidades y relaciones que eran imprescindibles para el desarrollo de las funciones que con más insistencia eran mencionadas por los analistas consultados, puesto que tratar de agotar el tema habría alargado innecesariamente esta parte del trabajo.

En cualquier caso, el procedimiento esquemático de presentación de las tablas, que es el habitual en estos casos cuando se diseñan aplicaciones, es en mi opinión el idóneo. Cada tabla o registro se debe presentar en forma de cuadro de doble entrada en el que se recogen las propiedades y características de éstas, de cada entidad y relación. Estos esquemas irían acompañados de breves textos explicativos que llaman la atención sobre los elementos más crípticos de cada cuadro. Por otro lado, en esas breves explicaciones se haría siempre mención de ciertas peculiaridades de los tratamientos en los que se pueda ver envuelta cada tabla o registro, especialmente en aquellos casos en los que los aludidos tratamientos impliquen transformaciones complejas de los datos o transferencias de los mismos por razones relacionadas con la lógica de los procesos en juego, tanto si dichas transformaciones o transferencias se realizan de forma provisional como permanente.

La presentación de los cuadros debería seguir el orden del análisis organizativo, de tal manera que sea fácilmente asociable cada entidad o relación con su cuadro correspondiente. Ni que decir tiene que, en ocasiones, este paralelismo en la exposición no será posible por la simplificación de la realidad que implica siempre una representación esquemática frente a una exposición narrativa, en cualquier caso, no debería ser difícil relacionar una parte con otra.

II.C.2. *Subfase II: Organización de los tratamientos*

En esta subfase se definirían los tratamientos que compondrían el conjunto de la aplicación. Esta definición implica la realización de una descripción somera de cada proceso a realizar, con la consignación de las tablas y/o registros afectados, las entradas de datos necesarias y las salidas obtenidas. Como dije anteriormente, algunos procesos son activados desde la propia aplicación por otros procesos, y sus salidas pueden ser a su vez entradas de otros. A este tipo de procesos se les llama procesos internos. Mientras tanto la mayor parte de los procesos de este tipo de aplicaciones son activados por un usuario, y sus salidas son devueltas por una vía u otra al «llamador». A éstos se les llama procesos externos, es decir, a aquellos que son evidentes con el simple manejo de la aplicación y, por tanto, reconocibles fácilmente por los usuarios. Tanto si son de un tipo como de otro, en la descripción de los mismos se debe hacer constar siempre que sea relevante.

La división clásica de los procesos en batch, transaccionales e interactivos debe ser entendida siempre como una clasificación de modos de ejecución, de tal forma que cuando se alude a alguno de los tipos de esta clasificación, siempre se hace a manera de recomendación de un modo de ejecución específico, aunque dando por puesto que ese proceso puede ser ejecutado de manera distinta. Por defecto, todos los procesos deben ser interactivos mientras no se especifique lo contrario. Estos procesos siempre implicarán operaciones con la información de la base de datos, por lo que tendrán que ser o bien de actualización o bien de modificación o de consulta; razón por la que cada proceso tendría asociadas unas tablas o registros sobre las que impacta.

Cuando los procesos impliquen cambios de cualquier naturaleza en la base, esto tendrá lugar como consecuencia de nueva información suministrada por el usuario o de la información generada por el propio proceso que es añadida a la tabla o registro correspondiente. Si la operación, en cambio, es de salida —cualquier forma de consulta, por ejemplo— se debería especificar siempre la forma de introducción de las restricciones permitidas, la forma de salida de la respuesta y el dispositivo al que envía dicha salida. Si en este último caso fuera posible elegir el destino, también se debería especificar.

II.C.3. *Subfase III: Soluciones técnicas informáticas. El software básico*

La elección del software básico que utilizaría en el desarrollo de la aplicación, como se puede imaginar, condicionaría, en gran medida, las fases sucesivas. Por esta razón, a priori, yo concedo una gran importancia al método de selección. Pero, al mismo tiempo, no considero que la elección de un lenguaje de programación u otro pueda introducir variaciones en el resultado final; de tal manera que se conviertan en fundamentales cuestio-

nes de tipo sintáctico que, en el fondo, debían estar siempre en último plano. Por otra parte, si esta decisión hubiera sido tomada con absoluta libertad, es decir, contando con todas las posibles opciones, quizá se podría atribuir más valor a lo elegido, pero la decisión cuenta siempre con la restricción de los recursos disponibles, que pueden ser amplios o no.

La decisión debería ser tomada a tres niveles, dado el modelo de referencia conceptual elegido en la fase anterior. Un primer nivel tendría que ver con el soporte de la base de datos relacional utilizada, lo que llevaría aparejado un determinado lenguaje de programación, que, por razones de agilidad en el desarrollo, debería ser un lenguaje de cuarta generación (4GL), transformable en códigos de lenguaje de bajo nivel, mediante preprocesador, pero previo a la compilación final.

El segundo nivel tiene que ver con la elección del sistema de Recuperación de Información free-text. En este caso, adquieren una especial relevancia los problemas de comunicabilidad con otros softwares, especialmente con el gestor relacional de base de datos. En relación con esta decisión, también tiene especial importancia la programabilidad del sistema de recuperación de información.

Por último, en un tercer nivel se afronta la decisión de elegir un software de bajo nivel que permita desarrollar rutinas de transformación compleja de datos o de interfaces entre ambos gestores.

Estos tres niveles de decisión, como es obvio, deben estar relacionados, en todo momento, porque las soluciones software elegidas tendrán que operar en el mismo entorno. Esta cuestión, al final, puede resultar tan determinante que condicione las elecciones realizadas en los otros dos niveles.

II.C.4. *Subfase IV: Interface hombre/máquina. Hardware y Software*

En este apartado se deben incluir una serie de consideraciones sobre los problemas de "la interfaciación" entre el usuario y la aplicación, así como las soluciones específicas que a estos problemas de relación se han dado en el desarrollo de la aplicación. Aunque soy consciente de que para dar una solución global al problema de la relación hombre-máquina a nivel de una aplicación concreta no sólo hay que plantear el problema desde la perspectiva del software, sino también del hardware, yo no considero aquí más que la primera parte de la cuestión, puesto que los aspectos relativos a la ergonomía de los dispositivos hardware, así como otros relacionados con la "usabilidad" de las máquinas, salen del objeto de este trabajo, aunque reconozco que son cuestiones de gran interés.

El procedimiento que sugiero en esta subfase es, primero, inventariar las características más sobresalientes de los interfaces de usuario utilizados en las aplicaciones bibliotecarias, para, a continuación, extraer de los estudios publicados sobre el tema la relación de prescripciones que los expertos hacen en relación con el diseño de los interfaces de usuario para programas de bibliotecas. A partir de aquí, se elaborará un conjunto de especificaciones técnicas que el interface desarrollado debería cumplir. Esta secuencia de hechos permite tomar en consideración la experiencia acumulada durante estos últimos años en el desarrollo de interfaces en los entornos bibliotecarios.

II.C.5. *Subfase V: La programación*

El proceso de programación se debe realizar por tratamientos, es decir, escribiendo las rutinas que resuelven cada tratamiento independientemente. La técnica de programación debe ser por definición de funciones, de tal forma que el esfuerzo de desarrollo se reduzca. Cada rutina debe ser escrita para ser integrada en una aplicación con las demás, pero probadas independientemente unas de otras, de tal forma que, aparentemente, todas juntas no formen una aplicación.

III

ESTRUCTURA FUNCIONAL BÁSICA DEL SIA DE UNA BIBLIOTECA

Para dar comienzo al estudio de la estructura organizacional de la biblioteca, me enfrenté con la dificultad inicial de que la gran cantidad de funciones de control informativo que se realizan en estos centros y en las que, por tanto, participan los distintos elementos que componen su estructura, me exigía partir de una estructura funcional básica que facilitara la disección del conjunto hasta el punto de que pudiera aislar entidades y relaciones, objetivo principal como queda dicho, de esta fase del trabajo.

Esta pretensión, que tenía a priori ventajas de tipo metodológico, después de consultadas la bibliografía de una parte y algunas aplicaciones de otra, se truncó en necesidad puesto que pude apreciar en la documentación consultada una gran insistencia en organizar el estudio funcional de la organización bibliotecaria en base a una serie de funciones básicas que terminaban siempre por coincidir con las estructuras que se daba a las aplicaciones integradas de gestión bibliotecaria [Reynolds 89, Tedd 88, Clayton 91, Saffady 83]. Aunque, evidentemente, la coincidencia no es absoluta entre unos autores y otros, existen paralelismos que aparecen con demasiada frecuencia para ser considerados fruto de la casualidad. Traté, entonces, de indagar un poco en los orígenes de estas coincidencias, lo que me llevó a poner de manifiesto dos cuestiones interesantes. Por un lado, la estructura interna de la organización bibliotecaria era una estructura departamental y los departamentos habituales en cualquier biblioteca coincidían en su denominación con las funciones básicas a partir de las cuales se pretendía definir el SIA de la biblioteca. Por otro lado, estas funciones eran también muy similares a las opciones básicas de cualquier SIGB, lo cual resulta aún más evidente cuanto que el sistema analizado tenga una estructura más modular.

Este conjunto de similitudes me llevó a plantearme la posibilidad de considerar como punto de partida de mi análisis una estructura funcional básica, extraída de la forma en que la documentación especializada presenta los estudios de automatización bibliotecaria. Además, analizando algunos trabajos de evaluación de SIGB que cayeron en mis manos [Mat-

thews 85, Kraft 91], pude comprobar que, en estos, se repetía una y otra vez la misma estructura funcional de partida. Pero, en cualquier caso, creo imprescindible hacer algunas puntualizaciones sobre las implicaciones que tiene partir de una estructura funcional básica en el análisis de la organización bibliotecaria:

1. Soy perfectamente consciente de que la persistencia de una opción no confiere más veracidad a la misma, no al menos desde una perspectiva científica. Pero, quiero hacer constar que, por este procedimiento, he tratado de obviar un tipo de descripción funcional sin relación alguna con la estructura real de las bibliotecas.
2. No incurro en ningún riesgo al utilizar este punto de partida, puesto que esta división funcional no condiciona en absoluto la nómina de funciones a partir de la cual irán surgiendo entidades y relaciones y, para garantizar esta falta de condicionamiento, no consideraré, en ningún momento, a las funciones básicas como compartimentos estancos, de tal forma que dedicaré oportunamente atención a los problemas relacionados con la conexión entre subfunciones de una y otra función básica, de lo contrario, el concepto de integridad del sistema se vería perjudicado.

La estructura funcional básica, a partir de la cual voy a trabajar, consta de cinco funciones: Adquisiciones, Catalogación, Circulación, Control de publicaciones periódicas y Referencia. Como se recordará, lo que sigue es una descripción, función tras función, de las operaciones mecanizables que contienen, para extraer de ellas la estructura de entidades y relaciones que le corresponde al conjunto.

III.A. Función de adquisiciones

El proceso general que denominamos «función de adquisiciones» se puede encontrar implícito en la siguiente descripción: La selección de los materiales que se van a adquirir puede partir de iniciativas de diferentes personas: del personal bibliotecario, de la institución que financia el centro, de los usuarios, etc. Todos ellos presentan propuestas de adquisición tras consultar distintas fuentes, tales como revistas científicas, bibliográficas, catálogos y relaciones de novedades preparadas por los distribuidores. El personal de la biblioteca es quien tramita las propuestas y se encarga de llevar a cabo las compras. Para ello, comienza por comprobar si los materiales cuya compra se ha aprobado, se encuentran ya presentes entre los fondos de la biblioteca o en anteriores pedidos. A continuación, reúne to-

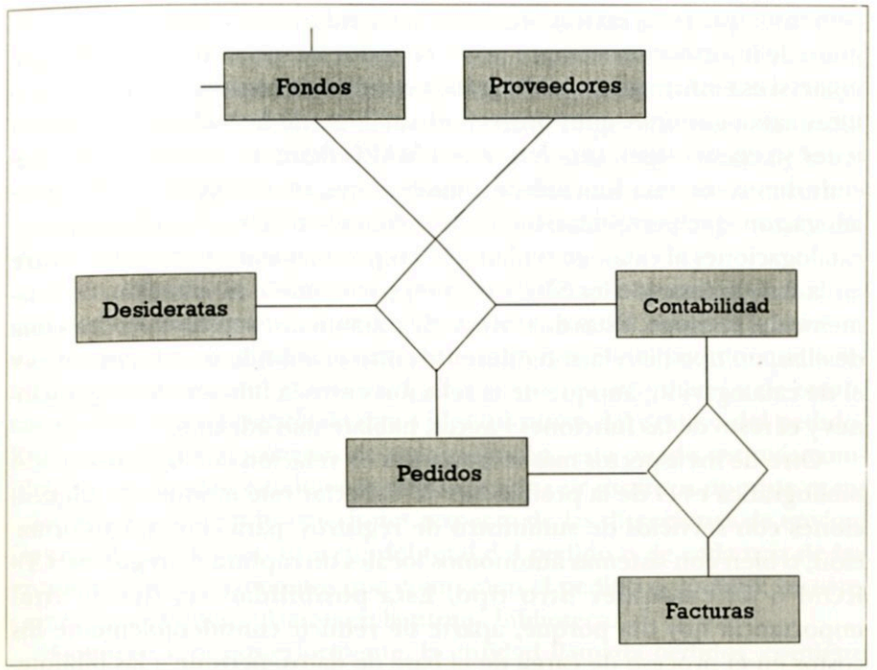


Gráfico 1: *Adquisiciones.*

dos los datos necesarios para hacer la petición, selecciona el distribuidor y redacta la orden de compra. Después de cierto tiempo, es necesario remitir reclamaciones o anulaciones de pedidos. Recibidos los materiales, que pueden ser libros, publicaciones seriadas, microfilms, microfichas, grabaciones, manuscritos, folletos, etc., se ha de comprobar si concuerdan o no con lo que se solicitó y con las facturas que llegan. Se realiza el pago y se actualiza la contabilidad. Los nuevos fondos también pueden provenir de donaciones al centro o como fruto de convenios con determinadas instituciones. En cualquier caso, el material recibido se registra y pasa al departamento de catalogación.

Para llevar a cabo este proceso general que acabo de describir, es necesario poner en juego un conjunto de cuatro informaciones básicas o entidades y una quinta de carácter secundario [Grieder 78]. Esas cinco informaciones básicas son: Las informaciones bibliográficas, de proveedores, de pedidos, de facturación y contables. La información secundaria adicional es la información de desideratas que, a partir de este momento, serán consideradas como entidades de la función de adquisiciones.

En cuanto a la llamada información bibliográfica, evidentemente, se refiere al conjunto de datos que forman parte de las referencias de los do-

cumentos que están en trámite de ser adquiridos. Con respecto a este conjunto de información siempre existe en todos los sistemas un problema similar: si esa información bibliográfica es utilizada como germen de las futuras catalogaciones que integren el catálogo de la biblioteca, deberán tener ya en su origen una estructura MARC definida, lo que obliga a incluir dentro de esta función de adquisiciones, una subfunción de precatalogación, que permitirá o bien la incorporación automática de estas precatalogaciones al catálogo o bien que las precatalogaciones, como ocurre en la mayor parte de los SIGB, formen parte, desde su creación, directamente del catálogo. Estas dos formas de tratamiento determinan, cada una de ellas, un tipo de relación diferente entre el módulo de adquisiciones y el de catalogación, aunque de la relación entre la función de adquisiciones y el resto de las funciones básicas, hablaré más adelante.

Otro de los aspectos más importantes en relación con la información bibliográfica es el de la posibilidad de conectar este módulo de adquisiciones con servicios de suministro de registros, para capturar información, o bien con sistemas autónomos locales de captura de registros, CD-ROM o de cualquier otro tipo. Esta posibilidad resulta de vital importancia hoy día porque, aparte de reducir considerablemente los costos en el proceso de carga de la base de datos, permite a las bibliotecas servirse de las facilidades que les proporcionan los distribuidores de documentos [Pearson 75].

El segundo tipo de información es la de los proveedores. Estos, como es sabido, actúan como receptores de los pedidos y esto implica que, muchas veces las bibliotecas se vean obligadas a seguir las normas de funcionamiento de estos proveedores, lo que supone una serie de ventajas mutuas. Por lo que estos proveedores deben estar consignados en la base de datos de la biblioteca con un perfil de funcionamiento lo más completo posible. Por este motivo, junto a los datos de referencia mínimos, como nombre, dirección, teléfonos, etc., es necesario, además, consignar en el registro de cada proveedor una serie de datos que conforman su perfil, como, por ejemplo, plan de descuentos, ciclos de reclamaciones, tipo de proveedor, etc. Estas informaciones permitirán al sistema hacer una serie de previsiones en relación con los pedidos que se dirijan a cada uno de estos proveedores con su correspondiente perfil.

Todos los datos que conforman el registro del proveedor se pueden dividir en dos grupos: unos, solamente tienen una función descriptiva y se utilizan, fundamentalmente, para identificar al proveedor. Otros, en cambio, servirán para realizar procesos de actualización de la información por medios automáticos, como, por ejemplo, prever qué cantidad puede costar un determinado documento, aún no sabiéndose de antemano, en base,

sencillamente, al precio medio de los documentos pedidos a ese proveedor.

La tercera entidad es la de pedidos. La información de pedidos es una información que combina diversos elementos. El pedido, por definición, debe ser el resultante de la relación entre la entidad bibliográfica y la entidad proveedor, uniendo a estas dos entidades una parte de la información contable, concretamente la que se refiere a las partidas presupuestarias.

Pero, además de estos elementos de relación que aparecen en la entidad pedido, existen otra serie de datos, como son los relativos a la información de envío y la información de manipulación del pedido, que también forman parte de esta entidad. En cuanto a la información de envío, consiste en una referencia de datos identificativos del destino del pedido. En cuanto a la información de manipulación, esta puede ser más compleja, pero, fundamentalmente está formada por distintos tipos de mensajes, que contienen instrucciones respecto de las direcciones de envío y facturación, mensajes respecto del total del pedido o de cada una de las entradas de los documentos que componen el pedido o, incluso, instrucciones para la manipulación en la propia biblioteca.

Como se verá posteriormente, la entidad llamada pedidos, contiene una serie de informaciones que son el fruto de la relación entre otras entidades, así como otras informaciones que deben ser consignadas específicamente por el usuario en el momento de la tramitación del pedido. Este tipo de entidades mixtas son muy corrientes, contienen información de su relación con otras entidades e información suministrada por el usuario en el curso de la utilización del sistema.

La cuarta entidad es la de facturación. Contiene básicamente datos provenientes de la entidad pedido, como los datos que identifica el/los documentos, es decir, los datos de información bibliográfica, así como los datos relativos al proveedor, la partida presupuestaria y, junto a eso, las informaciones propias que identifican una factura cualquiera, especialmente por lo que se refiere al destino de la misma.

La información de facturación, a su vez, se utilizará por la entidad de contabilidad, en la misma medida en que ésta utiliza las informaciones contenidas en la entidad «pedido». Pero es importante saber que en ciertos datos provenientes de la entidad pedido debe existir necesariamente una modificación por intervención del usuario, puesto que, en ocasiones, cuando se tramita un pedido no se conoce el precio exacto del documento y se hace una previsión que, luego, en el momento de la facturación, deberá ser ajustada con carácter definitivo. Como veremos después, tanto un proceso como otro tienen su correspondiente repercusión en la contabilidad.

La entidad de contabilidad se usa fundamentalmente para identificar cada partida presupuestaria o presupuesto de la biblioteca. Junto a estos datos contables deben aparecer una serie de informaciones destinadas a poder hacer una selección mediante otros parámetros —algunos de ellos bibliográficos y otros definidos por el usuario— en relación con la administración del presupuesto de la biblioteca.

La entidad contable, normalmente está sujeta a unas restricciones, que resultaría prolijo e innecesario describir aquí, y que tienen que ver con la estructura presupuestaria permitida por el sistema: número de cuentas, número de dígitos por cuentas, número de transacciones por apartado contable (operación), etc.

En cuanto a la quinta entidad, la entidad secundaria, llamada de desideratas, considero que no es imprescindible para el funcionamiento del sistema de adquisiciones, a diferencia de lo que ocurre con las entidades anteriores, pero viene siendo una práctica habitual que las bibliotecas hagan una gestión, previa al trámite de la adquisición, de las desideratas que reciben, y sea a partir de este conjunto de informaciones bibliográficas, a partir del punto del que seleccionan los documentos para iniciarse el trámite de su adquisición. En este sentido, la entidad desideratas contendrá las informaciones necesarias para referenciar documentos y esto, desde luego, sin que estas referencias deban estar incluidas en el catálogo ni tampoco tengan una estructura de formato estándar.

Una vez descritas las cinco entidades que componen el subsistema de adquisiciones, empezaré por dar algunos datos sobre el proceso general de tramitación de las adquisiciones.

En principio, tengo que decir que la función de adquisiciones en algunos casos se podría solapar en los procedimientos que la integran con la función básica de Control de Publicaciones Periódicas. Especialmente, por lo que afecta al trámite de las suscripciones de las publicaciones periódicas y sus posteriores renovaciones. La solución que he utilizado es la de considerar que el trámite de la suscripción inicial y sus posteriores renovaciones son un caso particular de adquisición de documentos y, por tanto, forman parte de este subsistema de adquisiciones que estoy describiendo.

Por otro lado, los procesos de entrada y actualización de la base de datos en el subsistema de adquisiciones deben permitir la emisión y cancelación de pedidos, la modificación de los mismos, así como, por supuesto, la actualización y modificación de la información bibliográfica y todo ello mediante la intervención directa del usuario.

Indudablemente, para poder realizar estas operaciones, la localización de las informaciones previas es un proceso que se debe poder realizar con la mayor agilidad posible. Por este motivo, cuantas mayores prestaciones

tenga el subsistema en lo que se refiere a la búsqueda de información, más eficaz será el conjunto.

El proceso básico que se realiza en el subsistema de adquisiciones es el proceso llamado «pedido», alrededor del cual giran una serie de procesos auxiliares o subprocesos; pero es el proceso del pedido —en sus distintas fases— el que condiciona el funcionamiento del conjunto. Esto quiere decir que, por una parte, una gran variedad de tipos de pedidos y, por otra, un gran control del estado de esos pedidos, permitirán a los usuarios del sistema estar permanentemente bien informados de la situación de esos pedidos y de las consecuencias de la tramitación de los mismos.

Los distintos procesos que pueden dar lugar a la tramitación de un pedido se determinan a partir de la llamada tipología de adquisiciones que servirá como indicador de las posibilidades del sistema en este aspecto. Es importante señalar que esta tipología supone la realización de procesos diferentes en los que se ven afectados las distintas entidades y relaciones en función de cada uno de los tipos. Algunos de esos procesos son lo bastante complejos como para que tengan que estar definidos con toda precisión en el sistema. Lo que quiere decir que un subsistema de adquisiciones abierto por lo que afecta a la tipología de pedidos es muy poco factible hoy día, no tanto porque no sea posible definir nuevos tipos, sino porque lo que no es fácil es definir los procesos asociados a cada uno de esos tipos.

Una relación de posibles tipos podría ser: Pedido nuevo, urgente, lista de documentos seleccionados, donación, canje, pedido telefónico, a vista, pedidos permanentes, prolongación de pedidos, telepedidos, pago anticipado, suscripciones, pago aplazado, etc. [Matthews 85, Reynolds 89].

Como he insinuado antes, creo que la clave de los procesos que se realizan en el subsistema de adquisiciones es el pedido, y el momento de su creación determina, en gran medida, las posibilidades que el sistema tendrá de controlar las informaciones relacionadas con este pedido.

El pedido se puede crear a partir de entradas existentes en la lista de desideratas. Evidentemente, puede ser simple o múltiple, en función del numero de documentos que contenga y debe ser posible controlar múltiples direcciones de envío por pedidos, para el caso de bibliotecas con depósitos descentralizados o redes de bibliotecas controladas por un solo sistema. Debe ser también posible asociar múltiples partidas presupuestarias a un solo pedido.

Aunque la generación automática de pedidos a partir de otros ya existentes para agilizar el proceso de su generación es una función muy deseada normalmente en las bibliotecas, una de las dos misiones fundamentales del sistema es la del control de los estados de los pedidos. El sistema debe ser capaz de controlar tanto los llamados «estados externos» como los internos.

Los estados externos son los que tienen lugar hasta el momento en que se realiza la recepción del documento y los internos son los que siguen a la recepción del documento hasta su catalogación definitiva. El sistema deberá poder informarnos en todo momento de cuál es el estado de ese documento desde que se pide (cuando la desiderata se convierte en pedido) hasta que se recepciona. Los posibles estados, como se puede imaginar, tienen que ver con reclamaciones, retrasos por parte del proveedor, los trámites de facturación, para, después, en el control de los estados internos, pasar al proceso que se sigue en los distintos departamentos que procesan técnicamente el documento hasta la catalogación definitiva del mismo. Los posibles incidentes producidos en cada uno de los estados deben ser convenientemente avisados por el sistema. El sistema, asimismo, en el momento de la creación del pedido, calculará automáticamente precios probables y efectuará los descuentos en los presupuestos indicados por el usuario de esas cantidades probables. En el caso de que algunas de las partidas presupuestarias, como consecuencia de la realización de un pedido, esté próxima a su liquidación, el sistema deberá avisar de este incidente. Igualmente, cuando los pedidos excedan de una determinada cantidad, será preciso utilizar una autorización específica para la tramitación de pedidos de esa naturaleza. Por último, el sistema debe permitir, también, la realización de pedidos en reserva para su tramitación posterior.

Una vez emitido el pedido se pasa a las dos fases siguientes que hacen referencia a dos estados posteriores de dicho pedido, que son la recepción y la facturación.

Evidentemente, tanto en el momento de la recepción como en el de la facturación se debe poder controlar una serie de informaciones. Especialmente en el caso de la facturación esas informaciones son informaciones contables, puesto que en ese momento ya se conocerá el precio definitivo del documento y, por tanto, se podrá repercutir en la contabilidad de manera más definitiva el efecto de esa facturación. En este momento, si no se ha elegido anteriormente, se puede elegir la forma de pago, que puede ser diversa y que implica, así mismo, diferentes tipos de repercusión contable.

En cuanto a la recepción, se puede hacer de forma íntegra o parcial, tanto en lo que se refiere a ejemplares como a títulos. Lo mismo por lo que afecta a la factura que puede llegar con el pedido, antes del pedido o después del pedido y, en cualquier caso, debe ser posible su tramitación.

Un caso particular es la subfunción de adquisiciones que regula la renovación de las subscripciones de la Publicaciones Periódicas. Esta renovación se realiza como un tipo más de adquisición que tiene carácter cíclico y automático. Algunos sistemas fueron diseñados de tal manera que no ponga en marcha automáticamente esa renovación, pero también el

sistema debe ser capaz de renovar automáticamente dichas suscripciones. El procedimiento, por defecto, establecido, debe ser la renovación con arreglo a las condiciones establecidas en la suscripción anterior, pero, evidentemente, dichas condiciones pueden variar y el sistema debe permitir la modificación de las condiciones de una manera cómoda y ágil.

Un capítulo especial merecen, dentro del subsistema de adquisiciones, los productos impresos. No voy a hacer aquí una relación exhaustiva de los productos impresos necesarios en el trámite de las adquisiciones, solo diré que son difícilmente previsibles, es decir, algunos productos impresos son obvios, como las cartas de pedido, de reclamación, etc., pero, en otras ocasiones, especialmente por lo que se refiere a los listados que se pudieran producir, la necesidad del producto está muy condicionada por intereses específicos de cada usuario del sistema. En ese sentido, creo que el sistema debe permitir la emisión de productos impresos de forma selectiva a instancias de condiciones impuestas en cada momento por el usuario y, así, dejar esta opción lo más abierta posible, para que sean los propios usuarios quienes determinen los tipos de pedidos que necesitan.

Por último, me gustaría mencionar algunos de los problemas relacionados con la conexión del subsistema de adquisiciones con otras funciones básicas [Boss 82]. Por una parte, el status de adquisiciones de los documentos debe ser una información accesible desde el subsistema de referencia y del de circulación, puesto que puede ser una información importante para cada uno de lo usuarios de los distintos subsistemas. Así, por ejemplo, si un usuario encuentra una referencia de un documento formando parte del catálogo y no sabe que ese es un documento que está en proceso de pedido y que no ha sido aún recibido, encontrar esa referencia en el catálogo podría inducirle a confusión si no aparece una información clara sobre el estado de adquisición del documento. Por la misma razón es importante que el sistema controle muy bien la posibilidad de hacer reservas de circulación a documentos que están en trámite de adquisición. Esta cuestión debe ser decidida previamente por la biblioteca y el sistema debe estar en condiciones de realizar ese control en base a la decisión que la biblioteca haya tomado.

III.B. La función de catalogación

En cualquier organización bibliotecaria, la función de catalogación básicamente consiste en la generación del instrumento de acceso a los documentos que se pondrá a disposición de los usuarios. Este instrumento de acceso se denomina catálogo.

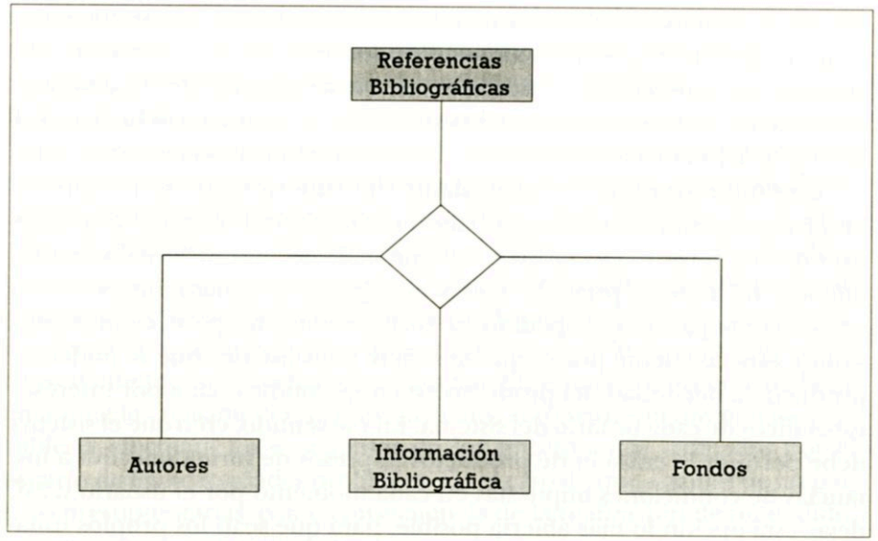


Gráfico 2: *Catalogación*

Esta operación de generación del catálogo se divide en dos fases: En una primera fase se crean las descripciones bibliográficas que referencian los documentos que posee la biblioteca y en la segunda se añaden lo que llamaremos informaciones locales, que normalmente son informaciones incluidas por la biblioteca junto a la descripción bibliográfica para localizar el ejemplar físico del documento con el fin de poder suministrarlo a los usuarios que lo soliciten.

Estas dos operaciones realizadas convenientemente, permiten la generación de lo que será la herramienta fundamental de acceso a los documentos de que dispondrán los usuarios de la biblioteca.

Para la realización, especialmente, de la primera fase del proceso, existen normativas aceptadas nacional e internacionalmente, que constituyen un estándar para la realización de las catalogaciones. Me refiero, naturalmente, a la ISBD, como normativa internacional y a las Reglas de Catalogación Españolas o las Reglas de Catalogación Angloamericanas, en cuanto a las normativas nacionales. Sin embargo, esto no implica que las descripciones realizadas por distintas bibliotecas del mismo documento vayan a ser iguales, especialmente en ciertos campos, como son los campos de materia, por mucho que se estén utilizando listas de encabezamientos de materias comunes. Precisamente por este motivo, muchas veces, las bibliotecas suelen recurrir a catalogaciones hechas por bibliotecas prestigiosas y, o bien copian en todo o en parte esas catalogaciones o simplemente las incorporan a su propio catálogo.

Como se verá, la realización de la catalogación en sus dos fases, implica el manejo de tres conjuntos de informaciones bien distintas, que darán lugar a las tres entidades básicas gestionadas desde la función de catalogación:

— la información de autoridades
— la información bibliográfica
— la información de fondos.

Las características de cada una de estas informaciones, a diferencia de lo que ocurría con las entidades adscritas a la función de adquisiciones, se ajustan a formatos definidos de manera estándar por los organismos competentes. Esto quiere decir que la definición de propiedades y atributos de estas tres entidades se hará con arreglo a un patrón conocido, que se denomina patrón MARC [Usmarc 88] y que, en el caso de España, su denominación es IBERMARC [Ibermarc].

III.B.1. *Entidad Autoridades*

Es una entidad que recogerá la información asociada con los encabezamientos principales y secundarios de nombres (personales, corporativos, congresos, etc.), de títulos uniformes, de materias y de clasificaciones. Estos tipos de autoridades tienen definidos sus correspondientes modelos MARC y deben ser considerados, en el conjunto del sistema, como entidades independientes aunque de funcionamiento similar.

III.B.2. *Entidad Información Bibliográfica*

Existen distintos modelos, según los diferentes tipos de materiales que el centro procesa: monografías, series, materiales cartográficos, registros sonoros, videograbaciones, fondo antiguo, en el caso de IBERMARC. A pesar de todo resulta factible la definición de todos los tipos de materiales en una sola entidad, de tal manera que, el conjunto de información bibliográfica manejada por una o varias bibliotecas sea único, esto tendrá la ventaja de que si el usuario quisiera diferenciar por tipos de materiales podría hacerlo y si no quisiera hacerlo, también podría ser. En el caso de que estuvieran separados el asunto sería bastante más complicado, especialmente en el caso de que no se quisiera diferenciar por tipo de materiales.

También hay que decir que como en el proceso de catalogación en su primera fase, descrito anteriormente, implica tanto a la información de

autoridades como a la información bibliográfica, es evidente que tendrán que existir unas relaciones bastante estrechas entre la información de autoridades y la información bibliográfica, hasta tal punto que las descripciones bibliográficas como tal serán, en realidad, una entidad virtual formada por la relación entre autoridades e información bibliográfica.

III.B.3. *Entidad Fondos*

Contiene la información relativa a las existencias de documentos en la biblioteca en cuestión. Esta información también debe estar en la información bibliográfica y cada una de sus entradas hará referencia a un ejemplar físico del documento contendido en la biblioteca.

Una vez definidas las entidades básicas que intervienen en el proceso de la catalogación, paso a definir un conjunto de subfunciones que se desarrollan en el curso del proceso de actualización de la base de datos catalográfica.

Los modelos de referencia para las catalogaciones, como he dicho antes, se ajustan a normas nacionales. Estas normas, frecuentemente, están sujetas a cambios, lo que plantea problemas de adaptación de los sistemas de catalogación a esa nueva normativa. Especialmente, por lo que hace referencia a la compatibilidad de las catalogaciones, que se van haciendo día a día, con el conjunto de catalogaciones existentes anteriormente. Por ello, las entidades, tanto de autoridades, como de información bibliográfica y de fondos, deben ser adaptables por los usuarios, de tal forma que exista la posibilidad, en todo momento, de mantener el catálogo de acuerdo con la última versión de las normas vigentes.

Por otra parte, por lo que afecta a la información bibliográfica, deberá, también, existir la posibilidad de adaptar esta entidad a la aparición de modelos MARC para nuevos tipos de materiales, como, por ejemplo, el caso de los ficheros de ordenador (CF) o el material de archivos (AMC). En este mismo sentido es importante tener en cuenta al trabajar con formatos estándar aspectos tales como el del «set» de caracteres utilizado en el almacenamiento de la información, puesto que existen también normas al respecto que deben ser cumplidas. (Véanse, por ejemplo, las normas ISO 646, 5426, 6630)

La definición, por tanto, del formato de referencia bibliográfica, implica la adaptación de este formato a una norma conocida. También implica la definición de un conjunto de información recuperable frente a otro conjunto de información no recuperable. Es decir, en todo formato bibliográfico hay una serie de datos recuperables en una fase posterior por los usuarios y otros datos que tienen un valor puramente descriptivo del

documento. Pues bien, por este motivo, es necesario en el sistema, definir cual va a ser la información recuperable y cuál la no recuperable.

En cuanto a la entidad de autoridades, será necesario tener en cuenta, lo mismo que en las otras entidades, que la carga de información correspondiente a las autoridades podrá realizarse tanto de forma on line como batch. Al mismo tiempo, el sistema debe permitir el control de referencias cruzadas, es decir, la introducción de referencia cruzadas entre las entradas de autoridades y la contemplación de esas referencias en el momento de la consulta, de tal forma que el propio sistema tenga en cuenta esas referencias cuando el usuario pretenda hacer una consulta. Todo ello, por supuesto, sin olvidar que la estructura básica de la información de autoridades debe ajustarse a la norma MARC de autoridades, en este caso la norma IBERMARC.

Los tipos básicos de autoridades considerados aquí serán los que definen como tales las GARE [Gare 84], es decir, los llamados nombres, las materias y los títulos uniformes. Pero me gustaría hacer aquí una referencia a algo que no se puede considerar una información de autoridades, las entradas de clasificación. Recientemente, la Biblioteca del Congreso ha publicado una norma USMARC de clasificaciones [Usclass 91], convirtiendo este tipo de entradas en una información tratable en los mismos términos que el resto de la información de autoridades.

En general las autoridades son utilizadas como clave primaria para la realización de ordenaciones y es la información básica de punto de acceso a las informaciones bibliográficas que contiene el catálogo. Por este motivo el sistema debe realizar un tratamiento muy especial de estas entradas de autoridades, de tal forma que, a partir de los datos que el usuario suministra en el momento de la catalogación, el sistema deberá generar entradas equivalentes para la ordenación, visualización e impresión de estas autoridades. Esto implica, habitualmente, una combinación de procesos, unos relacionados con el tratamiento de ciertos códigos, incluidos en el formato MARC, otros, mediante el tratamiento de set de caracteres y, por fin, otros que tienen que ver con el funcionamiento de determinados dispositivos de visualización, impresión, etc.

En relación con esto, de cara a la visualización en diferentes formatos de entrada de autoridades es importante tener en cuenta que la información de control de los campos MARC —indicadores e identificadores— debe aparecer oculta cuando el formato de visualización elegido sea uno no MARC. Quizá el más interesante para los registros de autoridades, como alternativa al MARC, sea el formato «thesaurus», aunque también, la Biblioteca del Congreso utiliza uno que llama formato ficha y la propia IFLA ha definido uno que podemos llamar formato GARE [Gare 84].

La entrada de información de autoridades en el catálogo, indudablemente se realiza en el proceso de catalogación; lo que implica la existencia de un control on line contra ficheros, preferiblemente «kwic». Esto debe estar ligado a un proceso de validación de las entradas contra información externa. Esta información externa se puede encontrar almacenada localmente, fundamentalmente en CD-ROM, o puede ser información remota. En ese sentido, quizá convenga recordar aquí los dos sistemas más utilizados en el mundo para esta función, que son el LSP y el WLN.

El sistema, por otra parte, tendrá que generar un mecanismo de permutación de las entradas de autoridades que sea selectivo, a nivel de subcampo y los subcampos que sean permutados deben ser definibles por el usuario. Es decir, el usuario debe tener la capacidad de elegir, de entre los distintos campos que forman cada registro de autoridades, aquellos subcampos que el sistema permutará. Por otra parte, como complemento de las subfunciones relacionadas con la gestión de autoridades, debe haber una serie de subfunciones de mantenimiento de las autoridades, como por ejemplo, la posibilidad de reemplazar autoridades obsoletas en los registros afectados, sin necesidad de modificar uno a uno todos los registros, la posibilidad de realizar trasvases automáticos de referencias asociadas de una autoridad a otra, etc.

Pasando directamente a la creación y modificación de las catalogaciones, es necesario tener en cuenta que esta operación se realizara con mucha frecuencia, casi constantemente se introducirán y modificarán registros, lo que obliga a que el proceso de entrada de datos sea lo más amigable posible. Un buen sistema puede ser el de pantallas formateadas, que permite que la relación con las entidades de autoridades se realice de forma transparente para el usuario, sin que tenga la sensación de estar saliendo de un proceso para entrar en otro distinto, sino que todo forma parte del mismo proceso. Ello supondrá, entonces, que se podrá hacer una validación automática de las autoridades y que se podrán añadir autoridades en el curso de la catalogación. Pero como es indudable que con frecuencia se modificarán las catalogaciones, debe ser posible aprovechar todas las prestaciones de recuperación que el sistema ofrezca para localizar registros que luego podrán ser modificados y estas modificaciones se deberán poder hacer utilizando funciones de edición avanzadas, las mismas que estén disponibles en el momento de la creación de los registros. Estas funciones de edición, en el caso de las modificaciones, permitirán añadir campos, rectificar o suprimir cadenas de caracteres dentro de los subcampos o los campos, o eliminar registros, campos o subcampos, indistintamente.

Como culminación del proceso de catalogación el sistema debería contar con un procedimiento automático de verificación de la consistencia de las catalogaciones, de tal manera que el sistema pueda detectar posibles errores en la introducción de los datos. En este sentido la combinación de información codificada e información textual dentro de los registros MARC, suponen una gran ayuda, ya que al sistema le será fácil detectar la existencia de incompatibilidades entre determinadas informaciones codificadas y textuales. Pero, como dije en el caso de la información de autoridades, la carga de los datos habrá que hacerla unas veces en tiempo real y otras, mediante procesos en diferido. Ello quiere decir que deberán permitirse las dos operaciones. Significa, por tanto, que estará disponible una subfunción de carga externa en sus distintas modalidades: Carga de información externa remota, mediante la conexión de servicios de suministro de registro, o la carga de información externa en modo local, con sus dos variantes básicas: carga de información a partir de un «pool bibliográfico» o carga de información a partir de sistemas autónomos del tipo CD-ROM [Gredley 90].

Todo el sistema de catalogación se desarrollara exclusivamente para satisfacer unas necesidades, como dije al principio, de recuperación de la información. Estas necesidades de recuperación de información deben ser satisfechas por el sistema en los siguientes términos: de todos los sistemas de recuperación probados para la información bibliográfica, el sistema que permite la recuperación por palabras claves mediante la creación de set de búsquedas es el más adecuado para la información textual que contienen las descripciones bibliográficas. Ello implica, por supuesto, la posibilidad de utilizar operadores lógicos y de continuidad, el almacenamiento para usos posteriores de perfiles de búsqueda y todo esto, a partir de la existencia de unos puntos de búsqueda que podemos dividir en dos grupos:

— los puntos de acceso de identificador único.
— los puntos de acceso de identificador bibliográfico o textual; lo que significa múltiples identificadores en la misma entrada.

Es precisamente la existencia de puntos de acceso de este tipo, nombres, títulos, materias, editor, etc., lo que obliga a utilizar un sistema de recuperación basado en palabras clave. Al mismo tiempo, junto a los puntos de acceso tradicionales, en cualquier sistema de bibliotecas, es necesario que, al menos, exista la posibilidad de limitar las búsquedas por una serie de informaciones que tradicionalmente no se consideran de punto de acceso; informaciones tales como, la lengua, el país, la fecha de publicación, etc. Por supuesto, entre los puntos de acceso deben estar incluidos aquellos que tienen que ver con la información no sólo de autoridades y bibliográfica, sino también de fondos. Por otra parte, el sistema de recupe-

ración debe proporcionar toda la información posible sobre la frecuencia de utilización de las distintas entradas con el fin de facilitar la utilización de técnicas avanzadas de recuperación de información.

El sistema de recuperación de cara al usuario debe ser variado. Es decir, no utilizará una sola modalidad de «interface de usuario», sino que, de forma combinada, podría utilizar un sistema de menús y un sistema de comandos, adaptables ambos al distinto conocimiento que los usuarios tengan de los mismos.

La visualización de la información debe poder ser presentada en formatos diversos, también en función del tipo de usuario. Estos formatos podrían ir desde el formato MARC hasta formatos etiquetados, con mayor o menor cantidad de información, pasando, por su puesto, por la visualización de formatos ISBD o ISO en sus distintas versiones.

En cuanto a la información de fondos, que el sistema debe controlar, decir lo mismo que dije anteriormente de la información de autoridades y de la información bibliográfica, que es una información que debe ajustarse también a los formatos normalizados y relacionada con la información bibliográfica. Pero, al mismo tiempo, debe contener ciertos datos importantes de cara a la posibilidad de desarrollar otras funciones. Me estoy refiriendo a las informaciones que tienen que ver con el tratamiento de cara al préstamo que recibirá cada ejemplar físico contenido en los depósitos de la biblioteca y el tratamiento en el control de publicaciones periódicas que recibirá cada fascículo de una revista. Este tipo de informaciones debe formar parte de la información de fondos para facilitar el desarrollo de los puentes necesarios entre las funciones de catalogación y las funciones de circulación y control de publicaciones periódicas.

En cuanto a la relación de la información de fondos con la información bibliográfica, he de decir que esta relación está prevista en los propios formatos estándar y por consiguiente no debería plantear ningún problema.

Como es lógico, habrá también una serie de subfunciones destinadas al mantenimiento general del sistema y al control de la información catalográfica. Una de ellas podría ser la de acceso selectivo a la información según perfiles de autorización por categorías de usuarios, lo que facilitaría la modificación o eliminación de las informaciones del catálogo con restricciones y permitiría también el control de las informaciones relacionadas, ya que, como se puede ver, una de las claves del funcionamiento de un sistema catalográfico integrado como este es el mantenimiento de la coherencia de las relaciones entre las distintas entidades. Por consiguiente, los mecanismos de seguridad, en este sentido, tienen gran importancia.

Por último, me gustaría plantear, aunque fuera muy someramente, alguna cuestión en relación con la posibilidad del uso compartido de la función de catalogación por distintos sistemas bibliotecarios. Es difícil, aunque hablemos de una sola biblioteca, que no se plantee la necesidad de una cierta descentralización de los depósitos. Tanto más si hablamos de diferentes bibliotecas conectadas a un sólo sistema. Esto nos lleva a plantearnos el problema de la compartición de la información bibliográfica y de autoridades y del uso particular no compartido de las informaciones locales relativas a los fondos. Esta posibilidad debe existir tanto si el sistema es utilizado por una sola biblioteca como si es utilizado por varias bibliotecas. Por otra parte, ello implica que cada una de las bibliotecas deberá conservar variantes de la información bibliográfica y de autoridades, al mismo tiempo que utiliza la información común, que seria una información de catálogo colectivo. Por otra parte, este sistema debe estar en condiciones de suministrar información de fondos del conjunto o bien de las partes, según las peticiones que hagan los usuarios.

Como ya hice anteriormente en la función de adquisiciones, termino diciendo alguna cosa de los productos de salida, que, en este caso, no solo son productos impresos, sino también en soportes magnéticos. Los productos impresos relacionados con la catalogación, normalmente, son los propios catálogos impresos en sus diferentes variantes que exigirán procesos específicos para la generación de esos catálogos, ya que la transformación de las estructuras MARC que el sistema maneje en descripciones de formato ISBD y ajustadas a las reglas de catalogación española exige desarrollos «ad hoc» que deben estar muy ligados a las especificidades de los formatos de origen y de destino.

III.C La función de circulación

Los fondos de los centros bibliotecarios, con frecuencia, suelen ser trasladados dentro y/o fuera del propio recinto de forma temporal. No sólo cuando son utilizados por los usuarios de forma directa, sino también en prestamos interbibliotecarios [Iso 10160, 10161] o con motivo de su envío al departamento de encuadernación o restauración, etc. Es necesario que el personal de la biblioteca tenga actualizada la información sobre cada material que ha sido retirado o ha sido trasladado a un sitio distinto del habitual. El control de esta información que afecta a los documentos en circulación, esta formado por un conjunto de subfunciones que, agrupadas, forman la función básica de control de la circulación. Las entidades informativas que entran en juego en la realización de estas subfunciones serían tres entidades básicas y una derivada. De las tres básicas, dos

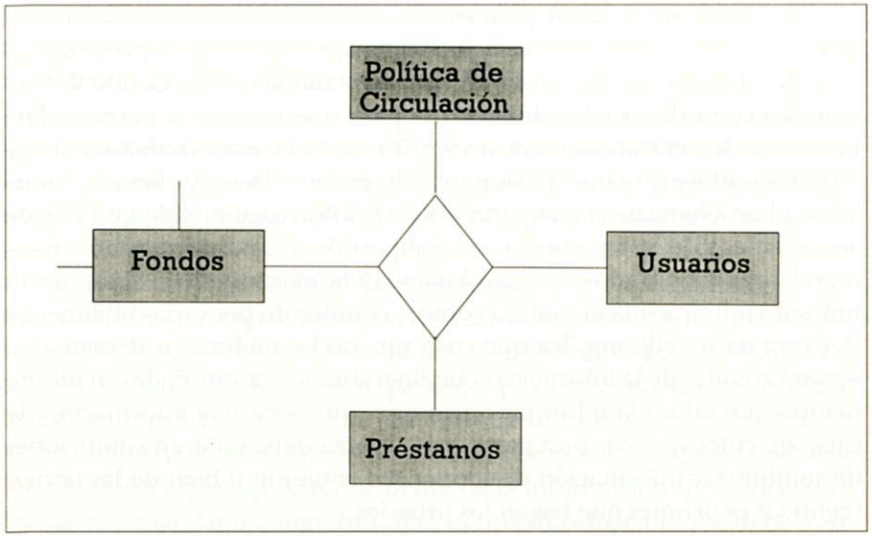


Gráfico 3: *Circulación.*

tendrán que entrar en relación permanentemente, que son la entidad de información bibliográfica, ya descrita en funciones anteriores, y, por otro lado, la entidad de usuarios. En realidad, todas las tareas de control de la circulación que describiré a continuación, se controlarán, desde un punto de vista informativo, mediante la puesta en relación de elementos de estas dos entidades informativas. Pero, esta relación, no se puede realizar de forma anárquica e indiscriminada, sino que, como veremos después, tendrá que realizarse a través una serie de reglas. Estas reglas que permiten y que regulan la relación entre informaciones bibliográficas y usuarios se denominan «política de circulación de la biblioteca» y forman la tercera entidad básica de esta función.

A efectos de diseño de una estructura de datos como la que estamos describiendo aquí, esta relación es de grado tres, es decir, una relación entre tres entidades diferentes que darán lugar a una cuarta entidad derivada de esta relación, llamada entidad de préstamos, o quizá, de manera más práctica, la podríamos denominar «entidad de circulación», puesto que la denominación préstamo recoge solamente un caso de los muchos tipos de circulación que se pueden dar en una biblioteca. Esta entidad es derivada, como dije anteriormente, en un doble sentido, por una parte las informaciones que contendrá la entidad derivada son en su mayor parte extraídas de las entidades primarias que le dan lugar. Por otro lado, se trata de una entidad subordinada a las otras porque sólo es posible añadir información a la entidad de circulación con ocasión de la puesta en relación de las entidades básicas. Algunas de las subfunciones que son causa directa

de relaciones como las que acabo de mencionar serán descritas a continuación.

La subfunción de «salida de documentos» es un caso típico de relación entre información bibliográfica e información de usuarios con la mediación de la política de circulación. Solamente si la política de circulación ha sido diseñada convenientemente será posible que el sistema controle con eficacia cada una de las salidas de documentos. En realidad, la salida de documentos se convierte en la ocasión para comprobar si ese usuario está autorizado a llevarse ese documento en concreto. Pero ésta es sólo una parte de la operación, ya que habría que añadir, además, si no es autorizado, por qué, y, si lo estuviera, hasta cuándo (fecha de devolución). Esta operación de préstamo, lo mismo que la de devolución, puede ser realizada en tiempo real, es decir, en el momento en el que el usuario en cuestión quiere disponer de ese documento, se realiza la comprobación informativa que acabo de describir, pero, también puede ser realizada en diferido, lo que, indudablemente, hace perder parte de efectividad al sistema, puesto que esta tarea en diferido significa que las comprobaciones contra la política de circulación no podrán ser realizadas en el momento en el que se ejecuta el préstamo, sino que tendrán que ser realizadas posteriormente, lo que entraña cierto riesgo. Pero, existen razones, en las que no voy a entrar aquí,que podrían obligar a realizar estas operaciones en forma diferida.

Pero no sólo se realizan préstamos y devoluciones para controlar la circulación de los documentos. También sobre los documentos prestados se solicitan renovaciones, que pueden ser de diverso tipo, como veremos después. Cuando los usuarios no cumplen las normas, en lo que se refiere a plazos de devolución, establecidas expresamente por la biblioteca, son sancionados y existen distintos tipos de sanciones. En el mundo anglosajón las sanciones suelen ser económicas, aunque en España las sanciones suelen ser diferentes. Así, por ejemplo, las bibliotecas públicas tienden a utilizar el procedimiento de la desautorización por periodos de tiempo concretos a los usuarios, que normalmente son proporcionales a los días de retraso con los que el usuario devuelve el documento a la biblioteca. Por otra parte, existirá también la posibilidad de hacer reservas de documentos que están prestados o que aun no han terminado de ser procesados por la biblioteca (en trámite de adquisición o catalogación). Las reservas, básicamente, se hacen de dos tipos, reservas a fondos, a copias concretas, o reservas a todo el documento, es decir, reservas a la totalidad de las copias de ese documento. Esta subfunción de reserva plantea toda una casuística del tratamiento de las mismas que puede ser más o menos compleja, en función de la importancia que a esta tarea haya asignado la propia biblioteca. Por último, una tarea que también es necesario realizar

es la de «las cancelaciones», como consecuencia de que se ha agotado todo el trámite de reclamaciones de un préstamo y éste no ha sido devuelto, por lo que puede terminar por ser cancelado como tal préstamo.

Para todo esto, deben existir, como se podrá imaginar, métodos ágiles de entradas de datos. Hoy día se utilizan sistemas de lecturas de códigos de barras, para introducir estos datos rápidamente, pero, como esto es una cuestión que afecta al hardware del ordenador, no voy a entrar especialmente en ella, pues normalmente los problemas de entradas de datos rápidos requieren dispositivos hardware específicos, que no afectan para nada al funcionamiento del sistema software [Evans 83].

Quizá sí merezca especial mención la entidad de política de circulación para tratar de ver sus posibles contenidos y se puede así deducir de qué forma se opera con esta información. En primer lugar, la política de circulación debe poder establecer una tipología de préstamo que recoja las distintas variantes que puedan existir. Me estoy refiriendo a si será posible realizar préstamos o renovaciones telefónicas, o si se va a poder hacer un préstamo normal, de días o en sala y si eso será controlado por el sistema, lo que sería un préstamo de horas, y, por otra parte, qué pasará con los prestamos interbibliotecarios. Por otro lado, además, es necesario establecer un control del calendario de apertura de la biblioteca para que el sistema pueda calcular adecuadamente los plazos de devolución y los períodos de reclamación y cancelación de los prestamos. La política de prestamos que se defina, indudablemente deberá ser decidida por cada biblioteca, pero también será necesario tener en cuenta la posible casuística que implica la existencia de distintas sucursales dentro de la misma biblioteca, que pueden tener políticas de prestamos diferentes.

El objetivo principal de la política de préstamo es que el sistema calcule automáticamente la fecha de devolución de los documentos —si éstos son prestables—, lo que no obsta para que el bibliotecario, en cada operación de préstamo, pueda modificar los parámetros generados automáticamente y sobre éstos introducir él unos propios.

La política de préstamo, además, debe recoger toda la variedad de casos que se puedan dar en una biblioteca. Son tantos como elementos contenga una matriz de dos dimensiones formada por los tipos de materiales para el préstamo existente en la biblioteca y sus tipos de usuarios. Los elementos de esta matriz contendrán cada una de las situaciones de préstamo que se puedan dar en la biblioteca, que afectan, en primer lugar, a los días que cada tipo de documento está disponible para cada tipo de usuario, los distintos períodos de reclamación correspondiente a ese documento para cada tipo de usuario, y lo mismo por lo que afecta a las renovaciones. Es decir, cuántas posibilidades de renovación y por qué períodos existen en la biblioteca para cada tipo de documento y por tipo de usuario. Este conjunto de información, al que se debe añadir, el dato del numero máximo de documentos en préstamo por usuario, de todos los tipos, es el que forma la entidad llamada política de circulación. Con arreglo a esta información, el sistema deberá ser capaz de hacer la desautorización del usuario por procedimientos informáticos como consecuencia del incumplimiento de algunas de las reglas establecidas en esta política de circulación. Pero, como siempre, el sistema debe plegarse a los deseos del usuario y, por ese motivo, debe establecerse un sistema de control de autorizaciones que permita al bibliotecario contravenir las normas que él mismo ha establecido, por razones que se puedan considerar en cada caso.

Siguiendo con las dos entidades primarias, aunque antiguamente los sistemas de control de préstamos utilizaban como información bibliográfica unas referencias mínimas que consideraban suficientes para poder localizar los documentos y reclamarlos en su caso, en la actualidad, en los sistemas integrados, se tiende a utilizar simplemente la información bibliográfica disponible, que son los registros en formato MARC que se encuentran en el catálogo. Pero considero importante poner de manifiesto que, la información bibliográfica, en el caso de control de los préstamos, no tiene otra misión que la de referenciar mínimamente el documento, es decir, no hay necesidad de hacer descripciones muy completas porque una información muy abundante sólo entorpecería el funcionamiento del sistema de control de circulación. En este sentido, se da, incluso, la circunstancia de que, en ocasiones, cuando se pretende poner en funcionamiento rápidamente un sistema de control de circulación y no existe un catálogo, se utiliza un procedimiento de referencia de los ejemplares disponibles en la biblioteca, en base a números de fondos, de registro, y esa es la única referencia que se realiza para hacer los prestamos, hasta tanto no se dispone de una información bibliográfica más completa. El control que se realiza por ese procedimiento es efectivo. Evidentemente, se trata de una situación excepcional, aunque, sin llegar a este extremo, es indudable que la información bibliográfica necesaria para controlar la circulación no tiene por qué ser muy abundante.

En cuanto a la información de usuario, existen dos tipos de informaciones que son importantes. Unas tienen que ver con la identificación del mismo: su nombre, direcciones, teléfonos, etc. Esa información, normalmente es textual, de elementos informativos simples, pero textual. Por otra parte, existen un conjunto de informaciones codificadas que normalmente se consignarán contra tablas de códigos predefinidas, como puedan ser el tipo de usuario, su sexo, su situación académica —en el caso de bibliotecas universitarias—, etc. Estas informaciones codificadas son muy importantes, pues se utilizan con frecuencia para elaborar estadísti-

cas. De este apartado de las estadísticas me ocuparé más en detalle posteriormente.

En cuanto a la seguridad de los datos, es un aspecto que ha sido mencionado repetidamente por los autores que se han ocupado de estos temas. Yo me remito al procedimiento de los perfiles de usuarios con las autorizaciones correspondientes, porque creo que es la única forma de conservar, con una cierta fidelidad, los datos de los usuarios fuera del alcance de quienes no tienen por qué conocerlos y manejarlos.

De todo este conjunto de información entiendo que, por lo que afecta a la información bibliográfica, es necesario poder disponer, en todo momento, de los datos correspondientes al status de préstamo de cada uno de los ejemplares que se encuentran en el depósito de la biblioteca. Esta disponibilidad de la información es especialmente importante en el caso del catálogo de acceso publico en línea, pero también lo es —y casi con mayor motivo— en la función de control de la circulación, puesto que el sistema debe estar en condiciones de dar a los bibliotecarios cumplida información de la disponibilidad de los fondos, no sólo en lo que afecta a si están prestados o no, sino también si están reservados o no, o si se encuentran en situaciones diferentes (en encuadernación, en el expositor, etc.). En definitiva, que se refleje cualquier incidente que afecte a la disponibilidad del documento.

Es importante señalar que existen ciertas funciones que están muy ligadas a la información de la disponibilidad de los documentos, como la función de inventario, que sólo es posible ejercerla si se tienen en cuenta los documentos que el sistema considera en circulación, puesto que los documentos existentes en la biblioteca son la suma de los que se encuentran en el depósito más todos aquellos que, estando en circulación, son controlados por el sistema. Esta operación de sumar ambos tipos de fondos es fácil de realizar por un sistema de estas características, pero solamente es posible si existe un correcto control de los fondos.

En cuanto a la información relativa a los lectores y a sus posibilidades de consulta, ha existido siempre una cierta polémica respecto a la conservación de los datos en las bibliotecas. Durante mucho tiempo se tendía a hacer desaparecer el «histórico» de los préstamos en la biblioteca y a conservar, en todo caso, un pequeño resumen de documentos prestados, por períodos de tiempo concretos, para fines estadísticos. Últimamente, sin embargo, la tendencia es justamente la contraria, tratar de conservar la mayor información posible sobre los préstamos realizados, lo que nos permitiría, incluso, estudiar retrospectivamente, usuario a usuario, sus peticiones de documentos, o, si se quiere, la operación inversa, estudiar documento a documento su historial de circulación. Esta cantidad de información conservada por el sistema es de capital importancia para elaborar una información estadística completa, que ayude al bibliotecario a desarrollar la colección. Lógicamente, ello nos obliga a insistir, una vez más, en la necesidad de proteger toda esta cantidad de información, que en la mayoría de los casos debe ser de carácter confidencial.

Respecto de los productos impresos, deberé decir aquí algo similar a lo que dije en la funciones anteriores, que existen una serie de productos impresos inmediatos, como son cartas de reclamación, notificaciones de cancelaciones, o de desautorizaciones de usuarios concretos, etc., que deben ser emitidos por el sistema de forma ágil, con el fin de repercutir adecuadamente el control informativo en el funcionamiento de la organización. Junto a estos productos impresos o salidas de información impresa, deben existir gran cantidad de salidas impresas que no deben estar sujetas a una definición «a priori», sino que cada biblioteca tendrá que definirlas según sus necesidades, e, incluso considerando que en cada biblioteca podrán variar esas necesidades, según las épocas. Por ello, mi impresión es que debería existir un procedimiento, lo más flexible y adaptable posible, para que cada usuario pudiera definir, en cada caso, la modalidad de producto impreso que necesite, estableciendo unos criterios de selección, en cuanto a las salidas de la información, y también estableciendo unos formatos para la salida de dicha información.

En cuanto a la información estadística, a la que he hecho referencia anteriormente, creo que habría que establecer un procedimiento similar, aunque hay que tener en cuenta que ésta, que afecta a operaciones controladas de circulación, realizadas por usuarios y de documentos concretos, afectará a cualquier información contenida en los registros bibliográficos. Por ello, es necesario que el sistema actúe de la manera más integrada posible, de forma que si alguna información bibliográfica no está disponible para estas funciones estadísticas, será difícil que se puedan realizar determinadas combinaciones de datos. Por ejemplo, es indudable que no es necesaria la notación de CDU contenida en las catalogaciones, para realizar el control de los préstamos. Pero, para funciones estadísticas sí que resulta necesario saber cuáles son los libros prestados por números de la CDU, ya que permitiría a la biblioteca establecer con rigor cuáles son los intereses, por materias amplias de sus lectores y, en base a eso, definir una determinada política de desarrollo de la colección. Por lo tanto, es importante distinguir, en base a esto, la información necesaria para referenciar documentos en las operaciones de control de la circulación y las informaciones necesarias para el tratamiento de datos estadísticos, en relación con el control de la circulación, pues mientras que las primeras pueden ser unas informaciones mínimas, en el segundo caso, las informaciones deben ser lo más completas posible para permitir a los usuarios establecer los criterios de salida de datos que ellos deseen.

Por último, me gustaría terminar con una cita en la que se resume, con bastante acierto, el objeto fundamental de un sistema de control de la circulación:

> "...el sistema debe comprobar que un usuario es apto para recibir materiales en préstamo, confirmar que cada documento presentado puede prestarse y determinar que la combinación categoría de usuario y de documento constituyen un cargo permitido. Si no encuentra alguna de estas condiciones, el sistema debe bloquear el cargo y visualizar el motivo porque lo hace" [Reynolds 89, pág. 569].

III.D. El control de las publicaciones periódicas

La función de control de las publicaciones periódicas se compone de dos subfunciones básicas. La primera es la del control de las recepciones de dichas publicaciones periódicas y, la segunda, la del control de las encuadernaciones. Aunque a primera vista parece que esta función de control de las publicaciones periódicas no es demasiado compleja, sin embargo, numerosos autores han puesto de manifiesto las enormes dificultades que entraña este proceso de control [Osborn 80, Tuttle 83], ya que dichas publicaciones están sujetas, en su evolución, a una serie de incidencias que por procedimientos tanto materiales como automáticos, resultan difíciles de prever. la siguiente cita ilustra con bastante precisión esta situación:

> "...el escepticismo ha sido mucho mayor en cuanto a que la automatización fuese apropiada para el control de las publicaciones seriadas. La causa principal para esta reticencia puede atribuirse, probablemente, a los mismos factores responsables del diseño y uso de sistemas separados cuando se aplica la automatización al control de las publicaciones seriadas, principalmente, la naturaleza y complejidad de las publicaciones mismas. Los ordenadores están mejor equipados para tratar la regularidad y la predecibilidad cosa a la que, como es bien sabido, las publicaciones seriadas se resisten con todas sus fuerzas" [Reynolds 89, págs. 542-543].

La subfunción de control de las recepciones implica el registro de los fascículos de las distintas publicaciones periódicas, a las que está suscrita la biblioteca, debe ser realizado a medida que estos fascículos van llegando, aunque el sistema deberá ser capaz de controlar así mismo el ven-

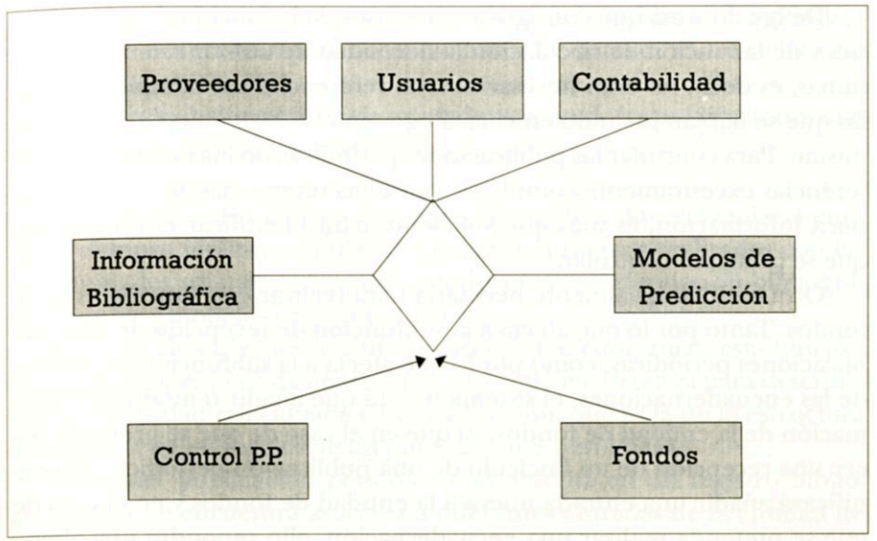


Gráfico 4: *Control de publicaciones periódicas.*

cimiento de cada fascículo y, por tanto, su reclamación, así como los usuarios a los que se debe enviar dicho fascículo y, por supuesto, cuantos y en qué momentos deben ser enviados al encuadernador para agruparlos en volúmenes. Para realizar estas operaciones, se precisan una serie de medios que hasta hace bien poco tiempo siempre eran manuales. Téngase en cuenta que no hace ni diez años, todavía, que algunos autores ponían de manifiesto que era preferible seguir hablando de control manual de las publicaciones periódicas, ya que las dificultades que los analistas de aplicaciones habían encontrado para desarrollar programas que fueran capaces de controlar las publicaciones periódicas, hacían difícil que se generalizara el uso de estas aplicaciones. Sin embargo, hoy día, existen ya una gran cantidad de sistemas, autónomos o que forman parte de SIGB, que se ocupan de este control.

Las informaciones que deben estar presentes en el proceso de control de las publicaciones periódicas son las siguientes: Por una parte, la información bibliográfica que, en este caso, no será la totalidad de la información bibliográfica, como ocurría en los casos de las funciones de adquisiciones y de circulación, puesto que toda la información bibliográfica es «adquirida» y toda la información bibliográfica es «potencialmente circulable». Sin embargo, la función de control de las publicaciones periódicas, como su propia denominación indica, es una función que sólo afecta a un subconjunto de toda la información bibliográfica, concretamente al subconjunto formado por las referencias de las publicaciones periódicas.

De igual forma que consigné en los casos de la función de adquisiciones y de la función de circulación la necesidad de utilizar referencias mínimas, es decir, no eran necesarias unas referencias tan completas como las que se habían incluido en el catálogo, en este caso tengo que decir lo mismo. Para controlar las publicaciones periódicas no hacen falta unas referencias excesivamente complejas, con unas referencias abreviadas, con poca información, es más que suficiente para identificar el documento que se pretende controlar.

Otra entidad igualmente necesaria para realizar esta función es la de fondos. Tanto por lo que afecta a la subfunción de recepción de estas publicaciones periódicas, como por lo que afecta a la subfunción de control de las encuadernaciones, el sistema tendrá que añadir o modificar información de la entidad de fondos, ya que en el caso de que se pretenda hacer una recepción de un fascículo de una publicación periódica, ello significará añadir una entrada nueva a la entidad de fondos y en el caso de que se pretenda realizar una encuadernación, ello supondrá una alteración de alguna de las entradas contenidas en la entidad de fondos. La forma en que se realizan estas modificaciones o actualizaciones de la entidad de fondos, es automática, ya que el sistema de control de las publicaciones periódicas debe ser capaz de predecir en todo momento la llegada de los fascículos y, como consecuencia de ello, actualizar la entidad de fondos automáticamente. Este es su objetivo fundamental. Para hacerlo posible, es necesaria la existencia de una tercera entidad, que denominaremos entidad de los «modelos de predicción», ya que en esta entidad se contendrán un conjunto de informaciones relativas a la evolución de la publicación, en base a los cuales se podrán hacer los distintos subprocesos automáticos enunciados anteriormente. Estas informaciones, en todo momento, deben estar asociadas a informaciones bibliográficas concretas y, mediante los tratamientos adecuados, con la sola identificación de una información bibliográfica y un modelo de predicción asociado a ésta, se podrá realizar el proceso de recepción de dicha publicación periódica.

Por último, es necesario, también, una entidad que contenga un conjunto de informaciones diversas, relacionadas todas ellas con el control de las publicaciones periódicas, denominada entidad de «control de la publicación periódica». Contendrá, al menos cuatro tipos de informaciones diferentes:

— referencias de los fascículos desaparecidos.
— relación de los fascículos recibidos (una especie de registro histórico de la publicación periódica).

— relación de los lectores con interés en cada publicación periódica (ello es necesario para hacer las llamadas «listas de distribución» de cada publicación).
— relación de los pedidos de encuadernación de las publicaciones que se haya decidido encuadernar.

Estas cuatro informaciones podrán estar contenidas en la misma entidad y siempre serán manipuladas a partir de procesos realizados con las tres entidades primarias descritas anteriormente (información bibliográfica, fondos y modelos de predicción).

Antes de describir los distintos procesos que conforman esta función de publicaciones periódicas, es importante que me detenga para describir una peculiaridad relacionada con esta función, que es la de la estructura informativa de los fondos de las publicaciones periódicas.

Cualquier publicación periódica está descrita en un registro bibliográfico que se encuentra asociada a diferentes entradas de la entidad de fondos. Estas entradas tienen, a su vez, relaciones entre sí, puesto que de la misma publicación periódica puede haber, en primer lugar, una serie de volúmenes, formados a su vez por distintos fascículos, de los que puede haber también distintas copias. Esto determina una estructura de datos piramidal, que debe reflejarse en la entidad, con sus correspondientes relaciones. Además, esta estructura no es estática, sino que puede variar, es decir, que debe ser definida con carácter dinámico. Así, por ejemplo, si un conjunto de fascículos se encuaderna formando un volumen, uno de los niveles de la pirámide habrá desaparecido para ser asumido por el nivel inmediatamente superior. Los ejemplares o copias de los fascículos son siempre unidades documentales físicas diferentes, mientras que el resto de los elementos que forman la estructura piramidal, no son más que «nodos» de un modelo de datos que utilizamos para gestionar la información.

En este sentido resulta difícil, a veces, trasladar la realidad a esta estructura, teniendo en cuenta que muchas publicaciones periódicas no respetan este esquema de fascículos que se agrupan formando volúmenes. A pesar de ello, si cambiamos la denominación y convertimos los volúmenes en grupos de fascículos conseguiremos que esas agrupaciones se puedan realizar a cualquier nivel de la estructura piramidal y, en ese caso, será más fácil trasladar cualquier realidad, por atípica que resulte a nuestra estructura de datos.

Esta casuística que acabo de describir, refleja lo que decía al principio de la descripción de esta función, y es que el control de las publicaciones periódicas, por la falta de normalización de las mismas, es una tarea que resulta verdaderamente complicada en ocasiones. Pero, a pesar de las dificultades, la clave del control de las publicaciones periódicas se encuen-

tra, precisamente, en lo que algunos autores han llamado «modelos de predicción». Estos modelos de predicción, si están bien definidos, permitirán el control automático de las publicaciones periódicas y, si no lo están, supondrán un inconveniente añadido a los existentes. Por ello, la tarea primera a realizar para la puesta en marcha de un sistema de control de las publicaciones periódicas, consiste en definir unos buenos modelos de predicción. Este contendrá un resumen dividido en tres niveles de los datos que conocemos a priori sobre el funcionamiento de la publicación periódica. Pero, de entrada, tengo que decir que previendo las dificultades con las que nos enfrentaremos al intentar controlar con los modelos de predicción las publicaciones periódicas, por bien que éstos hayan sido definidos, será necesario permitir la posibilidad en el uso del sistema de que una misma publicación periódica posea varios modelos de predicción, que pudieran ser elegidos alternativamente, según la casuística que traiga aparejada cada suceso en la historia de la publicación.

Las informaciones que conforman el modelo de predicción son las siguientes: En un primer nivel, se encuentran todos los datos relativos a la frecuencia de esa publicación, organizados de la siguiente manera, en primer lugar, lo que podríamos llamar la frecuencia básica de la publicación, es decir, la frecuencia de los fascículos normales. Esta frecuencia debe ser definida en el modelo estableciendo simplemente una ratio fascículo/tiempo. Además, será necesario un modelo de frecuencia que refleje un ciclo completo de tiempo con las incidencias previstas en ese ciclo. Ello es necesario porque las publicaciones periódicas, independientemente de la frecuencia básica que tengan, están sujetas a una serie de variaciones. Por ejemplo, una publicación mensual puede editar un fascículo combinado en los meses de verano, de tal forma que, la frecuencia mensual, se interrumpe en esos meses, para convertirse en una frecuencia trimestral. Esto se refleja en el modelo de predicción, utilizando una plantilla de ciclo completo que permita formalizar este tipo de incidencias [Grosh 76].

A continuación será necesario consignar, también, los datos relativos al incremento de volúmenes y fascículos o, lo que es lo mismo, los datos relativos a la enumeración de volúmenes y fascículos, puesto que éste es el procedimiento más corriente de secuenciación de las publicaciones periódicas.

Otra información importante es la relativa a las reclamaciones, para lo que es necesario, en primer lugar, señalar un período de tiempo que se considera razonable como tiempo de demora en la recepción de cada fascículo. Este período deberá variar en función de la frecuencia de la publicación misma, aunque también es necesario que el sistema sea capaz de ajustar estos períodos de demora en función de la propia historia de la publicación. Ajustarla, se entiende, automáticamente, de manera que, aunque inicialmente se ajuste manualmente un período, pueda luego el sistema variar este período de demora, en función de los retrasos que se vayan produciendo en la recepción de los fascículos. En relación con este dato, deberán establecerse unos ciclos de reclamación. Probablemente, la forma más lógica de hacerlo es definir unos tipos de reclamaciones estándar y que cada modelo de predicción utilice uno de estos tipos. A propósito de los tipos de reclamación, luego comentaré qué informaciones puede contener cada uno.

Otros datos a tener en cuenta son los relativos a los ejemplares de índice de cada publicación periódica, debiendo especificar la frecuencia de estos índices y su demora. Es decir, una información similar a la que se ha introducido para los volúmenes normales. También es cierto que las publicaciones periódicas no sólo editan números de índice, sino también editan números extraordinarios o monográficos, aunque los números extraordinarios reciben un tratamiento diferente.

Con el fin de poder establecer lo que serán las informaciones relacionadas con la predicción de encuadernaciones es necesario consignar, también, como parte del modelo de predicción, los datos relativos a volúmenes y fascículos, de tal manera que el sistema pueda, en función de esto, determinar cuándo se han recibido los fascículos correspondientes a un volumen y, por tanto, deben ser enviados al encuadernador. Esta relación fascículo/volumen, en realidad no es más que una «ratio» que se puede establecer de varias formas: Por períodos de tiempo, por números fijos de fascículos y, en cualquier caso, con independencia de la predicción que automáticamente realice el sistema, manualmente se puede determinar cuándo se han recibido fascículos suficientes para conformar un volumen.

Al describir la estructura de datos piramidal que refleja los fondos de una publicación periódica, dije anteriormente que las entidades físicas que tiene la biblioteca no son los fascículos, sino los ejemplares que se tienen de cada fascículo. Esto quiere decir, que una biblioteca puede disponer de tantos ejemplares físicos de un fascículo como suscripciones haya hecho de una revista. El modelo de predicción debe reflejar también esta circunstancia, asociando a cada conjunto de fascículos existentes de una publicación periódica, una serie de informaciones, como son, en primer lugar, la localización, es decir, el código de la sucursal donde se encontrará dicha suscripción y las informaciones de signatura topográfica que le correspondan, el tipo de material para el préstamo, tanto para el caso de que sea o no sea prestable y la lista de distribución asociada a dicha suscripción, junto con los datos relativos a la encuadernación que a continuación reseño.

Los datos de encuadernación recogen la referencia del encuadernador al que será enviada dicha publicación periódica para ser encuader-

nada, las notas de aviso para el encuadernador, de tal forma que éste sepa si tiene que realizar un tratamiento específico de la encuadernación, con independencia del tipo de encuadernación, que también vendrá consignado y, por supuesto, la información relativa al conjunto de fascículos que deben formar cada uno de los volúmenes, junto con una información abreviada de la propia publicación periódica.

Estos datos, forman el llamado modelo de predicción y, en base a ellos, el sistema debe ser capaz de permitir la visualización de las existencias de una publicación periódica, reconstruyendo, en orden inverso, la historia de esa publicación, pues no sólo se nos podrán mostrar los fondos que existen en ese momento, una vez realizadas las encuadernaciones, sino también un histórico de la recepción de los diferentes fascículos. Pero, como ya dije anteriormente, el sistema debe ser capaz de controlar todas las incidencias por las que atraviese la publicación. Quizá la más frecuente sea la que afecta a los fascículos perdidos, que deben ser recogidos en la entidad de información de control de la publicación periódica. Este registro de fascículos perdidos contiene una información que ha salido del patrón de predicción —la información relativa a la información del fascículo—, así como una información adicional sobre el fascículo que falta que puede ser añadida por el usuario. Realmente existen dos procedimientos para generar una entrada en el registro de fascículos desaparecidos. Un procedimiento es manual: El usuario, en el proceso de recepción, puede descubrir que el fascículo que el sistema espera recibir no es el que ha llegado sino que es uno posterior, de manera que el anterior hay que registrarlo como desaparecido. Pero, también, existe un procedimiento automático que consiste en que una vez superados los tiempos consignados en el modelo de predicción, el sistema, automáticamente, generará una carta de reclamación y dará de alta una entrada en el registro de fascículos perdidos.

También hablé antes de las llamadas «listas de distribución», que son, normalmente, entradas extraídas de la entidad de usuarios del sistema que se asocian, mediante los modelos de predicción, a informaciones bibliográficas de publicaciones periódicas, con el fin de poder confeccionar listas de usuarios, a los que se notificará o bien la llegada de un fascículo de la publicación periódica por el que han mostrado interés, o bien el propio fascículo cada vez que llegue, según las normas de funcionamiento de la biblioteca.

Por lo que afecta a las normas de reclamación, es preciso tener en cuenta que estas normas pueden suponer más problemas que beneficios para la propia biblioteca. La cantidad de retrasos que se producen en la recepción de las publicaciones periódicas, especialmente cuando son extranjeras, hacen que el control de reclamaciones, cuando es muy estricto,

se convierta en un impedimento para el buen funcionamiento del sistema, más que una ventaja de control informativo. Por ello, es necesario inicialmente, no ser demasiado exigentes en la política de reclamaciones. Las reclamaciones se deben efectuar tanto de los fascículos que no han sido recibidos, como de las encuadernaciones que no han sido devueltas. Se emitirá una carta de reclamación en función de la política de reclamación establecida. Esta se basará esencialmente en cuatro datos: Los días de retraso en los que se emitirá la primera, segunda y tercera carta de reclamación y la comunicación de cancelación de pedido. El momento a partir del cual se considera que un fascículo debe ser reclamado es, por tanto, clave en el proceso de reclamación. Teniendo en cuenta que el sistema conoce la fecha de recepción esperada, los días de demora aceptados, que es un dato que forma parte del modelo de predicción, y el número de días que establecen las propias normas de reclamación, el sistema tendrá, entonces, que sumar estas tres cantidades y, si el día en curso es posterior a la fecha resultante de esta operación, se iniciará el trámite de la reclamación, imprimiendo el correspondiente aviso.

En cuanto al proceso de control de las encuadernaciones, una vez definido en el modelo de predicción los datos relativos a las encuadernaciones, el sistema debe ser capaz de reconocer aquellos conjuntos de fascículos que han llegado y están listos para ser enviados a la encuadernación. Ahora bien, lo que el modelo que predicción no le permite conocer al sistema son las informaciones relativas al trámite presupuestario de la encuadernación, con lo cual, en el momento de enviar a encuadernar un conjunto de fascículos, se debe elegir, no sólo el proveedor correspondiente, sino también la partida presupuestaria de la entidad de contabilidad descrita en la función de adquisiciones a la que se cargará dicha operación de encuadernaciones. En ese momento, el sistema ya tiene información suficiente sobre los cambios que deberá realizar en la entidad de fondos, ya que los fascículos individuales han sido consignados como fondos independientes y, con motivo de la encuadernación, deberán desaparecer como fondos independientes y aparecer como un solo fondo, el nuevo volumen. Esta operación se realiza a la recepción de los volúmenes encuadernados. Si se produce alguna alteración de estas informaciones sobre lo previsto, manualmente, en el momento de la recepción se pueden cambiar los datos.

Evidentemente, todas estas operaciones suponen un menor esfuerzo por parte de los usuarios del sistema, tanto más, cuanto que las publicaciones periódicas se ajustan, en su funcionamiento, a una secuencia preestablecida de acontecimientos, lo que no siempre se puede garantizar. En cualquier caso, esta función de control de las publicaciones periódicas, no debe olvidarse, tiene por objeto mejorar el servicio al usuario, en un triple

sentido: a) Dando una información lo más precisa posible de las existencias de publicaciones periódicas de que dispone la biblioteca. b) Controlando mejor y garantizando, por tanto, el desarrollo de la colección de publicaciones periódicas. c) Controlando mejor la conservación y, por consiguiente, conservando mejor, la colección de publicaciones periódicas. Como se puede observar, para lograr estos objetivos, las operaciones implicadas, a veces, resultan un tanto complejas, pero son imprescindibles si se quiere garantizar el cumplimiento de estos objetivos.

III.E. La función de referencia

Las operaciones que se realizan en las bibliotecas, relacionadas con las funciones de referencia, son todas aquellas que tienen por objeto facilitar a los usuarios de la biblioteca el acceso a la información que la biblioteca controla. Evidentemente, este objetivo es muy genérico y necesita de una mayor precisión, ya que a lo largo del análisis de las funciones básicas estamos viendo que se realizan distintas operaciones de consulta, unas veces en relación con procesos de catalogación, otros con procesos de adquisición e incluso con procesos de circulación, etc. Y estas operaciones de consulta, que están ligadas a las funciones básicas ya descritas, no tienen específicamente que ver con las operaciones de consulta que pretenden facilitar el acceso de los usuarios a una determinada documentación. De este tipo de consulta es del que voy a hablar aquí, advirtiendo de antemano que mi trabajo se ha centrado, muy especialmente, en todos los aspectos relacionados con las estructuras de datos y los tratamientos necesarios para mejorar los sistemas tradicionales de recuperación de información que se utilizan en los servicios de referencia de las bibliotecas.

Este tipo de consultas son realizadas tanto por usuarios profesionales —bibliotecarios—, como por usuarios finales de la biblioteca y se ejecutan tanto en relación con la información local, es decir, con la información que produce y administra la propia biblioteca, como con relación a información externa, no contenida en los catálogos de la propia biblioteca. En el caso de la información externa, no es una práctica única que los usuarios finales realicen este tipo de consultas, sino que, a veces, a instancias de estos usuarios, los usuarios profesionales —el personal técnico de las bibliotecas— realiza este tipo de consulta. Las operaciones relacionadas con cualquier tipo de consulta, realizada por cualquier tipo de usuario, estarán muy condicionadas por la forma en que el sistema ha producido la información, así como por el tipo de instrumentos de acceso a la información que el propio sistema haya generado. Por ello, esta función la describo al final, cuando se supone que todas las operaciones relativas

a la actualización de la información en la base de datos en la biblioteca, ya se han realizado y han sido descritas de acuerdo con las necesidades que, de cara al servicio de referencia de la biblioteca, plantean los usuarios.

Es importante recordar que la mayor parte de las consultas relacionadas con el acceso a la documentación, serán consultas de la información bibliográfica, aunque, como veremos también, existen informaciones muy puntuales, que no son bibliográficas, que también son demandadas por los usuarios en las operaciones de acceso a la documentación.

Para empezar y como marco general, en mi opinión, es de vital importancia que el sistema no plantee ningún tipo de restricción para la creación y mantenimiento de bases de datos diversas con distinto tipo de información, que puedan ser accedidas simultáneamente, si bien es verdad que la base de datos, propiamente catalográfica deberá ser única, sin embargo, existen otros tipos de informaciones, que no son los puramente catalográficos, que pueden ser demandados por los usuarios y que el sistema debe estar en condiciones de mantener. Para ello, el sistema deberá permitir la definición de registros con diferentes formatos, sin ningún tipo de limitación. Evidentemente, en lo que se refiere al catálogo, como ya se dijo, la base de datos que habrá que definir, será una base de datos capaz de mantener toda la información preceptiva en la norma MARC en sus diferentes versiones compatibles entre sí, aunque no exactamente iguales. Esta compatibilidad se cifra a nivel de campo, aunque no se establezca a nivel de subcampo.

Para hacer posible este conjunto de especificaciones, respecto de la base de datos, es necesario utilizar un sistema de gestión de la misma, que permita registros de longitud variable y campos de longitud variable. También es importante recordar que el usuario debe tener la capacidad de decidir cuáles serán las informaciones indexadas de esa base de datos. Evidentemente, en el caso de la información bibliográfica, esto incluirá todos los puntos de acceso definidos por la normativa internacional, más aquellos campos que de manera local se considere necesario que sean recuperables, esto incluirá también cierta información en lenguaje natural (abstracts, índices, sumarios, etc.).

En relación con este tipo de bases de datos, las posibilidades de recuperación deben ser establecidas en consonancia con las necesidades que los usuarios puedan plantear. Por ese motivo, en primer lugar, es importante establecer para la información bibliográfica cuáles son los puntos de acceso. Tradicionalmente estos deben ser: las entradas de autor o autores, de entidad, congreso, materias, series, etc., pero a estas entradas que tradicionalmente se consideran de punto de acceso hay que añadir, como dije anteriormente, las que localmente el usuario defina como tales.

La forma de recuperación, en todo caso, siempre deberá ser doble, a través de la entrada completa, tal y como ha sido introducida en la base de datos o bien a través de palabras claves. Esto, tanto para el caso de los campos definidos como autoridades, como para los demás casos. En este sentido la información relativa a la frecuencia de aparición de las diferentes entradas es vital para la ponderación de las mismas, lo que permitirá la utilización de técnicas de recuperación basadas en modelos matemáticos que superen los sistemas tradicionales (Boolean Matching).

Ni que decir tiene que será necesario permitir la realización de todo tipo de combinaciones mediante operadores entre los distintos campos definidos como recuperables. Asimismo se permitirán los truncamientos y los operadores disponibles serán operadores lógicos y de proximidad con los anidamientos que el usuario decida. Por otra parte, las técnicas basadas en la retroalimentación mediante la relevancia o los métodos probabilísticos también deben ser posibles.

Por último, respecto del procedimiento de indización, dos notas más: Por un lado, la indización de palabras claves se efectuará contra listas de «stopwords» definidas por los usuarios y, al mismo tiempo, habrá ciertos campos que normalmente contendrán información codificada que podrán ser utilizados como limitadores en las búsquedas. Estos campos, definidos como limitadores, desempeñan un papel muy importante; sobre todo con vistas a la consecución de conjuntos de información muchos más ajustados a las necesidades de los usuarios.

En definitiva todas estas posibilidades de búsqueda, son consecuencia de una estructura de acceso basada en la existencia de índices de palabras claves de diversa naturaleza y con gran cantidad de información de control asociada.

Para salvaguardar la posibilidad de utilización de terminologías alternativas, será necesario que en las búsquedas esté contemplada la posibilidad de poder recuperar a través de las referencias cruzadas, definidas en el proceso de entrada de los registros bibliográficos. Asimismo, con vistas a poder reflejar cierto tipo de relaciones complejas entre las referencias, como por ejemplo, las que se producen entre las obras multivolumen, que tienen autores y títulos diferentes, será necesario que el sistema sea capaz de realizar un control de entradas relacionadas de registros, que pueda ser, a su vez, utilizado en el desarrollo de los programas de consulta, para facilitar al usuario la recuperación de esos registros de forma relacionada. Al mismo tiempo, deben existir cierto tipo de ayudas en la recuperación, como por ejemplo, la posibilidad de reconstruir la historia de las búsquedas anteriores, salvar los perfiles de búsquedas realizadas con vistas a su uso en próximas sesiones de búsqueda, combinar búsquedas anteriores, por medio de todo tipo de operadores. En definitiva, todas aquellas funciones que permiten la realización de tareas de difusión selectiva de la información.

Teniendo en cuenta que una de las operaciones de referencia fundamental es el llamado catálogo de acceso público, habría que dedicar especial atención a cuestiones tales como la amigabilidad del sistema, ya que existen grandes diferencias entre desarrollar aplicaciones para el acceso a la información para profesionales de la información y hacer esos mismos desarrollos para usuarios finales, más teniendo en cuenta que muchos de estos usuarios finales son usuarios eventuales del sistema. Por tanto, el concepto de amigabilidad tiene que ser utilizado como punto de referencia obligado en el desarrollo del OPAC del sistema [Jones 88].

No voy a extenderme demasiado en los pormenores del desarrollo de un interface de usuario amigable para un OPAC, simplemente voy a enunciar algunas características generales que deberán ser tenidas en cuenta en el momento del desarrollo del interface. Estas características son fácilmente deducibles, a partir del análisis más elemental de las necesidades de los usuarios en relación con los OPAC.

En primer lugar, habrá que hacer una oferta diversa de interface, según que los usuarios sean profesionales o finales. En el primer caso, se podrá utilizar un interface de grandes prestaciones, aunque más complejo en su uso —como es el interface de comandos—, mientras que en el segundo caso, quizá sea más aconsejable la utilización de un interface de menús de menos prestaciones, pero mucho más fácil de utilizar. Dentro de estos interfaces hay que tener en cuenta que debe haber también niveles de uso, que puedan ser elegidos, en función de la experiencia que cada usuario tenga en el manejo de dicho interface.

Todo tipo de informaciones de ayudas y mensajes de error en el curso de la utilización del interface, deben estar disponibles, con el fin de facilitar la resolución de posibles incidencias de manera rápida.

Quizá una de las mejores soluciones existentes hoy día y que se comentará más adelante, en el último capítulo del trabajo, sea la utilización de herramientas para el desarrollo de interfaces gráficos, basados, fundamentalmente, en los llamados sistemas WIMP (Windows, Icons, Menus y Pointers) y entre las distintas opciones que a este nivel se plantean la más interesante, en mi opinión, es la que se basa en la filosofía cliente/servidor, que tiene por objeto facilitar el acceso mediante entornos operativos gráficos desde estaciones de trabajo independientes a una base de datos común, lo que facilita en gran medida el desarrollo de aplicaciones basadas en esos entornos gráficos. De cualquier forma, sólo me interesa ahora dejar constancia clara de la necesidad de desarrollar herramientas amigables, especialmente por lo que se refiere a la aplicación del catálogo de acceso público. En relación con esto, el tema de formatos de visualización,

muy discutido por los expertos en la materia, necesita también de un somero análisis.

La práctica más extendida, una vez realizada la búsqueda, se realiza siempre en dos fases: una llamada de formato abreviado y otra, de formato completo. Pues bien, en lo que afecta al formato completo, deberán estar disponibles diferentes versiones de este formato completo, algunas de ellas podrán ser definidas por el propio usuario que, en cualquier caso, en el momento de la recuperación podrá elegir entre las diferentes versiones disponibles la que más le convenga, y, sobre todo, es importante que el sistema permita visualizar de forma selectiva la información localizada y recorrer esta información en una dirección u otra, mediante procedimientos de navegación. Todo esto esto sin olvidar que los dos modos clásicos de búsqueda de la información en OPAC deben estar permitidos y claramente diferenciados: «Item Search» y «Browsing». A este respecto, llegado el momento me centraré en la relación que debe existir entre estos modos de búsqueda y algunas de las técnicas de recuperación de información avanzadas mencionadas anteriormente.

Por último, me gustaría hacer algunas observaciones sobre la relación con informaciones no bibliográficas que deben tener la función de consulta y referencia. Existen informaciones, tanto de adquisiciones, como de circulación, como de control de publicaciones periódicas, que deben estar disponibles para los usuarios junto con la información bibliográfica en todo momento. La información de adquisiciones disponibles se refiere, fundamentalmente, al status de los documentos que están en proceso de adquisición. En lo que se refiere a circulación, el status de préstamo y la información de reservas. Y por lo que afecta al control de publicaciones periódicas, es importante reseñar que si la información disponible es únicamente una información de fondos, ésta puede resultar incompleta para los usuarios de la biblioteca, pues la información de fondos no recoge las incidencias de números desaparecidos y, por tanto, se deberá tener acceso directamente a la entidad de control de publicaciones periódicas, además de la de fondos, para poder suplementar la información de fondos con las informaciones relativas a estas incidencias, en el control de la publicaciones periódicas. Todo este proceso de acceso a informaciones en zonas distintas del catálogo, evidentemente, debe ser transparente para el usuario que, en todo caso, tendrá una visión uniforme del conjunto de la información almacenada en dicha base de datos.

Aunque es un problema que técnicamente plantea serias dificultades, desde un punto de vista funcional, me gustaría hacer algunas observaciones del acceso a la información externa, que yo plantearía a dos niveles. De una parte, las bibliotecas tienen necesidad de acceder a la información contenida especialmente en las bases de datos bibliográficas de las restantes bibliotecas, para fines diversos, satisfacer las necesidades de información de sus usuarios, por lo que afecta a la disponibilidad de documentos en otros centros bibliotecarios, la necesidad de realizar operaciones de préstamo interbibliotecario, etc. Pero, por otro lado, existe también un activísimo mercado de información, fundado en las empresas que se dedican a dar servicios de índices y resúmenes, es decir, lo que se llama el mercado de información «on line». Las bibliotecas desempeñan un papel importante como centros que proporcionan a sus usuarios una información que obtienen a su vez de la que suministran estas empresas. Para lo cual, el sistema de información automatizado de la biblioteca debe estar en disposición de conectarse con los recursos de distribución de estas empresas con el fin de obtener la información que el usuario demande.

Aunque, a diferente nivel, tanto un problema como otro plantean una serie de dificultades técnicas que es necesario resolver. La solución puede establecerse a muy diferentes niveles, aunque desde que las organizaciones internacionales de normalización han empezado a intervenir en el tema de los procedimientos de interconexión de los sistemas informáticos y, más específicamente, de los procedimientos de interconexión de los sistemas informáticos para realizar operaciones relacionadas con el mundo bibliotecario. Es factible, y además es necesario, que los sistemas de información automatizados de las bibliotecas contemplen estas funciones que, en definitiva, son nuevas.

En este sentido, el conjunto de las normas llamadas OSI y sus correspondientes aplicaciones en el campo de las bibliotecas, marcan unas pautas que deben ser seguidas por los analistas de sistemas bibliotecarios en el desarrollo de las nuevas aplicaciones, pues facilitarán, no sólo el intercambio de información, sino también la conexión inteligente entre sistemas para dar un mejor servicio a sus usuarios [Iso 10162, 10163]. En este sentido, un buen punto de referencia al comentar el desarrollo de una aplicación es la documentación perteneciente al proyecto SIBI [Sibi 89], que tiene por objeto, precisamente, la interconexión de sistemas bibliotecarios en el territorio nacional, así como documentación relativa a otras experiencias de interconexión de sistemas en otros países del mundo [Martin 86]. De cualquier forma, éstas son algunas de las necesidades que tienen que ver con la interconexión de sistemas y para la satisfacción de estas necesidades, considero que algunas de las observaciones que voy a realizar pueden resultar útiles.

IV

EL MODELO RELACIONAL

Para empezar la descripción de este modelo comenzaré haciendo una breve introducción histórica, con el fin de situar la teoría relacional en su contexto adecuado.

La primera definición que se propuso de una gestión relacional de las bases de datos fue hecha en el año 70 por el doctor Codd, que en aquel momento trabajaba para la empresa IBM en sus laboratorios de investigación de California [Codd 70]. Las ideas propuestas por Codd fueron rápidamente difundidas en todo el mundo académico y, como consecuencia de ello, comenzó el desarrollo de ciertos prototipos, que, de manera absolutamente experimental, pretendían la implementación de las características definidas por Codd, para conseguir lo que debería ser un sistema relacional. Estos proyectos de investigación se empiezan a desarrollar en las universidades, donde sus propuestas tuvieron más aceptación.

Entre estos prototipos, uno de los primeros que se desarrolla es el que realiza el propio Codd en los laboratorios de IBM, que recibe el nombre de Sistema R y cuyo desarrollo tuvo lugar durante los años 70. De forma simultánea se inicia el diseño de otros prototipos, como por ejemplo el llamado Ingres, cuyo desarrollo se inicia también en esas mismas fechas en la universidad de Berckeley [Date 87]. Por supuesto estos no fueron los únicos, pero sí probablemente los que desde el principio se plantearon como proyectos más ambiciosos.

En definitiva, desde el primer momento podemos establecer una distinción clara entre aquellos proyectos que pretenden implementar en los productos resultantes todas las características que la definición teórica del modelo relacional preveía y, por otro lado, aquellos productos que probablemente se desarrollan con fines claramente comerciales y que en modo alguno pretenden implementar todas las características del modelo relacional, sino sólo aquéllas que permitan al producto resultante colocarle la etiqueta de relacional.

Desde un punto de vista estrictamente técnico, esto tiene repercusiones importantes, en las que no voy a entrar ahora, ya que, en este punto, lo único que me importa resaltar es que son justamente estos proyectos,

que desde el principio fueron más ambiciosos, los que se han convertido en el motor de desarrollo del propio modelo relacional, que no es un modelo estático, sino que, desde sus orígenes, ha evolucionado tratando de dar solución a los problemas de gestión de la información que se le iban planteando. De tal forma que el desarrollo de sistemas que siguen el modelo y su uso posterior han servido de feedback para la propia teoría relacional. La mayor parte de los productos comerciales que han surgido después en la empresa IBM son el resultado de las experiencias del Sistema R, lo mismo por lo que se refiere al proyecto Ingres. Cuando este proyecto estaba maduro se creó una empresa que comercializó un producto con este mismo nombre que todavía existe en el mercado. Con esto quiero resaltar que el modelo relacional siempre ha tenido una capacidad muy importante para dar lugar a productos que, en mayor o menor medida, reproducían en la práctica sus elementos esenciales, lo que se ha traducido a la vez en la generalización del uso de aplicaciones desarrolladas a partir de software básico más o menos ajustado al modelo relacional.

Desde finales de los años setenta hasta principio de los noventa a Codd se han unido una serie de autores —quizá uno de los más representativos sea Date, aunque no es el único— que han contribuido a mejorar la definición y a ampliar el conjunto de características del modelo relacional inicialmente definido [Date 90]. Esto ha supuesto la aparición de nuevas versiones de dicho modelo. La última, llamada RM2 (modelo relacional versión 2), contiene una más amplia gama de características además de las inicialmente definidas, hasta tal punto que el propio Codd nos dice que en esta versión dos, respecto de la uno, existen algunas diferencias, especialmente por lo que afecta a ciertos aspectos que sólo estaban apuntados en la versión uno y a funcionalidades nuevas que ni siquiera aparecían en la versión inicial [Codd 90]. Esta versión dos persigue, básicamente, tres objetivos fundamentales:

- En primer lugar, simplificar la interacción con los datos por parte del usuario, puesto que a esta cuestión se había dedicado poca atención, pues lo que se pretendía era definir un esquema conceptual de tratamiento de los datos sin prestar demasiada atención a los problemas de desarrollo de las aplicaciones, o a los de interfaciación con el usuario.
- En segundo lugar, se ha tratado de dotar al modelo de características que mejoren la productividad de los usuarios desarrolladores, de tal forma, que se hace más hincapié en las cuestiones relativas al lenguaje relacional.

- Por último, se dota al modelo de las herramientas necesarias para la administración de la base de datos, es decir, todo lo relacionado con los problemas de seguridad, acceso compartido y control de la integridad.

Junto a las especificaciones de esta nueva versión, Codd incluye algunas de las que han venido apareciendo como necesarias en los documentos de otros estudiosos, sobre todo, a lo largo de los años ochenta. Esto significa que el modelo ya no es simplemente la teoría de un investigador, sino que se ha convertido en una referencia efectiva en el mundo informático y también en un estándar, en cierta manera, como veremos más adelante.

Una vez visto a grandes rasgos el breve desarrollo histórico del modelo relacional desde sus orígenes, comenzaré con la descripción del modelo, diciendo antes que aunque el origen es matemático, en su desarrollo el modelo se aparta de este origen.

Existen ciertos conceptos iniciales que deben ser explicados para poder comenzar con la descripción del modelo completo. Empezaré diciendo que el modelo relacional, en realidad, no es sino una forma de tratamiento de datos, entendiendo estos datos como elementos informativos. Para hacer posible ese tratamiento son necesarias un serie de prescripciones. Este conjunto de prescripciones, que configuran una forma de procesar datos, forman a su vez un modelo denominado modelo relacional. En este sentido, existen tres aspectos diferentes de estas prescripciones que deben ser tenidos en cuenta:

— Las prescripciones de datos, es decir, el conjunto de normas que determinarán cuál es la estructura que deben tener los datos que serán manejados de acuerdo con este modelo.
— En segundo lugar, las prescripciones de integridad de esos datos.
— En tercer lugar, las prescripciones de manipulación.

En definitiva, aunque el modelo persigue la definición de unos criterios respecto de la manipulación de los datos, sin embargo tiene como prerrequisitos un conjunto de especificaciones, muy detalladas, respecto de la estructura e integridad de esos mismos datos, porque si esto no fuera así, las prescripciones de manipulación no serían aplicables.

IV.A. Prescripciones de la estructura de datos

Al tratar la estructura de datos relacional tendré que introducir algunos conceptos relativos a la forma concreta en que las informaciones se organizan para ser tratadas por los sistemas relacionales. Por ello, es necesario advertir de antemano que, aunque el modelo relacional en boca de algunos autores parece tener una aplicación de carácter universal por lo que al tratamiento de la información se refiere, sin embargo, el hecho de que sea necesario definir unas estructuras de datos con las que el modelo trabaja, significa que existen otras estructuras de datos con las que el modelo no trabaja. Por tanto, forma parte de la definición del modelo sus limitaciones en cuanto al tipo de información que es capaz de manejar.

Para empezar con la descripción de la estructura de datos relacional, diré que los datos de éste se agrupan formando relaciones, que no son sino estructuras tabulares que permiten albergar informaciones de distinta índole. Cada una de las filas de estas tablas en las que se almacena información recibe el nombre de tupla, mientras que las columnas se denominan atributos. El número de atributos que tiene una relación se denomina grado de la relación, y el número de tuplas que la compone se denomina cardinalidad. Por otra parte, cada una de las relaciones contiene una clave primaria, que es la única forma que existe de identificar cada una de las tuplas de una tabla, ya que se entiende que la claves primarias nunca tienen valores repetidos y, por tanto, se impide la confusión entre tuplas en el momento del acceso. Finalmente, quizá uno de los conceptos más importantes de la estructura de datos relacional es el concepto de dominio, que sirve para agrupar un conjunto de valores que pueden pertenecer a uno o más atributos de una relación y, como veremos después, desempeña un papel primordial a la hora de realizar ciertas manipulaciones de la información.

El punto de partida para la definición de la estructura de datos relacional es la unidad de información más pequeña en un sistema relacional, lo que se ha llamado hasta ahora datos individuales. Por ejemplo, el nombre de un editor en una descripción bibliográfica puede ser considerado un dato individual, o el nombre de un usuario. Los expertos en el modelo relacional dicen que los valores de los datos son escalares, lo que quiere decir que representan la unidad de información semántica mínima, y con más frecuencia se suelen referir a estos datos denominándolos como informaciones atómicas, o, lo que es lo mismo, que no tienen estructura interna y no se pueden descomponer, por lo menos no en el sentido en el que el sistema lo entiende. Es decir, un dato, al no tener estructura interna, será parte de una estructura más amplia, cuyo control permitirá el acceso y la manipulación individual de las informaciones más pequeñas, sin necesidad de su descomposición posterior. Este concepto de la atomicidad de la información debe ser desarrollado con cierta precaución, pues se podría dar la impresión de que los distintos valores de un atributo en una relación no pueden descomponerse de ninguna forma en partes más pequeñas, y esto no es del todo exacto. Aunque se asume normalmente que los valores de un dominio dado son atómicos y, por consiguiente, esta característica es inherente a los datos manejados por el propio modelo, quizá sería interesante evaluar cuáles son las consecuencias de esta asunción.

En primer lugar, esta asunción está ligada al concepto de normalización de la información en el sistema relacional, o, lo que es lo mismo, tal y como lo definió Codd en los años setenta, que la información deberá estar en primera forma normal. Esto significa que cualquier información, dentro de la estructura relacional, debe cumplir la propiedad de que toda intersección fila/columna de una relación (tabla) es siempre y exactamente un solo dato, nunca un conjunto de valores informativos [Date 90b].

Esta definición de lo que es una relación normalizada tiene una consecuencia práctica directa, y es que cualquier conjunto de información dado que pretenda ser gestionado por un sistema relacional, en primer lugar deberá ser trasladado a primera forma normal. Es decir, descompuesto de tal forma que sea asequible a una serie de estructuras tabulares.

Este es el principio de la atomicidad de las informaciones en el modelo relacional. Pero, indudablemente, algunas ventajas debe tener la aplicación de este principio. Existen, efectivamente una serie de ventajas de tipo práctico que han sido relacionadas por distintos autores. Por ejemplo:

- La normalización de las relaciones permite la visualización de la información de forma tabular, lo que la hace más asequible y comprensible para el usuario.
- Por otra parte, la normalización permite un mecanismo de direccionamiento de la información, por medios automáticos, que es relativamente fácil de usar y, sobre todo, muy eficaz, pues la organización interna de las memorias en los sistemas informáticos trabaja con facilidad con ese tipo de estructuras tabulares. Además, éstas permiten identificar las informaciones contenidas en ellas sin posibilidad de ambigüedad alguna.
- En tercer lugar, una relación normalizada es una estructura más simple que una no normalizada, al menos matemáticamente hablando. Esto permitirá también desarrollar una serie de operaciones para manipular esas informaciones que ya nos vienen dadas por el lenguaje matemático.

- Por otra parte y en relación con lo anterior, partir de una estructura de información como ésta nos permitirá contar con una teoría sólida que facilitará el desarrollo de procesos que manejen con eficacia la información. En torno a esta teoría y sus posibles desarrollos existen infinidad de trabajos.

En resumen, desde una perspectiva estrictamente de definición del modelo, la atomicidad es una característica inevitable, aunque tampoco se puede evitar que existan valores de datos compuestos. Bien es verdad que no es ésta la única razón para cuestionar este principio de la atomicidad. Como dije antes, la atomicidad significa que las informaciones contenidas en las relaciones no tienen estructuras internas, porque es la relación quien tiene esa estructura interna de atributos y tuplas. Pero eso no significa que no haya cierto tipo de datos que exigen la existencia de una cierta estructura interna. Me estoy refiriendo concretamente a datos como fechas, que están compuestos y se pueden subdividir en distintos elementos. Esto ha dado lugar a que diversos productos llamados relacionales implementen operadores o comandos para manejar la descomposición en distintos elementos del contenido del valor del atributo.

No hay, por otra parte, ninguna razón importante —excepto el hecho de que la atomicidad representa una buena disciplina para manejar la información— en la teoría del modelo relacional para justificar esta atomicidad; y ésto teniendo en cuenta que Codd insiste en que para ser fiel al modelo, los datos deben ser atómicos [Codd 70]. A pesar de todo, autores como Date han puesto de manifiesto que la atomicidad no es un requerimiento imprescindible, sino, simplemente, un requisito deseable en la mayor parte de las situaciones, aunque no en todas [Date 90].

Para mí, esta característica de la estructura de datos relacional es un tema clave, porque, desde distintos puntos de vista, parte de la información que una organización bibliotecaria maneja, cuanto menos resulta dudoso que no tenga que ser una información compuesta, y además una información difícilmente estructurable. Recuérdese aquí la diferencia entre información estructurada y no estructurada que hice en el segundo capítulo, porque, en mi opinión, tiene una gran relación con el concepto de atomicidad. De principio, todas las informaciones que yo he considerado estructuradas, serían informaciones fácilmente normalizables, mientras que las informaciones que he considerado como no estructuradas serían difícilmente normalizables en el sentido relacional del término.

A pesar de la anterior propuesta de Date, que niega la necesidad de considerar la atomicidad como un principio del modelo, sino más bien como una norma a tener en cuenta en el diseño en las bases de datos, son Date y otros autores los que dicen que los sistemas de bases de datos actuales no son particularmente buenos cuando trabajan con informaciones complejas, tales como gráficos o información textual no estructurada [McLeod 89].

Volviendo a los conceptos enunciados anteriormente, que componen la estructura de datos relacional y más concretamente el concepto de dominio, podríamos decir ahora que un dominio es un conjunto de valores escalares y, lo que es más importante, todos esos valores escalares o atómicos, deben ser del mismo tipo. Esta característica de la tipología común de los valores que forman el dominio tiene una importancia decisiva a la hora de gestionar los datos, pues permitirá al sistema determinar, como veremos después, cierto tipo de incoherencias lógicas en la definición de los procesos.

Como ya dije anteriormente, un dominio puede estar compuesto por uno o varios atributos, ahora bien, todos los atributos deben pertenecer a algún dominio concreto; de tal forma que en todo momento se sepa, al trabajar con los valores de los atributos, si aquéllos que se relacionan mediante operadores pertenecen al mismo dominio o a dominios distintos.

Si nosotros pretendiéramos comparar dos atributos del mismo dominio mediante la utilización de cualquier operador, tendríamos que admitir que al pertenecer al mismo dominio, la operación, como tal, tiene sentido. Sin embargo, si esa misma operación se intenta realizar entre atributos que pertenecen a dominios distintos, probablemente no tenga ningún sentido. Pongamos, por ejemplo, el caso de dos campos del formato MARC, como son el campo de serie como encabezamiento secundario y el campo de serie cuando no es encabezamiento secundario (etiquetas 440 y 490 respectivamente). Las informaciones pertenecientes a esos dos campos del formato MARC pueden pertenecer en una estructura tabular, de acuerdo con el modelo relacional, a dos atributos distintos, pero ambos al mismo dominio. Si en una operación de búsqueda normal hacemos una comparación para localizar una determinada información bibliográfica a partir del nombre de una serie, podremos comparar los valores de un campo con los de otro, pues ambos pertenecen al mismo dominio. Sin embargo, si pretendiéramos comparar los valores de esos campos con los valores del atributo donde se incluyera el encabezamiento de autor personal, la comparación no tendría ningún sentido. Sería algo así como si yo pretendiera buscar por autor en el catálogo de series.

En este sentido es en el que digo que el hecho de que un sistema relacional soporte la gestión de los dominios, sirve para controlar cierto tipo de manipulaciones de datos erróneas que quieran poder hacer, en un momento determinado, los usuarios. Por controlar entiendo la detección de una operación ilógica, desde el punto de vista del concepto del dominio, y que el sistema al mismo tiempo informe al usuario de la naturaleza del

problema. Todo esto sin olvidar que el dominio, por definición, es un elemento conceptual de la estructura y no forma parte de la base de datos como un conjunto de valores más, a no ser que se trate de un dominio que sólo tiene un atributo. En cuyo caso, al coincidir atributo y dominio, los valores que forman el atributo son también los que forman el dominio, y en ese caso sí que se podría identificar con una parte concreta de la base de datos. En realidad, el concepto de dominio tiene más su sentido en relación con el control de la manipulación de los datos que con la estructura misma de esos datos.

La forma en que un sistema relacional opera con el concepto de dominio permite, como he dicho antes, controlar posibles errores en la manipulación de los datos, aunque existen muy pocos sistemas comerciales que soporten este concepto y especialmente por lo que se refiere a la existencia de los llamados dominios compuestos, que están formados, a su vez, por varios dominios simples. Estos dominios compuestos, como combinaciones de dominios simples, facilitarían la gestión de datos con estructura interna que se pueden descomponer en varios subdatos. La descomposición se realiza en base al nombre de cada uno de los dominios simples que forman el compuesto, pero también en base a la posición que ocupa cada dominio en el conjunto.

En resumen, como se puede ver, el concepto de dominio es más complejo de lo que parece a simple vista, pues al ser en realidad el dominio un tipo determinado de dato, exige primero, por parte del sistema, la posibilidad de definir unas características que pueden ser muy variables, a veces a instancias del propio usuario, para especificar el tipo de datos de que se trata en cada caso. Pero al mismo tiempo el sistema deberá disponer de unos mecanismos para controlar los procesos en base a las características definidas de cada dominio. Esto, especialmente, de cara al control de lo que se llaman operaciones de comparación entre dominios. Por último, la disponibilidad de una amplia gama de operadores para realizar comparaciones complica, aún más, la gestión del concepto de dominio.

Para resumir, los requerimientos por parte del modelo relacional, en lo que se refiere al concepto de dominio, son los siguientes:

— La posibilidad de especificar un completo conjunto de dominios, que se puedan aplicar a cualquier base de datos.
— La posibilidad de especificar para cualquier dominio, qué operadores exigen la existencia de dominios comunes al comparar distintos valores y cuáles no. Y eso, tanto por lo que se refiere a operadores unarios como a operadores binarios.

Una vez hechas estas consideraciones sobre el concepto de dominio, estoy en disposición de abordar la definición de lo que considero que es el concepto básico del modelo relacional. Se trata del concepto de relación.

Desde un punto de vista estrictamente matemático, una relación debe ser entendida como una colección de dominios, no necesariamente distintos, o también como un conjunto de elementos. En el modelo relacional la relación consta de dos partes, llamadas respectivamente cabecera y cuerpo de la relación. Si utilizamos una terminología más al uso, en lugar de relaciones deberíamos hablar de tablas, puesto que la representación usual de una relación es una tabla. Siguiendo esta misma terminología, la cabecera de la relación sería la fila que encabeza todas las columnas, mientras que el cuerpo estaría formado por el conjunto de datos contenidos en las restantes filas. De todas maneras, desde la perspectiva de los lenguajes de programación quizá sea más acertado considerar que una relación representa una variable. El encabezamiento, de esta manera, sería el tipo de la variable, mientras que el cuerpo estaría representado por el valor de dicha variable. En cualquier caso, parece que, en base a los conceptos de cabecera y cuerpo, podríamos intentar una aproximación al concepto de relación.

La cabecera está formada por el conjunto de los atributos o, dicho de una manera más precisa, por el conjunto de los pares formados por los dominios y los atributos. En una formalización matemática de esta idea diríamos que:

(D1:A1),(D2:A2)..., (Dn:An)

En cuanto al cuerpo de la relación estaría formado por el conjunto de las tuplas o filas que componen dicha relación, considerando que dichas tuplas están formadas a su vez por un conjunto de n pares valor/atributo. La formalización en este caso sería:

(VI1:A1), (VI2:A2)..., (VIn:An)

teniendo en cuenta que I representa el número de tuplas que tiene dicha relación, mientras que n representaría el número de atributos. Estas dos variables son la expresión a su vez de dos conceptos enunciados anteriormente, son los conceptos de cardinalidad y grado de la relación. El número de tuplas que tiene una relación es su cardinalidad, mientras que su grado está representado por el número de atributos. En este sentido, una relación de grado 1 se llama unaria, una relación de grado 2 se llama binaria, una relación de grado 3 se llama ternaria... y así hasta llegar a una

relación de grado n, que se llama n-aria. El modelo relacional ha sido definido para poder tratar, precisamente, relaciones de grado n.

En relación con el concepto de cardinalidad hay que decir que su valor en una relación es algo muy variable, lo contrario de lo que ocurre con el grado. Esto se debe a que las relaciones aumentan o disminuyen constantemente el número de tuplas que las componen, mientras que su grado no cambia normalmente, a no ser en circunstancias excepcionales.

Todas las relaciones poseen unas determinadas propiedades, además de los elementos ya definidos. Estas propiedades son una consecuencia de las características que hasta ahora he mencionado. La primera de esas propiedades podría ser que en una relación no hay tuplas duplicadas. Esta propiedad, como otras, es consecuencia del hecho de que la relación, como concepto, surge como consecuencia del concepto matemático de conjunto, y éste no permite la existencia de elementos duplicados. En consecuencia, los sistemas que se desarrollan siguiendo este modelo relacional, no deben permitir la existencia de tuplas duplicadas.

Una cuestión relacionada con esta propiedad es que toda relación tiene claves primarias. Esto querría decir, en primer lugar, que cada tupla en una relación es única. Y esto porque en cada tupla existe un valor de atributo que es diferente de todos los demás atributos existentes en la relación. Esta característica será más ampliamente desarrollada cuando defina el concepto de clave primaria.

En algunos casos, al desarrollar sistemas de gestión de bases de datos relacionales, se ha considerado la necesidad de incluir atributos que no tienen ningún interés, de cara a la información que se pretende procesar, pero que permiten cumplir con la propiedad de la inexistencia de tuplas duplicadas. Esto es absolutamente innecesario, sobre todo si se tiene en cuenta que se puede acudir al procedimiento de la clave primaria como una clave compuesta de varios atributos.

Otra característica importante de las relaciones es que las tuplas están siempre desordenadas. También es una consecuencia del concepto matemático de conjunto, pues los conjuntos en matemáticas contienen una serie de elementos que no se encuentran en ningún orden. En este sentido, no se puede establecer ningún mecanismo de direccionamiento posicional de la información contenida en las relaciones. Por la misma razón también tendremos que decir que los atributos, dentro de una relación, no tienen ningún orden. En realidad, al ser éstos también considerados como un conjunto, es lógico pensar que tampoco pueden estar ordenados.

Aunque de esta cuestión hablaré más adelante, quizá aquí se podría introducir la idea de que al modelo relacional no le hace falta la consideración de los atributos como elementos ordenados, porque su identificación se hace siempre en base a la denominación de los mismos. Es decir, es inherente a la existencia del atributo que reciba un nombre concreto que le diferencie de los demás atributos de la relación. Como suele ser costumbre en el tratamiento de las bases de datos relacionales, la denominación de los atributos se hace mediante la asociación del nombre de dominio más el nombre de atributo.

Por último —y aunque esto ya ha sido mencionado anteriormente, creo que es importante repetirlo aquí como una propiedad más de las relaciones— todos los atributos deben contener valores escalares o atómicos. Esto es absolutamente necesario para poder desarrollar el modelo relacional, porque, como ya expliqué anteriormente, la manipulación de datos no atómicos sería inviable, de acuerdo con el modelo que estoy describiendo. Dicho de otra manera, en cualquier posición de fila y columna dentro de una tabla siempre existirá un solo valor, nunca una lista de valores. Pero, como esta cuestión ya ha sido suficientemente desarrollada anteriormente, no insistiré ahora más en ella.

Las relaciones, además de tener una serie de propiedades o características, también pueden ser de distintos tipos. Es precisamente Date quien define, recogiendo referencias de otros autores, diversos tipos que creo son importantes; si no todos por igual, sí en el orden en que los voy a dar, de más a menos [Date 90c].

- En primer lugar las relaciones que se denominan relaciones base o relaciones reales. Este tipo, está formado por aquellas relaciones que son lo suficientemente importantes en el conjunto que forman las bases de datos, como para que hayan sido definidas de forma expresa en la fase de diseño de la base y se les hayan asignado todos los elementos que conforman la relación. Al mismo tiempo, éstas servirán como base para desarrollar otras relaciones. En cierto modo, se puede decir que la base de datos está formada fundamentalmente por este tipo de relaciones. Creo que no estaría de más establecer aquí un paralelismo entre lo que yo he llamado en el capítulo anterior de este trabajo entidades primarias y las relaciones base de las que estoy hablando ahora, pues el concepto del que me he servido para elaborar esas relaciones base es el mismo que estoy definiendo en estas líneas.
- El segundo tipo es el de las llamadas relaciones virtuales o vistas. Se trata de una relación derivada de las relaciones base. Son unas relaciones que utilizan partes de los atributos de las relaciones base para existir y, aunque están definidas de manera permanente, tienen un carácter muy efímero, puesto que sus valores sólo conforman el cuerpo de la relación en el momento en que son requeridas por el usuario.

- Un caso parecido, aunque con ciertas diferencias, es el de las relaciones instantáneas. Son muy parecidas a las relaciones virtuales, con la única diferencia de que la relación instantánea existe realmente. Es decir, la información que contiene, el valor de sus atributos, en definitiva, se encuentra almacenado físicamente en un lugar concreto y no se configura expresamente en el momento en que el usuario hace una llamada a dicha relación. Lo que ocurre es que, aunque la relación existe realmente no es una relación básica, pues se trata de una relación cuyos valores y atributos han sido extraídos de relaciones básicas.
- Otro tipo de relación podría ser el denominado resultado de interrogaciones. Se trata de aquellas relaciones que se forman como consecuencia de una petición de información que hace un usuario. Las salidas de esa información que se entrega al usuario forman una nueva relación, de acuerdo con una serie de especificaciones condicionales que el usuario ha establecido.
- Otro tipo de relación sería la de resultados intermedios. Se produce cuando la petición de información que el usuario hace es una petición compleja que requiere varias fases en su tratamiento y, al final de cada una de esas fases, se produce una relación que llamamos intermedia, y de la que, muchas veces, el usuario ni siquiera percibe su existencia.
- Por último, existen también las llamadas relaciones temporales, que podrían ser consideradas, tanto como relaciones instantáneas o, incluso, como virtuales; pero con la diferencia de que estas relaciones temporales sólo existen el tiempo que el sistema las necesita para realizar algún proceso, e inmediatamente después las hace desaparecer.

Como ya dije al principio de la descripción del modelo relacional, además de existir una estructura de datos basada como vemos en los conceptos de dominio y relación, existen también, como parte fundamental del modelo, las llamadas reglas de integridad relacional, que paso a describir a continuación.

IV.B. Prescripciones de integridad

El conjunto de información que se almacena en una base de datos del tipo que sea está formado por una serie de valores que de manera más o menos certera reproducen una parte de la realidad. Esto significa que la lógica de los valores que forman la base de datos viene impuesta por la lógica de la realidad, pero esto no impide que el sistema que controla ese conjunto de información disponga de una serie de reglas, que llamaremos de integridad, cuyo objeto fundamental será replicar, en la medida de lo posible, en el sistema la lógica del mundo real mediante la existencia de una serie de imperativos que el sistema verificará en el momento de la realización de cualquier manipulación de datos, de tal manera que determinadas operaciones con los datos serán consideradas inviables y el propio sistema lo comunicará al usuario.

Sin embargo, es necesario tener en cuenta que estas reglas de integridad deberían ser definidas específicamente para cada conjunto concreto de datos, es decir, para cada base de datos; porque la lógica de los datos, al ser un reflejo de la lógica del mundo real, debe ser diferente según la parcela de la realidad que estos datos reflejen. No obstante, el modelo relacional en su definición incluye unas reglas de integridad, de carácter general, aplicables a cualquier conjunto de datos, y es precisamente de éstas de las que quiero hablar ahora.

Realmente el modelo relacional sólo define dos reglas, que tienen que ver con la existencia de las claves primarias y las claves secundarias [Date 90d]. Como ya insinué anteriormente, una clave primaria de una relación es el identificador único de la relación. Por ejemplo, si consideramos que los datos referentes a un usuario de una biblioteca forman una relación, podemos considerar como clave única de esa relación el número de usuario, puesto que se trata de una información que debe identificar sin posibilidad de error, confusión o ambigüedad a cada usuario. De todas maneras, el modelo relacional admite la posibilidad de que existan claves primarias compuestas, de tal forma que la identificación de la tupla, dentro de la relación, se pueda realizar en base a varios atributos.

Esto no significa que las claves primarias sean la única forma de acceso para el usuario a la información contenida en una relación. Muchas veces se confunde, incluso de manera interesada, el concepto de clave primaria con los llamados índices de una tabla. Son cosas totalmente distintas. Los índices son instrumentos para facilitar el acceso a la información y la clave primaria es el instrumento que el modelo relacional exige para identificar cada fila de una tabla. Por otro lado, también hay que decir que es perfectamente posible que una tabla tenga más de un identificador. Sin embargo, aunque en teoría es posible, habrá que definir, cuáles de los atributos que pueden ser identificadores únicos, serán considerados por el sistema como claves primarias, de tal forma que los otros, a partir de ese momento, pasarán a ser tratados como candidatos, pero no como claves primarias. Estos atributos candidatos, en realidad, son claves alternativas a la primaria.

En este sentido, habría que hacer varias observaciones:

- En primer lugar, una clave candidata puede ser considerada como tal si cumple las siguientes dos condiciones: la de que no existan dos tuplas en la relación que tengan el mismo valor en la clave candidata y la de que, si la clave es compuesta, ningún componente de dicha clave pueda ser eliminado sin que se incumpla la primera condición.
- Por otra parte, del conjunto de claves candidatas de una relación, únicamente una será considerada como primaria. El resto serán claves alternativas, teniendo en cuenta que, en toda relación, al menos habrá una candidata y por tanto una primaria. Esta cuestión, además, según el modelo no tiene excepción posible, pues si es precisamente la clave primaria la que puede identificar las filas de la tabla, sin clave primaria no habría posibilidad de acceso a la información de la tabla.

Las razones por las que de entre las distintas claves candidatas se elige una para convertirse en clave primaria, no tienen nada que ver con el modelo relacional en cuestión, sino que más bien tendrían que ver con la lógica que utiliza el diseñador a la hora de crear la base. Pero, sin embargo, el modelo sí que hace en este sentido una recomendación, y es que la clave primaria sea un elemento suficientemente significativo. Por otro lado, si ponemos en relación el concepto de clave primaria con los tipos de relaciones vistos anteriormente, tendremos que concluir que las claves primarias siempre forman parte de las relaciones base, aunque, por supuesto, esto no quite para que puedan formar parte de otro tipo de relaciones, pero sin olvidar que originalmente han sido creadas a partir de relaciones base.

Quizá la conclusión más importante que se puede extraer de este conjunto de consideraciones en torno al concepto de claves primarias es que éstas son la herramienta básica utilizada en el direccionamiento del sistema relacional, por lo menos en lo que se refiere a las tuplas. Ello implica que el direccionamiento del sistema relacional es un tipo de direccionamiento asociativo, es decir, que está fundamentalmente basado en los contenidos, en las informaciones, en los valores, no en las posiciones. Esto tiene unas claras implicaciones desde el punto de vista del funcionamiento de los sistemas de software básico en los que se basa el modelo relacional e incluso tiene implicaciones de tipo hardware, aunque por supuesto, los sistemas convencionales de almacenamiento y los métodos de acceso avanzados que soportan los sistemas operativos actuales son más que suficientes para permitir direccionamientos asociativos.

Una vez definido el concepto de clave primaria, estamos en condiciones de desarrollar la primera de las dos reglas de integridad generales del modelo relacional. Se trata de la regla de integridad de la entidad, que se puede enunciar de esta forma:

> No existen componentes de la clave primaria de una relación base que sean nulos. Hay que entender por componentes nulos aquellos atributos de una tupla dada en los que no existe valor alguno o éste es desconocido [Date 90e].

El razonamiento en que se basa esta regla es bastante simple: las relaciones base se corresponden con entidades del mundo real. Estas se tienen que distinguir unas de otras, pues son representativas de parcelas de la realidad y por tanto deben ser identificables. En el modelo relacional el proceso de identificación establecido es el de las claves primarias. Si existieran valores nulos en las claves primarias, esto querría decir que determinadas parcelas de la realidad no podrían ser identificadas. De acuerdo con el modelo esas tuplas con valores nulos, al no ser identificadas de manera alguna no tendrían sentido por no reflejar ninguna realidad. Date es muy tajante en este sentido y, siguiendo a Codd, dice que una entidad sin identificar no existe. Por tanto, en una base de datos relacional nunca se debe almacenar información sobre algo que no puede ser identificado [Codd 90].

Esta primera regla tiene, a su vez, una serie de implicaciones, como, por ejemplo [Date 90e]:

— Que todo valor individual de una clave primaria debe estar completo.
— En segundo lugar, que la regla sólo es aplicable en el caso de las relaciones base y, por tanto, sólo para las claves primarias, no para las claves alternativas.

Una vez definidas las claves primarias y su funcionamiento, estamos en condiciones de abordar un concepto que se deriva de éste. Una de las posibilidades inmediatas del modelo relacional es la de permitir el establecimiento de conexiones entre las relaciones base, para dar lugar a una nueva relación. En este tipo de operaciones desempeñan un papel importante las claves primarias, de tal forma que, a partir de una relación base o de varias, se puede elaborar una relación secundaria cuya clave sea idéntica a alguna de las claves primarias de las relaciones a que dan lugar. A esta clave, igual que a una primaria de otra relación, se la llama clave secundaria.

Las claves primarias que se corresponden con alguna clave secundaria pueden contener valores que no pertenezcan, como tales valores, a la clave secundaria con la que se corresponden. Este sería, por ejemplo, el caso de la relación de usuarios de una biblioteca y la relación de préstamos de esa biblioteca. En la relación de usuarios puede existir como clave primaria el numero del usuario, mientras que ese mismo número de usuario sería clave secundaria en la relación de préstamos. Pues bien, algunos de los números de usuarios en la relación de usuarios no tendrían por qué haber recibido ningún libro en préstamo y, por consiguiente, su número no aparecería en la relación de claves secundarias de préstamos. Esto es, sencillamente, una manifestación de lo que decía anteriormente: los contenidos de las claves primarias no tienen por qué ser reproducidos al completo en las claves secundarias.

Sin embargo, sí que existe un problema constante en todo el modelo relacional, que a veces se convierte en una verdadera obsesión, puesta de manifiesto en numerosos comentarios de sus creadores, al intentar garantizar que las bases de datos no incluyan como claves secundarias válidas valores de claves secundarias inválidas, es decir, que no se corresponden con valores de claves primarias, pues si esto fuera así, como veremos después, no se satisfacen algunas de las condiciones fundamentales de la regla de integridad correspondiente. A este problema se le llama problema de la integridad referencial, que, por otra parte, está muy relacionado con el imperativo de que los valores de las claves secundarias deben igualarse o emparejarse con valores de sus pares correspondientes en las claves primarias. A este imperativo se le conoce con el nombre del imperativo referencial.

Vistas las cosas de esta forma, se puede decir, entonces, que las claves secundarias deben cumplir dos condiciones fundamentales [Date 90e]:

- La primera es que cada valor de una clave secundaria es siempre o bien completamente nulo o completamente no-nulo.
- Por otra parte, se debe cumplir también la condición de que siempre que exista una relación base con su correspondiente clave primaria y, a su vez, una segunda relación con una clave secundaria asociada a esta primaria, los valores no nulos de la clave secundaria deben ser idénticos a los valores de la clave primaria en las tuplas correspondientes.

Estas dos condiciones, que deben ser cumplidas por las claves secundarias, traen consigo, a su vez, otra serie de condiciones. Así, por ejemplo, una clave secundaria y su correspondiente clave primaria deben ser definidas sobre la base del mismo dominio. No es concebible que si las claves

secundarias se emparejan con las claves primarias y reproducen por tanto sus valores, que ambas claves pertenezcan a dominios distintos. Por la misma razón, no es obligatorio que una clave secundaria sea un componente de una clave primaria. Esto es importante tenerlo en cuenta pues, en ocasiones —las más de las veces en el caso que nos ocupa— se produce una situación según la cual la clave secundaria es un elemento de la clave primaria. Pero esto no siempre tiene que ser así, sobre todo en los casos en los que se puedan elaborar claves secundarias compuestas y alguno de los elementos de la clave compuesta puede ser la clave primaria, pero no todos. Asimismo, es importante tener en cuenta que una relación puede ser, al mismo tiempo, la relación que sirve de referencia para la creación de una clave secundaria y, ella misma, referenciada, desde otra relación para contener una clave secundaria; de tal manera que, en principio, la clave secundaria de una determinada relación puede actuar, en definitiva, como clave primaria de cara a una tercera relación. Por otra parte, como consecuencia de lo expuesto anteriormente, las claves secundarias aceptan, en contra de las claves primarias, en ocasiones, la existencia de valores nulos.

Después de haber repetido mucho la expresión «valores nulos», quizá sea útil aclarar que en todo momento la estoy utilizando como sinónimo de ausencia de valor, es decir, valor nulo sería el de un atributo que en una tupla dada no ha sido cargado con ningún contenido. Es lo que se podría denominar información desaparecida.

Aunque pueda aparecer esta explicación innecesaria, creo que es importante reseñar que el tratamiento de los valores nulos tienen una importancia decisiva, de cara, sobre todo, al diseño de los sistemas relacionales, porque muchas de las operaciones que sólo son posibles, según el modelo relacional, en circunstancias muy especiales, están relacionadas con el tratamiento de estos valores nulos.

Por último, volviendo al tema que nos ocupa, decir que las claves secundarias actúan en las bases de datos como el aglutinante de las relaciones, de tal manera que son el vehículo de unión entre las distintas relaciones, puesto que obligatoriamente tienen que estar igualadas con claves primarias que se encuentran en otras relaciones.

Este concepto de claves secundarias es lo que da lugar a la segunda regla general de integridad dentro del modelo relacional, que se denomina regla de integridad referencial y se enuncia de la siguiente manera [Date 90]:

> La base de datos no debe contener ningún valor de clave secundaria desparejado de su correspondiente primaria.

Quizá lo que necesitaría de alguna precisión es la expresión valor desparejado, que aquí se está refiriendo a aquellos valores de las claves secundarias que no son nulos y para los que podrían no existir un valor igual en la clave primaria que da origen a esa secundaria.

Como quedó explicado suficientemente en el apartado de las claves primarias, solamente los valores de este tipo de claves, pueden ser considerados identificadores de entidad, es decir, los valores de las claves primarias sirven para identificar claramente las tuplas dentro de las relaciones. Como consecuencia del proceso de elaboración de las claves secundarias, es necesario aclarar que sus valores no tienen la función identificadora que tienen las claves primarias, sino una función referencial. En este sentido la regla de integridad referencial lo que dice es que si un valor referencia a otro, este otro tiene que existir. Entre otras cosas, esto significa que la integridad referencial exige claves secundarias iguales a claves primarias, no a claves alternativas. Y, al mismo tiempo, se puede decir que los conceptos de claves secundarias e integridad referencial están definidos de tal manera, que en los productos relacionales decir que dan soporte a la integridad referencial y que dan soporte a las claves secundarias significa lo mismo.

Sin embargo, esta aparente simplicidad de los conceptos puede no dejarnos ver cuáles son las consecuencias de tipo práctico que esta regla de integridad tiene en el tratamiento de la información. Por ello, para tratar de ahondar en esas consecuencias de tipo práctico, me gustaría terminar haciendo algunas observaciones en este sentido.

He insinuado anteriormente que las claves secundarias, en ocasiones, podrían tener valores nulos, lo que, en apariencia, parece contradictorio con el hecho de que las claves secundarias se deben igualar con valores de las claves primarias y, éstas, a su vez no deben duplicarse. También he explicado que esto puede ocurrir, sobre todo, porque puede haber valores compuestos de claves secundarias que no sólo estén conformados por valores de claves primarias, sino también por otros valores distintos y esto podría dar lugar a situaciones de esta naturaleza. Pero también puede ocurrir, simplemente, por el hecho de que determinado valor de clave secundaria no se corresponde con ningún valor de clave primaria. Esta situación, en el caso del sistema que estoy definiendo, no se dará nunca, pues al definir las especificaciones de este sistema, en cada caso la coherencia informativa del mismo impedirá este tipo de desajustes.

Probablemente convenga poner algún ejemplo de situaciones en las que se puedan dar este tipo de casos. Partamos de un sistema en el que exista una relación de usuarios que contenga, entre otros atributos, «número del usuario» y «tipo de usuario». Asimismo, existirán relaciones independientes de ésta de préstamos y de tipos de usuarios. En el caso de la

relación de usuarios, la clave primaria sería número del usuario, mientras que en la de préstamo la clave secundaria sería, precisamente, el número del usuario, mientras que la clave primaria sería el número del préstamo. En el caso de la relación de tipos de usuarios, la clave primaria sería el número del tipo, mientras que en la relación de usuarios, el número del tipo de usuarios sería la clave secundaria.

Descritas estas relaciones, podría admitirse —aunque ya digo, que en el sistema que yo definiré no será así— que un determinado usuario no perteneciera a ninguno de los tipos definidos en la relación de tipos, lo que quiere decir que el número del tipo que aparece como clave secundaria en la relación de usuarios estaría como nulo, o, si se quiere en blanco, mientras que los restantes usuarios podrían pertenecer a su tipo correspondiente y, por consiguiente, el atributo «tipo de usuario» tendría valor.

En este sentido es en el que digo que es técnicamente posible, por lo menos se define así, que ciertos valores de las claves secundarias estén sin asignar. Pero, ¿qué ocurriría en el caso de que se intentara realizar cualquier tipo de operación, como, por ejemplo, borrar o modificar las claves secundarias, o las claves primarias referenciadas por las claves secundarias?

Analicemos un caso en el que intentamos borrar una clave primaria que está siendo referenciada por una secundaria, por ejemplo, intentamos borrar un usuario —una tupla de usuario— que, como he dicho antes, tiene un atributo «número del usuario» utilizado para generar la clave primaria, y ese usuario tiene libros en préstamo y, por consiguiente, existen tuplas de la relación de circulación donde el número de ese usuario aparece como clave secundaria dentro de esas tuplas.

¿Qué debería hacer el sistema en un caso como éste? Debería hacer una de estas 3 cosas [Date 90]:

1. Limitar la posibilidad de borrar una tupla de la relación de usuarios al caso de que no haya ninguna referencia a esa tupla en otras relaciones. Por consiguiente, en este caso no se permitiría el borrado de la tupla en cuestión.
2. Borrar la información en cadena. Si se pretende borrar la tupla de un usuario que tiene libros en préstamo, el sistema borra la tupla del usuario y las tuplas de los documentos prestados a ese usuario.
3. Se podría considerar la posibilidad de convertir en nula la clave secundaria en la relación de circulación y borrar la tupla del usuario, es decir, sería tanto como cumplir con la intención del usuario de anular esa tupla, pero, al mismo tiempo, conservar la información referencial sin la referencia. En este caso, los valores

secundarios de la clave han sido convertidos en nulos, por si esta información referencial pudiera servir en el futuro para alguna cosa. Esto, siempre y cuando en el sistema se admita la existencia de valores nulos en las claves secundarias.

De estas tres opciones yo elegiría, para el caso del sistema que estoy tratando de diseñar, la primera opción; porque me parece que es la que más garantías de seguridad ofrece. En el caso de que lo que se pretendiera no fuera borrar, sino modificar la información, las opciones son las mismas: la de la modificación limitada, la de la modificación en cadena y la de convertir en nulas las claves secundarias.

En cualquier caso, como principio general, el modelo relacional, lo que establece es que forma parte del diseño de las bases de datos definir cuáles serán las condiciones de manipulación de las claves secundarias respecto de las primarias, y esto incluye la elección de una de las tres posibilidades que acabo de enunciar, aunque se podrían definir otras alternativas, algunas de ellas muy ligadas a la naturaleza misma de la información que se está tratando de manipular. Probablemente, entrar en la casuística que se puede dar, nos llevaría muy lejos. Por eso, lo importante es dejar constancia del hecho de que el diseñador del sistema debe establecer las condiciones de manipulación de las claves secundarias, de acuerdo, por su puesto, con la definición de la regla de integridad referencial que he descrito anteriormente.

IV.C. Manipulación de los datos

Hasta aquí, básicamente, lo que el modelo relacional ha hecho ha sido definir unos conceptos que permitirán normalizar la información para ser manejada por un sistema relacional y definir también unas reglas que nos permitirán operar con esa información previamente estructurada. Pero, como queda dicho, las reglas son de carácter muy general y necesitamos unos mecanismos para efectuar manipulaciones concretas de los datos. De esto es, precisamente, de lo que vamos a hablar en este tercer apartado.

En la definición que hizo Codd del modelo relacional incluyó, desde el principio, un conjunto de operadores que formaban lo que él llamó en su momento el álgebra relacional. Junto con esos operadores también había un conjunto diferente de lo que llamó «asignadores», que permitían hacer asignaciones de información a los distintos elementos de la estructura relacional [Codd 70]. Precisamente, tanto de los operadores como de los asignadores es de lo que voy a hablar a continuación.

Son los operadores los que forman la llamada «álgebra relacional», que está formada por ocho operadores que se utilizan para manipular las relaciones. Los operadores funcionan tal y como están descritos en las matemáticas. Normalmente, cada operador necesita de, al menos, una relación como entrada y produce otra relación como salida.

Los operadores definidos por Codd quedaron divididos en dos grupos [Codd 90]:

- El primero está formado por los operadores tradicionales de las matemáticas de conjunto: unión, intersección, diferencia y producto cartesiano. Esto no quiere decir que estos operadores sean exactamente iguales a los operadores de las matemáticas de conjuntos, aunque se parecen bastante y su origen es precisamente ese, por eso han recibido los mismos nombres.
- Los otros cuatro los consideró como un grupo especial de operadores porque no se parecían a los de las matemáticas de conjuntos. A esos los llamó: restricción, proyección, join y dividido.

A continuación, pasaré a describir cada uno de estos ocho operadores y pondré ejemplos de su funcionamiento con relaciones propias de un sistema de bibliotecas. Pero antes me gustaría insistir en la idea expresada anteriormente de que los operadores necesitan relaciones como entradas, que pueden ser una o varias y que, a su vez, generan varias relaciones como salida. Esto significa que la salida de una operación en la que interviene un operador, puede ser entrada de otra operación, en la que intervendrán otros operadores. Además, esto significa también, como ocurre con todas las operaciones, que es posible realizarlas de forma anidada, es decir, representarlas como expresiones con paréntesis y esto tendrá unas implicaciones importantes, por lo que se refiere a las propiedades asociativas, conmutativas, etc.

Por último quiero insistir en la idea de que la salida de cada operación es una relación. Esto, desarrollado en detalle, podría llevarnos algún tiempo. Aquí sólo quiero hacer alguna consideración de carácter general, en el sentido de que si esta cuestión se plantea de manera estricta, de acuerdo con las especificaciones ya enumeradas del modelo relacional, en lo que se refiere a la composición de una relación, habría que decir que, cualquier sistema debería ser capaz en el curso de una operación de crear nuevas relaciones con todos sus elementos. Y esto hay que entenderlo desde un punto de vista estrictamente conceptual, pues en la práctica, a veces, plantea ciertas dificultades. En este sentido, es importante tener presente que las salidas de las operaciones solamente necesitarán ser una salida de relación normalizada, en el caso de que vayan a ser considerada

como entrada de otra relación, lo que nos exigirá cumplir muy escrupulosamente con cierto tipo de reglas, que veremos a continuación, en el uso de estos operadores.

Para hablar del primer grupo de estos operadores, primero, habrá que introducir un concepto previo, que es el de compatibilidad para la unión, puesto que, realmente, si no establecemos unas condiciones mínimas, previas a la realización de las operaciones, éstas nos llevarían a situaciones que no tendrían ningún reflejo en la realidad. Así, por ejemplo, si pretendiéramos unir la relación de usuarios con la relación de proveedores, lo que obtendríamos sería una relación absurda que tiene un conjunto de información sin referente real alguno. Por ello, precisamente, es importante establecer —como garantía de coherencia previa— la necesidad de una compatibilidad para la unión que luego podremos utilizar como requisito, también, en otras operaciones. Esto quiere decir que el operador unión del modelo relacional no es exactamente igual que el operador unión de las matemáticas de conjuntos, porque este tipo de restricciones no existían en las matemáticas de conjuntos. Esta compatibilidad para la unión significa que las relaciones que se pretenden unir deben tener idénticas cabeceras, o, lo que es lo mismo, deben tener cada una el mismo conjunto de nombres de atributos y que estos atributos, que forman las relaciones, están definidos en los mismos dominios. Todos los operadores de este primer grupo, excepto el de producto cartesiano, requieren la compatibilidad para la unión.

Hecha esta consideración previa, podemos pasar a definir el «operador unión». Este operador sirve para crear una nueva relación que está formada por el conjunto de tuplas pertenecientes a las dos relaciones unidas por este operador.

Como ejemplo de este tipo de operación podríamos poner, en un sistema relacional que gestionara una red de bibliotecas, el operador unión, que podría servir para elaborar una relación de los usuarios de dos bibliotecas distintas mediante la unión de las relaciones de usuarios de esas dos bibliotecas.

En cuanto al operador intersección, sirve para crear una relación que está formada por las tuplas comunes a las dos relaciones unidas por este operador.

Utilizando las relaciones de usuarios de las dos bibliotecas anteriores, el operador intersección podría servir para determinar los usuarios comunes a ambas bibliotecas.

En cuanto al operador diferencia, sirve para determinar el conjunto de tuplas que perteneciendo a la primera relación, sin embargo, no pertenecen a la segunda de las relaciones unidas por este operador.

Utilizando el ejemplo anterior, mediante el uso de este operador, podríamos averiguar el conjunto de usuarios de la biblioteca uno que no lo son de la biblioteca dos.

Por último, el producto cartesiano, que Date define como producto cartesiano extendido, para diferenciarlo del producto cartesiano de las matemáticas, es un operador que funciona de la siguiente manera: sirve para crear una relación que estaría formada por cada uno de los valores del atributo de la primera relación, en combinación con cada uno de los valores del atributo de la segunda relación, de tal forma que si yo realizo el producto cartesiano del atributo «número de usuarios», con el atributo «número de fondos» de la relación de fondos de la biblioteca, lo que obtendría sería una relación que estaría formada por dos atributos, cuyos valores serían la combinación de cada uno de los usuarios, con todos los ejemplares de documentos que hay en la biblioteca; es decir, una relación que estaría formada por todos los posibles casos de préstamo que se podrían dar en una biblioteca. Evidentemente, este resultado no tiene ningún interés en la gestión de una biblioteca, y pone de manifiesto que este operador del producto cartesiano no será utilizado, en ningún caso, como tal en la gestión de una base de datos, no solo ya con información bibliotecaria, sino para cualquier tipo de base de datos, puesto que, en realidad, el producto cartesiano ha sido definido como operador por razones de tipo metodológico; es decir, que se le considera un operador llamado primitivo, que entrará a formar parte de operadores derivados de él que se componen de varias fases, una de las cuales es el producto cartesiano, como veremos más adelante.

Una vez definidos los operadores del conjunto básico del álgebra relacional, podemos definir las operaciones relacionales especiales del álgebra relacional. La primera de ellas es la restricción. Para definirla, tenemos que empezar definiendo lo que, en realidad, es una variable —la variable «theta»— que representa cualquier operador aritmético de comparación simple. Esta variable sería, por ejemplo, equivalente a «igual», «diferente» «mayor o menor que», «mayor que», «mayor o igual», «menor o igual», «menor que», etc. Todo este conjunto de operadores de comparación equivalen a esta variable que definimos como theta. Esto nos permitirá definir no sólo la restricción, sino también algunos de los otros operadores especiales.

El operador de restricción se puede definir como un operador que evalúa la verdad o falsedad de una comparación y, en función de esa evaluación, elabora una nueva relación. Así, por ejemplo, si nosotros pretendemos elaborar una relación que esté formada por los préstamos que vencen con fecha posterior a 12-8-91, tendremos que utilizar el operador de restricción, que creará una relación que será un subconjunto de las tuplas

pertenecientes a la relación de circulación o de préstamo. De esta manera la variable theta, actuando como operador, produce un subconjunto de la relación dada en el momento en que se eligen únicamente aquellas tuplas que satisfacen la condición establecida por el operador theta.

El segundo de los operadores especiales es el operador de proyección. Una proyección de una relación dada es otra relación que está formada por una parte de los atributos de la primera relación. En este sentido, por ejemplo, si yo quisiera una información de los nombres de los proveedores y las ciudades donde radican estos proveedores, lo que tendría que hacer es una proyección de la relación de proveedores con estos dos atributos.

Como se puede comprender, la combinación de los operadores de proyección y restricción es una tarea que se realiza constantemente en los sistemas relacionales. En realidad, sería tanto como decir que en las consultas de información se establecen dos condiciones básicas: el tipo de información que se pretende obtener y las condiciones según las cuales, se va a obtener esa información.

Quizá uno de los operadores más útiles del conjunto de los operadores del álgebra relacional es el llamado join, que tiene distintas versiones, de las cuales el join natural, sería, quizás, el más usado. Este tipo de join se utiliza para operar con relaciones que tienen atributos comunes. Así, por ejemplo, si cogemos las relaciones de usuarios y la de préstamos, encontraremos que el número de usuarios es un atributo común a ambas relaciones. Mediante este operador podríamos obtener una relación a partir de estas dos, que estaría formada por aquellos atributos que pertenecen a ambas relaciones y que, tupla a tupla, son combinados a través de los atributos comunes.

Por ejemplo, a partir de las dos relaciones mencionadas podríamos elaborar una relación de usuarios con sus préstamos en vigor que contuviera, por supuesto, los atributos de la relación de usuarios y los atributos de la relación de préstamos.

Evidentemente, si no existen atributos comunes entre ambas relaciones, la operación se convierte, simplemente, en una operación de producto cartesiano.

Pero, todavía más frecuente y eficaz que el join básico o natural que acabo de describir, es el join theta. Este tipo de join es similar al anterior, pero con el establecimiento de una condición. En realidad, es una operación doble: por una parte, tiene un producto cartesiano previo y, a continuación, una restricción de esa relación resultante del producto cartesiano. Así, por ejemplo, si utilizando la operación descrita anteriormente, añadiéramos la restricción de listas de usuarios con préstamos en vigor, lo que habríamos generado es una operación de join theta, es decir, un producto cartesiano de las relaciones de usuarios y préstamos y, a continuación, una proyección de una parte de la relación resultante, en base a la condición de que los préstamos estén en vigor, o sea, que los documentos no hayan sido devueltos; lo que se podría comprobar, simplemente, evaluando el contenido del atributo «fecha de devolución».

Este operador sirve, además, para ilustrar un concepto que ha tenido mucha importancia en el desarrollo del álgebra relacional, que es el concepto de operadores primitivos y operadores derivados. Cuando Codd definió los primeros ocho operadores —a los que luego se han sumado muchos más, algunos de ellos añadidos por el propio Codd y los demás por otros autores— estableció que había una serie de operadores que eran básicos, y que representaban las operaciones más simples; pero había otros —como era el caso de este join theta— que eran el resultado de la combinación de algunos de los operadores primitivos y esto hacía prever que, con el tiempo, aparecerían nuevos operadores, como así ha sido. En este momento, la relación de operadores derivados, que forman parte del álgebra relacional es muy extensa, aunque aquí sólo nos vamos a referir a algunos de ellos.

El último de los llamados operadores especiales es la división. Es una operación, según la cual se genera una relación a partir de dos, una de las cuales actúa como dividendo y otra como divisor, estando contenidos en esta nueva relación los valores de los atributos del dividendo, con los que se relacionan los valores del divisor. Así, por ejemplo, si yo establezco como dividendo la relación formada por los usuarios y los fondos prestados a estos usuarios y divido eso por la relación formada por los fondos de matemáticas seleccionados previamente, obtendría una relación formada por los usuarios a los que se ha prestado esos fondos en concreto.

Esta operación de la división es muy frecuente, sobre todo cuando se pretende, a partir de unas operaciones resultantes, obtener las relaciones que les han dado lugar.

Por último, para terminar con esta parte dedicada al álgebra relacional, hablaré de algunos operadores, como ejemplos que han sido desarrollados con posterioridad a la aparición de los ocho básicos.

Han sido muchos los autores [Codd 90, Klug 82, Lacroix 76, etc.] que, durante los últimos años, han estudiado la posibilidad de incrementar la nómina de tipos de operadores en los sistemas de gestión de bases de datos relacionales. El procedimiento que han seguido, normalmente, siempre ha sido muy similar: partiendo de los operadores primitivos han desarrollado nuevas combinaciones de estos, que dan lugar a operadores nuevos. El propio Codd al definir el modelo relacional versión 2 (RM/V2) [Codd 90] habla de muchos de estos nuevos operadores aunque no hayan sido definidos por él. Recoge los que le han parecido de más interés para

el desarrollo del modelo. Nosotros aquí vamos a hablar de algunos de estos operadores, especialmente los que considero que pueden tener cierta importancia, de cara a la gestión de información del contexto bibliotecario.

El primero de ellos podría ser sumarizar que, en realidad, no es un operador, sino un conjunto de operadores. Son todos aquellos operadores que realizan funciones, tales como: contar, sumar, calcular medias o calcular máximos o mínimos y algunos otros. Como se verá, son todas ellas operaciones de tipo aritmético, que afectan, la mayor parte de las veces, a los valores de un atributo o una parte del atributo.

El funcionamiento de estos operadores siempre es el mismo: dado un atributo o varios, o dada una relación o varias, se realiza una operación de las definidas, de tal forma que se genera una relación nueva que, en algunos casos, sólo tendrá un atributo y un valor —como pueda ser en el caso de contar, suma, media, máximo y mínimo—, pero puede haber otras operaciones que permitan la generación de relaciones con más valores.

En el caso de la información que estamos tratando se podrían poner algunos ejemplos, en los que se demuestra la utilidad de estos operadores. Así, el operador contar tiene infinidad de utilidades, sobre todo cuando se pretende una respuesta numérica a una demanda de información: si un bibliotecario, por ejemplo, quisiera averiguar el número de revistas de frecuencia semanal que existe en la base de datos de su sistema, tendría que utilizar este operador, al que habría que añadir la restricción correspondiente.

Si se opera, por ejemplo, con un determinado atributo de la relación de proveedores, para tratar de averiguar el total que suman las cantidades comprometidas con los distintos proveedores, habría que utilizar el operador sumar, también con la restricción correspondiente, aunque en este caso sería menos necesario, pues se supone que cuando no existe cantidad comprometida su valor sería cero y por tanto no importa que se sume ya que no altera el resultado final.

En cuanto a la media, tiene mucha utilidad para el funcionamiento del sistema, por ejemplo, a la hora de determinar los precios medios de los documentos pedidos, lo que permite evaluar con mayor exactitud el precio del documento antes de tramitarse su pedido.

En cuanto al máximo y al mínimo, por ejemplo, el mínimo se podría utilizar para determinar cuál es el presupuesto más bajo o la partida presupuestaria más baja de la biblioteca.

En fin, como se verá, hay muchos casos en los que este tipo de operadores resultan de gran utilidad.

Otro de estos operadores es el llamado extensión. A partir de una relación dada, crea una nueva relación que es similar a la relación original,

pero que tiene un atributo adicional, cuyos valores han sido obtenidos mediante la realización de una operación matemática con alguno de los valores de la relación original. Este operador tiene la ventaja de que permite calcular valores modificados de un atributo dado, de manera masiva; lo que nos permitiría entrar en un terreno de verdadero interés y que daría lugar, por sí solo, a otro trabajo de este tipo, como es el de la aplicación de sistemas de soporte de ayuda a las decisiones (DSS) en el mundo bibliotecario, puesto que con operadores como éste podríamos plantearnos preguntas como: ¿Qué pasaría si...?

Este operador, al margen de esta cuestión, se utiliza con frecuencia, como he dicho antes, para realizar cambios masivos en el valor de ciertos atributos. Pensemos, por ejemplo, en el caso de que los precios de un proveedor se hayan incrementado en un 10% y necesitamos cambiar los precios previstos en los pedidos en curso, realizados a ese proveedor, sumándoles un 10%. Pues el operador extensión serviría para realizar esta tarea.

Otro de los operadores que con más frecuencia aparece en los desarrollos del modelo relacional, es lo que se llama la división generalizada. Es un desarrollo de la división, que ya expliqué anteriormente y que, como se recordará, sólo permitía dividir relaciones que tuvieran una cabecera que fuera un subconjunto, en el caso del divisor, del dividendo. Por el contrario, este nuevo operador, realiza esta misma operación, pero con cualquier par de relaciones, de tal forma que produce una relación que es una mezcla de las cabeceras y cuerpos de las relaciones que se dividen entre sí. Uno de los casos en el que se puede aplicar este operador, sería el de la relación de usuarios y la relación de préstamos, en lo que se refiere, concretamente, a los atributos del nombre del usuario y el número del usuario que aparece en la primera, y el número del usuario y el fondo prestado que aparece en la segunda; de tal forma que, dividiendo ambas, crearíamos una nueva relación que estaría formada por los nombres de los usuarios y los fondos prestados a esos usuarios. Este tipo de operaciones es muy corriente en la gestión de información, no solo de las bibliotecas, sino también de otro tipo.

Existen muchos más operadores añadidos con posterioridad al modelo, pero creo que con estos ya tenemos unas interesantes muestras para usar como punto de referencia.

Para terminar esta parte dedicada al álgebra relacional, recordar que, además de operadores, el álgebra relacional estaba formada también por los llamados asignadores. Evidentemente, sobre los asignadores no hay mucho que decir. Su papel consiste, como dije con anterioridad, en cargar con un valor determinado a un elemento de la estructura de datos determinada, de tal forma que, si pretendemos cambiar el valor de un atri-

buto en una tupla concreta, tendríamos que utilizar uno de estos asignadores.

Aunque se utilizan con mucha frecuencia, su funcionamiento es bastante homogéneo y no hay mucho que decir de ellos. Siempre se suele caer en la tentación de utilizar los asignadores como sustitutos de algunos operadores, en casos en los que no quiero entrar aquí. Simplemente, recordar que su papel es el que he descrito inicialmente y no deben ser utilizados para suplir otras funciones.

IV.D. El modelo relacional y la gestión distribuida de datos

Una de las aportaciones más recientes del modelo relacional ha sido la definición de las reglas necesarias para el funcionamiento de bases de datos distribuidas, de acuerdo con la estructura relacional. Una base de datos se gestiona de forma distribuida cuando sus contenidos se encuentran en diferentes sistemas informáticos, localizados físicamente en diferentes lugares y, a pesar de eso, la base de datos se puede gestionar como una sola. En cierta manera, se puede decir que un sistema de gestión de bases de datos distribuidas, por una parte, estaría formado por datos dispersos en dos o más lugares físicos y estos lugares deberían estar relacionados entre sí por algún sistema de comunicaciones, sea del tipo que sea [Ceri 84].

En cada una de esas localizaciones físicas diferentes, los usuarios deberán poder trabajar contra la totalidad de los datos, como si de una sola base de datos se tratara, sin que en ningún momento, tengan que percibir la existencia de un sistema distribuido, o, lo que es lo mismo, la gestión de la base de datos distribuida será transparente para esos usuarios.

Todos los datos que estén radicados en una de esas localizaciones, aunque participen de la base de datos colectiva, en cualquier momento, podrán ser manejados exactamente de la misma forma, como si se tratara de una base de datos aislada del resto. Esto, con el fin de poder permitir que la gestión de los llamados datos locales que, en términos de consumo de recursos CPU son más baratos de gestionar que el resto de los que componen la base de datos distribuida, siempre puedan ser gestionados de manera autónoma, con independencia de que en cualquier momento puedan aparecer como pertenecientes al conjunto de la globalidad de los datos distribuidos.

Estas condiciones establecidas para fijar el marco general de actuación de los sistemas de gestión de bases de datos distribuidas nos permiten aproximar una definición general de sistema distribuido, de tal modo que, desde la perspectiva de una aplicación, el sistema distribuido debe ser capaz de permitir la operación transparente de datos que están radicados en una gran variedad de bases, incluso, que pueden estar gestionados por diferentes sistemas de gestión de bases de datos relacionales, que, a su vez, se ejecutan en distintos tipos de máquinas, las cuales pueden soportar diferentes sistemas operativos y que, a su vez, están conectadas entre sí a través de una gran variedad de sistemas de comunicaciones.

Lo más importante es que toda esta variedad implica, desde un punto de vista informático, una gestión también muy variada de los distintos recursos; pero este fenómeno de diversidad en la gestión es absolutamente transparente para el usuario, que no debe percibir otra cosa, sino que se encuentra trabajando contra una máquina, un sistema de gestión de base de datos y una base de datos.

En este momento de la exposición ya podemos anticipar que, la base de datos distribuida se debe concebir como una base de datos virtual, puesto que detrás de esta base de datos virtual, lo que encontraremos, será un conjunto de bases de datos reales, localizadas en distintos lugares.

El sistema de gestión de bases de datos distribuidas, entonces, resultará ser un conjunto de herramientas software que, añadidas al DBMS nos permitan realizar una gestión tal, que el usuario perciba la existencia de esa base de datos virtual. Como se podrá comprender, la ventaja fundamental de este tipo de sistemas combina eficiencia en el procesamiento de los datos, con un importante incremento de la accesibilidad, y todo ello, sin poner en juego recursos diferentes de los que el propio modelo relacional prevé.

Pero la dificultad fundamental, en relación con la gestión de bases de datos distribuidas, radica precisamente en lo que podemos llamar la implementación de este modelo de gestión distribuida en los sistemas concretos. Para explicar esto quizá convenga recurrir a la relación de 12 reglas que Date extrajo de Codd, con el fin de establecer los requisitos que debía cumplir un sistema de gestión de base de datos distribuida para ajustarse al modelo relacional. Junto a los enunciados mínimos de estas reglas, iré poniendo ejemplos de algunas de las implicaciones que el cumplimiento de esta regla tendría en la aplicación de un sistema de estas características en los entornos bibliotecarios.

Antes de empezar con estas reglas, recordaré lo que se denomina el principio fundamental de la gestión distribuida de bases de datos, que no hace sino resumir, de una manera muy gráfica, algunas de las ideas expuestas anteriormente.

Un sistema distribuido debe permitir al usuario ver las bases de datos como si éstas no fueran distribuidas.

Dicho esto, podemos pasar a enumerar las doce reglas [Date 90f]:

- La primera es la de la *autonomía local*, por la que las distintas localizaciones de un sistema distribuido deben ser autónomas. Esta autonomía local significa que toda operación realizada en un lugar concreto, será controlada por el sistema local. Esto nos lleva a plantearnos la necesidad de que el conjunto del sistema, en principio, nos debe permitir trabajar contra los llamados datos locales de igual forma que lo haría con el conjunto del sistema, pero con la diferencia de que, para trabajar con el conjunto del sistema necesita acceder a recursos remotos, mientras que para trabajar con datos locales, sólo utilizará recursos locales. A esto justamente se le llamará autonomía local. En relación con lo anterior y de cara a los entornos bibliotecarios, esta cuestión plantearía algún problema. Puesto que la consideración de recurso local o información local en un sistema de gestión bibliotecaria distribuida, debería ser precisada de antemano y, al menos para mí, en este momento, es algo que no está del todo claro. Pondré un ejemplo: la lista de autoridades contra la que se debe hacer cualquier catalogación, en un sistema distribuido o local, de gestión de catálogos, ¿es un recurso distribuido o es un recurso local? O lo que es lo mismo, ¿esa lista de autoridades debe existir completa en cada una de las localizaciones o se encuentra distribuida entre las distintas localizaciones? Cualquiera de las dos soluciones plantea, como veremos después, una serie de problemas, pero, a este nivel de análisis de la cuestión, lo único que debemos decir es qué datos del sistema de autoridades van a ser considerados como informaciones locales y cuáles serán consideradas como informaciones distribuidas y esta cuestión es especialmente importante en el caso de aquellos recursos informativos críticos que, en un sistema bibliotecario, se utilizan con mucha frecuencia y que no pueden ser entendidos como datos de cada una de las localidades, sino del conjunto del sistema.
- La segunda regla habla de la independencia necesaria de cada una de las localizaciones, respecto de un sistema central. En realidad, esto es una consecuencia directa de la regla anterior, puesto que el sistema distribuido exige autonomía local, esta autonomía sólo es posible en el caso de que los sistemas locales sean capaces de trabajar sin necesidad de recurrir a un sistema central. Desde un punto de vista exclusivamente de autonomía informativa, este objetivo solamente se puede cumplir, en el caso de un sistema bibliotecario, si las informaciones necesarias para realizar las operaciones normales —tales como actualización de información, consultas de la información a nivel local, modificaciones de la información local, etc.— se pudieran controlar sin necesidad de utilizar recursos externos, y eso incluye un sistema centralizado.

- Esta segunda regla nos permite despejar una incógnita planteada con el ejemplo de la primera regla y es que de entre las dos posibles alternativas para la realización de las catalogaciones, en lo que se refiere a la lista de autoridades, tenemos que desechar la opción de trabajar contra una sola lista de autoridades radicada en un ordenador central. Eso nos hace pensar que la solución de la lista distribuida entre las distintas localizaciones, por la misma razón también sería inviable, ya que si cada una de las localizaciones no es autónoma cuando recurre a un ordenador central, tampoco lo es cuando tiene que recurrir al conjunto de la base de datos, cada vez que hace una actualización del catálogo. Esta cuestión puede ser discutida, sobre todo si se tiene en cuenta que un sistema que dispusiera de un número reducido de localizaciones, podría no plantear excesivo problema que la lista de autoridades estuviera distribuida entre las distintas localizaciones. Ahora bien, me parece más dudoso que esto se pudiera resolver en el caso de que el sistema tuviera muchas localizaciones. Téngase en cuenta que cada vez que se va a hacer una catalogación ésta se realizará, en lo que a control de autoridades se refiere, contra la totalidad de la lista de autoridades, para lo cual, si la lista estuviera distribuida entre distintas localizaciones, se exigiría la aportación de recursos procedentes de todas las localizaciones, puesto que, en el momento que hubiera una localización no disponible en el sistema, la parte de la lista de autoridades que esa localización gestiona, no estaría disponible para los demás. La vulnerabilidad, por tanto, de un sistema en el que cada localización depende tanto de las demás, sería muy grande y esto es, precisamente, lo que pretende evitar esta segunda regla.
- La tercera regla es la llamada regla de la operación continua. Indica que un sistema distribuido no debe exigir, en ningún caso, un cierre del conjunto del sistema para realizar alguna operación concreta. Esta regla tiene gran interés para los desarrolladores de sistemas distribuidos, no tanto para los usuarios. Es una especificación de desarrollo de estos sistemas. Con esta regla lo que se pretende, básicamente, es evitar lo que ocurre en muchos sistemas informáticos en la actualidad y es que el reconocimiento de los recursos disponibles por parte del sistema se realiza en tiempo de arranque, de tal forma que si se añade un nuevo recurso al conjunto del sistema, solamente podrá ser reconocido en un arranque posterior, lo que quiere decir que ese sistema nunca será de funcionamiento continuo, puesto que se exigen cierres periódicos del mismo, cada vez que se hace una mo-

dificación de los recursos disponibles. Un sistema distribuido, por definición, debe ser un sistema de operación o funcionamiento continuo. Lo que quiere decir que debe ser posible incorporar nuevos recursos, nuevas localizaciones, nuevas relaciones, nuevas aplicaciones, etc. al sistema distribuido en el momento en que está funcionando, sin necesidad de cerrarlo.

- La cuarta regla es la de la independencia de las localizaciones. Según esta regla, los usuarios no tendrían por qué saber dónde se encuentran físicamente almacenados los datos que manejan. Es decir, la base de datos virtual se convierte, de esta forma, en una base de datos que está siendo manejada por cada usuario como si fuera la base de datos local. De acuerdo con este planteamiento, no existe ninguna dificultad en la gestión de una base de datos bibliográfica o el conjunto de la información gestionada por una serie de bibliotecas, como si de una base de datos distribuida se tratara. Así, por ejemplo, sería perfectamente posible, según esta regla, solicitar una información relativa al título de un documento y que la respuesta a esa demanda de información procediera de distintas localizaciones físicas, sin que el usuario percibiera este hecho. Pero me interesa especialmente recordar aquí que los sistemas distribuidos asumen, no tanto con el contenido de esta regla, sino más bien con el de otras, la diferencia práctica que existe entre gestionar información local y remota. Es decir, que la regla de la independencia de localizaciones que, como algunos han dicho, debería denominarse como de la transparencia de las localizaciones, es una regla que establece un objetivo de gestión de información, pero para aquellos casos en que no sea posible establecer un tratamiento local de la información porque esté radicada en una localización diferente.
- La quinta regla es la de la independencia de fragmentación. Un sistema distribuido debe ser capaz de soportar lo que se llama la fragmentación de datos, o, lo que es lo mismo, que las relaciones puedan estar divididas en distintas partes, cada una de las cuales se pueda encontrar en una localización diferente. Y entiéndase por parte aquí, tanto el hecho de que una porción de las tuplas de la relación se encuentren en una localización y el resto en otra u otras localizaciones, como el hecho de que una parte de los atributos de una relación se encuentre en una localización y, el resto en otras diferentes. Como consecuencia de esto, los usuarios en todo momento deberán ver dichas relaciones fragmentadas como si de una relación normal se tratara y, por consiguiente, no percibirán en ningún momento la fragmentación de las mismas. Esta característica permitiría —en el caso

de los sistemas que la soportaran—la gestión de un fichero de autoridades fragmentado entre distintas localizaciones.

- La regla de la independencia o transparencia de replicación consiste en que el sistema, si así es definido por el usuario, replicará automáticamente, es decir, realizará una copia de unos datos concretos en las distintas localizaciones que se hayan indicado, con el fin de mantener permanentemente actualizadas copias idénticas de cierto tipo de datos en todas esas localizaciones. La razón de que se utilice este procedimiento que, en principio parece atentar contra algunas de las normas de integridad de los datos en los sistemas relacionales, sin embargo, lo que se pretende es mejorar las prestaciones de los sistemas para evitar accesos remotos, cuando estos accesos a determinado tipo de datos son muy frecuentes; al mismo tiempo que se mejora la prestación del sistema, se mejora, también, la accesibilidad de los datos. El principal problema que tienen los sistemas relacionales a la hora de soportar esta característica, es lo que se ha dado en llamar el problema de la propagación de las actualizaciones. Si la información que pretende ser replicada, tiene que serlo en muchas localizaciones distintas y es una información que cambia con mucha frecuencia, esas replicaciones serán muy frecuentes y suponen un consumo de recursos importante. Hacer esto de manera que sea absolutamente transparente para el usuario no es nada fácil. En cualquier caso, esta podría ser una solución alternativa al problema del control de autoridades en un sistema distribuido, puesto que la existencia de copias o réplicas de la lista de autoridades común, en todas las localizaciones, mejoraría considerablemente la performance del sistema y, al mismo tiempo, se podría conseguir mantener permanentemente actualizadas las relaciones de autoridades, de tal manera que, cada vez que se hiciera una modificación de dichas relaciones en una localización concreta, automáticamente esta actualización se repercutiría en las copias de las relaciones de autoridades existentes en el resto de las localizaciones. Claro que, como planteaba antes, la propagación de estas actualizaciones, puede suponer un costo importante de recursos en un sistema complejo.
- La séptima regla es la del procesamiento de consultas distribuido. Esta regla pretende poner de manifiesto el hecho de que en un sistema relacional distribuido la interrogación a la base de datos virtual que hace un usuario puede ser respondida, en términos de procesamientos de datos, de muy diversas maneras por el sistema. Sin embargo, como todos los procedimientos empleados a la hora de utilizar el sistema darían la misma respuesta, pero no consumirían los mismos recursos, es absolutamente necesario optimizar el funciona-

miento del sistema para que las demandas de información planteadas por los usuarios contra el conjunto de la base de datos virtual consuman la menor cantidad de recursos posibles. Así, por ejemplo, a una base de datos distribuida, un usuario, desde una localización concreta, lanza la pregunta de cuáles son los documentos de tema informático publicados desde el año 90 hasta hoy, esta pregunta, probablemente, va a exigir al sistema distribuido buscar información en cada una de las localizaciones existentes y remitir dicha información a la del usuario. Pero el trasiego de información que en el sistema de comunicaciones producirá esta interrogación, será muy diferente en función de cómo el propio sistema distribuido haga circular la información de unas localizaciones a otras. Y también, en función de cómo el sistema de comunicaciones haya sido establecido. En cualquier caso, será necesario hacer un correcto análisis de una cosa y otra para optimizar el funcionamiento del conjunto en orden a lograr los tiempos de respuesta adecuados.

- La octava regla es la de la gestión de las transacciones distribuidas. Esta regla hace referencia a la necesidad de que el sistema distribuido realice, a nivel local, un control, lo más estricto posible, de la ejecución de cada una de las transacciones en base a la relación o interdependencia que cada uno de los subprocesos que estas transacciones requieran. Esto, unido al hecho de que el sistema deberá controlar también la concurrencia de dichas transacciones, de tal forma que unas no se opongan en su ejecución a las otras. Con ello, en definitiva, lo que se pretende es establecer unas reglas de funcionamiento mínimas que el sistema necesita, para que cada una de las peticiones lanzadas por los usuarios no se interfieran con las otras.
- Por lo que afecta a las cuatro últimas reglas, se pueden describir las cuatro juntas. Hacen referencia, respectivamente, a la independencia del hardware, del sistema operativo, del soporte de red y del DBMS específico. En resumen, estas cuatro reglas, lo que pretenden poner de manifiesto es que el sistema distribuido no debe ser, en modo alguno, dependiente ni del hardware ni del software básico en el que funcionan. O lo que es lo mismo, que el sistema distribuido debe poder funcionar, tanto si los sistemas informáticos que lo componen son del mismo fabricante, como si son de distinto. Y lo mismo por lo que afecta a los sistemas de comunicaciones. En principio, un sistema de gestión distribuido debe ser capaz de lanzar demandas desde cada una de las localizaciones a todas aquellas donde existan sistemas de gestión de distribución de datos diferentes, porque todos utilizarían los mismos estándar para el intercambio de peticiones de información.

Para concluir este apartado dedicado a los sistemas distribuidos diré que, como una de las razones que han ocasionado el desarrollo del modelo relacional, siempre ha sido el interés en proteger las inversiones realizadas en el desarrollo de aplicaciones, de tal manera que si estas aplicaciones estaban realizadas de acuerdo con un modelo que se mantuviera invariable como esquema conceptual de desarrollo a lo largo del tiempo, las aplicaciones desarrolladas de acuerdo con este modelo, por muchos cambios que hubiera en los sistemas de desarrollo, seguirían siendo igualmente útiles. Por lo que se refiere a los sistemas distribuidos, este objetivo general ha permanecido invariable. Como se verá, muchas de las reglas hacen referencia, o bien a la independencia de los datos respecto de las aplicaciones, o a la independencia del hardware y software básico, respecto de las aplicaciones. Lo que, en definitiva, significa que las aplicaciones son igualmente válidas, por mucho que se modifiquen los sistemas de gestión, porque, en cualquier caso, serán independientes de los datos e independientes de los soportes.

IV.E. Los lenguajes relacionales

Prácticamente, de forma simultánea al desarrollo del modelo relacional se planteó la necesidad de que existieran unos lenguajes de programación que permitieran la gestión de datos contenidos en estructuras relacionales. Desde el primer momento esto exigió a los desarrolladores del modelo la definición de una serie de principios de diseño de lo que se ha dado en llamar lenguajes relacionales.

Pero hay que aclarar que los llamados lenguajes relacionales no son en realidad lenguajes de programación en sentido estricto, sino que son lo que se llama sublenguajes, es decir, son conjuntos de comandos, funciones, operadores, etc., que están orientados al manejo de datos en estructuras relacionales, pero que no forman por ellos mismos el conjunto de comandos, funciones, operadores, etc. necesarios para convertirse en un lenguaje de propósitos generales. Todas aquellas instrucciones que aparecen en los lenguajes de programación y que no tienen nada que ver con la gestión de datos o con su manipulación, en un lenguaje relacional no existe. Esto tiene una implicación directa, para desarrollar una aplicación utilizando un sistema relacional es necesario desarrollar esa aplicación utilizando un lenguaje que permita la inclusión de instrucciones del lenguaje relacional, porque esas instrucciones por sí solas, nunca podrán formar una aplicación. Por ejemplo, no existen instrucciones para el manejo de una pantalla ni de otros dispositivos, sino, simplemente, instrucciones para la gestión de los datos [Date 89].

Esta forma que estoy describiendo de utilización del lenguaje relacional no es la única. Esta fórmula de utilización se llama lenguaje relacional incluido, porque las sentencias del lenguaje relacional se incluyen dentro de un programa hecho en otro lenguaje. Sin embargo, desde el primer momento, se pensó que las instrucciones de estos lenguajes relacionales también deberían poder ser ejecutadas como instrucciones enviadas por un usuario desde un terminal, sin que llegaran a formar programas. Es decir, ejecución en modo directo. Esta modalidad de ejecución es común también a todos los lenguajes relacionales, aunque la dificultad de estos lenguajes de programación, llevó a muchos de los desarrolladores de sistemas relacionales a la creación de lenguajes completos de programación que permitieran la gestión de las estructuras relacionales por sí mismos, de tal forma que, en la actualidad, los sistemas relacionales disponen casi todos de lenguajes completos de programación, normalmente son de cuarta generación. Uno de los casos más conocidos es el llamado Quel, del sistema Ingres, que es un lenguaje de cuarta generación para la manipulación de estructuras relacionales y el desarrollo de aplicaciones completas [Date 89]. Al mismo tiempo, se fue desarrollando un lenguaje estándar para la manipulación de los datos, de acuerdo con las especificaciones que al respecto había hecho Codd en su momento [Codd 70].

Esto es lo que dio lugar a la aparición del SQL (Structured Query Languaje). Este sublenguaje se ajusta en la actualidad a una norma ISO, la norma Iso TC97/SC21/WG3 N117 y a su vez se ajusta también a la norma ANSI X3.135-1986. Esto quiere decir que con el paso del tiempo el SQL ha terminado por ser un lenguaje con pretensiones de universalizarse en su sintaxis para la gestión de las estructuras relacionales. Pero aquí quiero comentar sobre todo los aspectos funcionales más importantes que hacen referencia al SQL, puesto que éste es el único lenguaje relacional que ha llegado a tener carácter de estándar. Los otros son desarrollos particulares de quienes diseñan sistemas relacionales.

Cuando se definió el modelo relacional, como dije antes, apareció la necesidad de incluir en dicho modelo unas especificaciones sobre el lenguaje que debía manejar sus estructuras de datos, con el fin de evitar la proliferación de lenguajes que hicieran dependientes las aplicaciones. Aunque, en principio, las especificaciones fueron muy generales, sin embargo inmediatamente estas especificaciones dieron lugar —fundamentalmente por parte de IBM— a un desarrollo inicial llamado SQL, que contenía un conjunto mínimo de instrucciones que se añadían a desarrollos hechos básicamente en PL/I.

Durante los años ochenta, el desarrollo del SQL y la generalización de su uso le llevó, primero a convertirse en un estándar ANSI —1982— e inmediatamente después en un estándar ISO —1987—. En la actualidad

—y este es un aspecto que me interesa resaltar muy especialmente— no sólo es un estándar a nivel de organismos internacionales, sino que también lo es a nivel de grupos de desarrollo tan importantes como el grupo X/OPEN, que se ha convertido en el punto de referencia obligado para el desarrollo del sistema operativo Unix [X/OPEN 87].

Defender aquí la necesidad de un estándar para el manejo de estructuras de datos relacionales no resultaría demasiado costoso. Daré en cambio únicamente unas notas que pueden tener cierto interés, aunque trataré en todo momento de evitar la referencia a aquellas motivaciones que tienen más que ver con el marketing de los sistemas relacionales y procuraré centrarme en las que hacen referencia a cuestiones de tipo funcional que se relacionen con la gestión de información en un entorno bibliotecario.

La primera de ellas es que la existencia de un estándar facilita la portabilidad de las aplicaciones. Es decir, el estándar incide directamente sobre uno de los aspectos que se han considerado clave en el desarrollo del modelo relacional: La independencia de todo hardware y software básico. Es la normalización del lenguaje lo que más contribuye a hacer que las aplicaciones desarrolladas sean independientes de hardware y software, aunque el hecho de que el lenguaje no sea completo hace que las aplicaciones sean independientes, pero sólo por lo que se refiere al manejo de los datos, no al resto de la aplicación. Esto supone que estas aplicaciones sean mucho más duraderas en el tiempo. Esta cuestión ha traído de cabeza a los diseñadores del lenguaje durante muchos años, pues los tiempos de vida medios de los recursos informáticos se han ido reduciendo cada vez más y muchas veces la renovación de estos recursos traía como consecuencia la obsolescencia de las aplicaciones, de tal manera que no podían ser ejecutadas por haber desaparecido los dispositivos para los que habían sido diseñadas.

Por otra parte, la existencia de un estándar tiene un efecto importante sobre la homologación de los procedimientos de gestión, puesto que el esfuerzo de desarrollo que se hace en un lenguaje que es poco utilizado, no permite que las estructuras de programación sean mejoradas por la incidencia en los mismos procesos de muchos usuarios distintos. En este sentido, la utilización de un lenguaje estándar ha permitido que determinados procesos se hayan optimizado mucho en su funcionamiento dentro del estándar, puesto que muchos usuarios han ido incidiendo sobre el mismo problema aportando cada vez mejores soluciones.

Los comandos del lenguaje SQL se pueden dividir en una serie de apartados, como son: la definición de datos, la manipulación de los datos, y el control de los datos. Estas tres partes nos permiten agrupar estos comandos en base a sus funciones básicas. En realidad, el conjunto de ins-

trucciones que utiliza el SQL es bastante reducido, puesto que la manipulación de las estructuras relacionales es su único objetivo y, por consiguiente, no se necesita un gran número de comandos para manejar únicamente estructuras de datos. Pero, por otra parte, son unos comandos muy parametrizables, de tal forma que el mismo comando permite ejecutar funciones muy distintas únicamente cambiando sus parámetros.

Todos estos comandos que forman parte del lenguaje SQL, como ya dije, pueden ser ejecutados en modo directo o incluidos como parte de otros programas. El procedimiento de inclusión de una sentencia SQL en el código de un programa escrito en cualquier lenguaje de programación, es diferente según el lenguaje de programación de que se trate. Uno de los procedimientos más corriente es poner delante de la sentencia SQL la expresión EXEC SQL, lo que significa que lo que vienen a continuación es una sentencia SQL.

Los elementos básicos del lenguaje SQL son los típicos de cualquier lenguaje de programación. Existen caracteres que tienen un significado especial, como puedan ser signos aritméticos, paréntesis, separadores, etc. Pueden utilizarse también literales que van entrecomillados, como en todos los lenguajes de programación y existen una serie de palabras que identifican o bien los parámetros de las instrucciones, o bien las propias instrucciones. Las sentencias del SQL que reciben el nombre de anotaciones se construyen normalmente por agrupación de estos elementos.

No pondré aquí ejemplos de sentencias SQL, porque me parece que, en este momento, lo único necesario es llamar la atención sobre el hecho de que el modelo relacional tiene aparejado un lenguaje propio para la gestión de sus estructuras de datos, aspecto funcional que no encontraremos con facilidad en otros modelos. Sin embargo, es importante hacer constar que la existencia de tal lenguaje tiene implicaciones en la gestión de los datos, puesto que la mayor parte de los procesos a los que deberíamos someter la información que se gestiona en una biblioteca ya están definidos con rutinas específicas SQL para procesamiento de información, muchas de las cuales han sido publicadas. Ahora bien, en algún caso se ha especulado con la posibilidad de que el SQL fuera utilizado como lenguaje para la consulta o la gestión de información por parte de usuarios finales. Es cierto que, en su diseño, se pretendió que este lenguaje pudiera ser utilizado, indistintamente, por programadores y no programadores, pero las estructuras de datos que tendría que manejar SQL en un entorno de gestión de información bibliográfica serían muy complejas y esto necesariamente hace compleja también su gestión desde las anotaciones de este lenguaje. Existen referencias más que sobradas a propósito de la complejidad de utilización de SQL para usuarios finales; incluso, en algunas de estas referencias, se compara la simplicidad con la que se puede trabajar con un

IRS, frente a la complejidad que exige el lenguaje SQL [Macleod 91]. En este sentido, creo que es importante dejar bien sentado cuál es el ámbito de aplicación de ese lenguaje en los entornos bibliotecarios.

En mi opinión, a pesar de que el lenguaje haya sido considerado como un estándar, es poco probable que se pudiera generalizar su uso para la consulta porque su sintaxis es bastante compleja. Y en el caso de la información bibliográfica muy especialmente, ya que la estructura de datos a la que nos obligaría el modelo relacional exigiría unas anotaciones SQL para la consulta de dicha información que serían largas y complicadas. Si se consulta algunos de los documentos que he citado anteriormente se podrá comprobar esto que digo con bastante exactitud. Por otra parte, esto no obsta para que pueda utilizarse el SQL en el desarrollo de aplicaciones para entornos bibliotecarios, ya que de esa manera estas aplicaciones se beneficiarían de las ventajas de la utilización del estándar que enuncié anteriormente.

IV.F. Conclusiones

a) Resulta sobradamente evidente que los DBMS en general, y los RDBMS en particular, son sistemas de gestión de la información que se amparan en un modelo de tratamiento de datos que se basa en la estructuración de los referentes informativos reales. Las estructuras definidas deben cumplir la condición de estar perfectamente normalizadas —información escalar por atributo— con el fin de que en el sistema definido exista coherencia entre la estructura de datos definida y las reglas de integridad que la rigen.

b) Habida cuenta que la información bibliográfica es en su mayor parte textual, y que los textos son un tipo de atributo en el que las características físicas pueden variar mucho de unos a otros; operaciones complejas de tratamientos de cadenas serán necesarias para realizar la gestión de este tipo de información. Operaciones que determinarán la existencia de estructuras de acceso a dichos atributos cuyas claves deberían ser múltiples por cada atributo de cada tupla. Estas diferencias en la estructuración de la información han hecho decir a muchos autores que los RDBMS no son sistemas adecuados para la gestión de información textual, especialmente por problemas de performance de las aplicaciones desarrolladas.

c) Aunque es cierto que se han hecho en los últimos años diversas propuestas de diseño de estructuras de datos que permiten la gestión de información textual en base a sistemas RDBMS, estas propuestas en su mayor parte no cumplen con algunas de los princi-

pios de normalización informativa impuesto por el modelo. Muchas de estas propuestas se basan en la idea de que las palabras clave, que actúan como términos de indización en un IRS, se conviertan en atributos de la estructura definida en un RDBMS. Esto, en mi opinión, significa que el sistema es utilizado como soporte de desarrollo exclusivamente o, como veremos en el capítulo dedicado a los FMS, en un DDL —Data Definition Languaje—, lo que significa que esa misma estructura podría ser gestionada utilizando cualquier lenguaje de programación convencional.

d) La normalización exhaustiva a la que obliga el modelo relacional lo hace especialmente apropiado para la manipulación de la información no textual, por lo que sería una herramienta muy útil en el desarrollo de aplicaciones bibliotecarias por lo que afecta a la información no bibliográfica. Todo esto mientras no se incluyan entre las especificaciones del modelo relacional algunas destinadas al soporte de funcionalidades de recuperación de información.

e) Si se intentara realizar una representación tabular de las descripciones bibliográficas a fin de cumplir con las exigencias de normalización del modelo —primera forma normal—, el efecto inmediato al no ser atómica esta información de forma natural, es su desmembración en infinidad de tablas. Llegados a este punto el proceso de recuperación de los documentos desmembrados obliga a la realización de infinidad de operaciones de manipulación para volver a componer las referencias.

f) La complejidad de esta estructura normalizada de datos no sólo tropieza con el problema de los tiempos de respuesta para su gestión, sino que la recuperación utilizando el lenguaje relacional estándar —SQL— es muy costosa para el usuario en dos sentidos: 1) Al haber tanta diferencia entre la representación interna de los datos y la externa, el usuario debe conocer la interna con el fin de poder traducir en términos de atributos de tabla y contenidos de los mismos sus necesidades de recuperación. 2) A continuación, tendrá que especificar en el momento de la definición de la consulta los atributos que quiere visualizar y de qué forma han de relacionarse. En resumen, la utilización de SQL para la recuperación de información textual normalizada de acuerdo con el modelo relacional es muy dificultosa para el usuario y, en mi opinión, inadecuada para su utilización en OPACS.

V

LOS SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN

Aunque en mi opinión lo más característico de los sistemas de recuperación de información es la estructura de datos en que están basados, no estaría de más hacer aquí una pequeña descripción de los aspectos funcionales básicos de estos sistemas que sirva de introducción.

La concepción de un IRS —a menudo también llamados «Free Text Retrieval System» FTRS— parte del principio de que la información procesable por un sistema informático se organiza en base a documentos. La consecuencia inmediata de esta idea es que los IRS se desarrollan con el múltiple objetivo de almacenar, recuperar y mostrar grandes cantidades de documentos, entendidos éstos como secuencias más o menos extensas de caracteres que se agrupan formando palabras, que se agrupan en frases, que a su vez conforman párrafos, y que, por fin, en número variable, componen los documentos. En este punto ya podríamos sacar una conclusión de carácter general: Los IRS son sistemas informáticos capaces de procesar documentos que tengan una cierta estructura formal interna. Por otra parte, aunque de manera superficial, me parece importante recordar que cuando hablo de documentos me refiero también a cualquier tipo de representación documental que cumpla con los requisitos expuestos anteriormente.

Como se puede imaginar, para definir estos sistemas me apoyo en autores que han estudiado con profundidad el fenómeno [Salton 83, Kemp 88, Ashford 88, Salton 88]. Aunque trataré de hacer una descripción general que refiera la mayor parte de las prestaciones de estos sistemas, me detendré con posterioridad en aquellos aspectos que tienen una mayor relevancia en el tratamiento de información bibliográfica en formatos normalizados, sin olvidar que algunos de los IRS ofrecen posibilidades de procesamiento de información altamente estructurada.

Estos sistemas tienen comportamientos especialmente eficaces cuando tienen que generar estructuras de acceso a los documentos que faciliten la localización de los mismos con rapidez. La mayoría de los sistemas comerciales existentes construyen estas estructuras de acceso a partir de las palabras que forman los documentos, proceso que se denomina indiza-

ción automática mediante palabras claves. Este proceso puede variar de unos sistemas a otros, pero en la mayoría de ellos se genera un índice que asocia cada palabra con su posición exacta dentro del documento. En otros casos la información posicional no es tan precisa y se consigna sólo la frase, el párrafo o incluso sólo el documento al que pertenece la palabra en cuestión. Estas diferencias en la generación y composición de los índices ponen de manifiesto que existe una relación muy estrecha entre la estructura de acceso generada y las posibilidades reales de acceso a la información por parte de los usuarios. Por esta razón insistiré más en los aspectos estructurales en el desarrollo de este modelo.

Si como dije antes la información procesada por un IRS está compuesta por documentos completos o referencias complejas, es obligado admitir que estos documentos contienen textos en lenguaje natural, lo que abre una serie de posibilidades interesantes de gestión de las demandas de información de los usuarios que superan las expectativas de las operaciones booleanas. En principio, al usar un IRS la primera diferencia que se observa con respecto a otros sistemas es que las respuestas del IRS son siempre conjuntos de documentos, lo que facilita la gestión de operaciones —lógicas o de otro tipo— posteriores. De hecho, casi todos los sistemas de este tipo deberían permitir la utilización de operadores de proximidad, técnicas de ponderación, y búsquedas vía thesaurus. En definitiva, esta flexibilidad en la estructuración de la información, combinada con la variedad de posibilidades de acceso, induce a pensar que los IRS han sido concebidos sobre principios contrarios a los DBMS, puesto que su objetivo es facilitar información a usuarios con necesidades poco previsibles y procesar documentos que tengan estructuras poco definidas. Es más que probable que esta «indefinición» generalizada se deba a que los potenciales usuarios de estos sistemas, como se ha puesto de manifiesto recientemente [Ftrs 89], tienen intereses muy diversos:

— Suministradores de información: Centralizan en sus oficinas grandes cantidades de recursos informativos que ponen a disposición de sus clientes mediante interfaces de usuario final para la consulta. Por lo que afecta a la carga de documentos, puede ser realizada por profesionales, por lo que no son imprescindibles sistemas amigables de edición y carga de textos. No suelen requerir cambios en el diseño de las bases de datos, y son las cuestiones relativas a los tiempos de respuesta en la recuperación las que más interesan a este tipo de usuarios.
— Investigadores individuales o en grupo: Las exigencias en este caso son menores en cuanto a prestaciones. El uso fundamental es la gestión de los recursos informativos utilizados en los trabajos de investigación. En este caso priman todos los aspectos relacionados con los interfaces de usuario final, así como las posibilidades de relación del IRS con sistemas de otra naturaleza —DBMS, hojas de cálculo, procesadores de texto, etc.— para intercambiar información. Son, así mismo, importantes altas prestaciones en el apartado de impresión de productos y versatilidad en la indización.
— Especialistas en tecnología de la información: En este caso resulta más difícil explicitar las prestaciones necesarias puesto que el uso básico será la docencia y estará muy condicionada por el nivel al que se dirija.

En cualquier caso, como se observará, la gran variedad de situaciones de uso hace imprescindible que estos sistemas sean por encima de todo versátiles en su funcionamiento para poder adaptarse a las mas diversas necesidades. Esto se hace posible con la generación de índices que recogen en el momento de la carga toda la «información situacional» posible de cada uno de los elementos informativos —palabras— que contiene la representación del documento. Estas exigencias orientan favorablemente los IRS hacia la gestión de información textual y los hacen menos adecuados para gestionar información numérica o gráfica.

En resumen, los aspectos funcionales básicos de un IRS son los siguientes [Ashford 84]:

- Diseño y modificación de las estructuras de los documentos que formarán la base de datos.
- Recuperación de conjuntos de documentos.
- Salida formateada de documentos recuperados.
- Control de los términos indizados.
- Mantenimiento de los documentos en la base de datos.
- Sistemas de recuperación y auto arranque. Tolerancia a los fallos.
- Gestión de tipos de usuarios mediante niveles de autorización de los mismos.
- Monitorización del sistema.

V.A. La estructura de datos

La generación del sistema de índices que resulta necesario para la recuperación de la información en un IRS es un proceso complejo que se realiza mediante diversas fases que a continuación describiré y que genéricamente se denomina proceso de carga.

El comienzo de la operación exige la definición previa de una estructura de base de datos que estará formada por la información relativa a la estructura interna de los documentos, así como a las características textuales de cada elemento de esa estructura, especialmente por lo que afecta a aquellos datos que son imprescindibles para la generación de las distintas entradas en los índices. En definitiva, el proceso de carga de los documentos es un proceso automático mediante el que se enfrenta cada documento con la información de diseño para generar la estructura de acceso que forma una serie de índices junto con el documento completo.

Algunos de los datos contenidos en la información de diseño resultan vitales para la realización de la primera fase de carga: la composición de posibles párrafos que forman los documentos y las pautas para reconocer los comienzos de los mismos, las opciones de indización cuando alguno de los elementos de la estructura documental requiere un proceso de indización diferente del convencional, y, por último, una posible lista de «palabras vacías» que, junto con la relación de caracteres separadores de palabras que ya tiene el sistema, le permitan identificar en el texto los términos de indización.

En esta primera fase el sistema lee todas las palabras del documento y almacena de forma temporal cada una de las que considera claves junto con el número del documento al que pertenece, el código del párrafo dentro del documento, el número de frase dentro del párrafo y número de orden del término dentro de la frase. Estos cuatro datos por término de indización son la base utilizada por el sistema para facilitar todo tipo de recuperaciones. Al mismo tiempo cada nuevo documento se almacena en un fichero de texto, mientras que la posición de comienzo de ese documento junto con su número de orden se anota en un fichero de índice de textos.

La segunda fase comienza con la ordenación alfabética del fichero temporal generado en la fase anterior, de tal forma que los términos repetidos aparecerán juntos. A partir de esta información ordenada se puede actualizar el fichero diccionario que actúa como parte de un fichero invertido real y contiene todos los términos de indización diferentes aparecidos en los distintos documentos, junto con el número de documentos en que aparece y el total de apariciones.

Una vez generado el fichero diccionario se completa la información del invertido con otro fichero que contendrá las informaciones posicionales de cada una de las claves del diccionario, al tiempo que se añade una referencia en el diccionario para poder recuperar desde éste rápidamente las informaciones posicionales.

La descripción de estas tres fases del proceso de carga nos permiten vislumbrar la estructura de datos que está «detrás» de cualquier demanda

informativa dirigida a un IRS. Aunque puede haber pequeñas diferencias entre unos sistemas y otros, básicamente los elementos descritos hasta aquí son comunes. La constatación de que la información contenida en la estructura de acceso a la información en los IRS es posicional y de frecuencias se realiza con facilidad estudiando algunos de los manuales de referencia de los sistemas comerciales mas usados [Stairs, Brs].

Aunque no es mi intención reabrir la vieja polémica sobre la pertinencia del uso de estos sistemas en los entornos bibliotecarios, sí quisiera recordar que han sido muy numerosos los autores que desde hace tiempo se han venido manifestando partidarios del uso de este tipo de estructuras de datos para la gestión de información bibliográfica, muy especialmente en los entornos bibliotecarios [Prywes 72, Warheit 69]. La prueba más palpable de que ésta es una idea muy extendida es la gran cantidad de sistemas de esta naturaleza que están operativos en el mundo bibliotecario para la gestión de los OPAC's. Sistemas mundialmente conocidos como Gladis, Melvyl, Epic, Rlin, Carl, Notis, etc. utilizan estructuras de datos similares a la que acabo de describir, de tal forma que se puede afirmar sin miedo al error que algunas de las más grandes bases de datos catalográficas del mundo son accesibles a través de sistemas de la familia de los IRS.

V.B. Análisis funcional

Una vez descrita a grandes rasgos la estructura de datos que sustenta un IRS ha llegado el momento de entrar en detalles sobre las capacidades operativas que permite esta estructura, centrándome específicamente en los aspectos que más relevancia puedan tener de cara al tratamiento de información bibliográfica y de información altamente estructurada, con el fin de poder ir evaluando las posibilidades de aplicación que tiene un sistema de esta naturaleza en la gestión automatizada de la información bibliotecaria.

Para analizar funcionalmente estas aplicaciones de manera sistemática he dividido todas las funciones en siete bloques que iré describiendo seguidamente.

V.B.1. *Estructura de la base de datos*

Cualquier base de datos debe disponer de un diccionario de datos que el sistema mantiene y que permite el acceso por parte de los usuarios a toda la información relacionada con las características de los documentos que se pretende almacenar en la base. La manipulación de estos diccio-

narios de datos es vital para realizar las tareas de mantenimiento de la base. Aunque en el caso de la información catalográfica no ocurre con mucha frecuencia que sea necesario hacer modificaciones en la definición de los datos, cuando se produce una actualización de los formatos MARC estas modificaciones son imprescindibles, por lo que una buena gestión de los diccionarios de datos facilitará este tipo de tareas, que en la actualidad se están haciendo tan habituales. Estas operaciones de administración de la base incluyen las modificaciones de las características de indización de los documentos y, en general, todas las opciones asociadas a los distintos elementos de la estructura de los documentos que afecten a las operaciones de carga de los mismos.

Es indudable que la diversidad de tipos de datos gestionados por el sistema condiciona de manera definitiva las posibilidades del gestor para procesar ciertos tipos de información. Esto resulta esencial en el caso de que tratemos de procesar informaciones que incluyen datos de diversa naturaleza, números de distintos tipos, fechas, datos lógicos, texto con o sin formato, etc.

Muy ligado con el tema de los tipos de datos está el de la indización, que debe ser también variada para poder responder a la diversidad de los datos almacenados. Pero, como vimos en el apartado anterior, la combinación del fichero diccionario e invertido facilita las operaciones de recuperación mediante operadores booleanos y de proximidad [Heaps 78]. Ésta debe ser considerada la indización base que se realiza en todos los párrafos del documento a menos que el usuario especifique opciones diferentes para determinados párrafos. Por ejemplo, en los documentos MARC los campos de notas —5XX— no necesitan ser indizados y por tanto el administrador de la base puede anular la indización de esos párrafos, con lo que se consigue una reducción del tamaño del diccionario e invertido, así como del tiempo de carga.

La composición de campos y subcampos, repetibles o no, de los formatos MARC puede plantear algunas dificultades cuando esta estructura debe ser gestionada por un IRS. En principio estos sistemas no deben tener dificultad para trabajar con documentos que tengan una estructura jerárquica y cuyos elementos puedan repetirse un número indeterminado de veces, lo que facilitaría la gestión de los formatos MARC. Con lo que se puede plantear alguna dificultad es con la convivencia en los mismos campos de información codificada y no codificada, especialmente de cara a la determinación de los elementos informativos que deben ser indizados y cuáles no. En definitiva, con la típica estructura jerárquica formada por Base de datos, documentos, párrafos, subpárrafos, frases y palabras es perfectamente posible encajar los formatos MARC. A partir de aquí sólo bastará indizar los contenidos textuales de los registros separándolos de las informaciones codificadas que no deben ser indizadas.

Cuando la información que va a ser introducida en la base se ajusta a normas estrictas —Reglas de Catalogación, Control de Autoridades, Normas MARC, etc.— es preciso que el sistema disponga de procedimientos de verificación de la entrada de datos y cotejos posteriores a la entrada y previos a la carga con el fin de garantizar la homogeneidad de la información y la calidad de las recuperaciones. Algunos de estos procedimientos de verificación son los siguientes:

- Ajuste a un modelo
- Valor sólo alfabético
- Valor sólo numérico
- Valores máximos y/o mínimos
- Control de longitud
- Control de formato de fecha
- Control de presencia
- Conversión automática de caracteres
- Valor dependiente de otro
- Generación de formas canónicas
- Validaciones contra tablas, ficheros u otras bases
- Utilización de caracteres de ocultación...

En resumen, como se puede ver, la capacidad del sistema para estructurar la información y mantener esa estructura depende de una serie de características funcionales que de no estar presentes en la aplicación harán inviable su uso para la gestión de información bibliográfica normalizada o información de gestión estructurada.

V.B.2. *Funciones de recuperación*

Las capacidades básicas de recuperación de la información están severamente condicionadas por las posibilidades de indización del sistema. Ahora bien, ciertos aspectos formales de la recuperación, como la necesidad o no de utilización de un lenguaje de recuperación y en general todo lo relacionado con la manipulación de las búsquedas, no depende de manera tan estrecha del método de indización sino más bien de condicionamiento de diseño del propio sistema. En cualquier caso, tanto unos aspectos como los otros, relacionados con las funciones de recuperación, serán comentados aquí.

Como ya anticipé anteriormente, los índices de la estructura de datos que sustenta a un IRS permiten la realización de operaciones de búsqueda

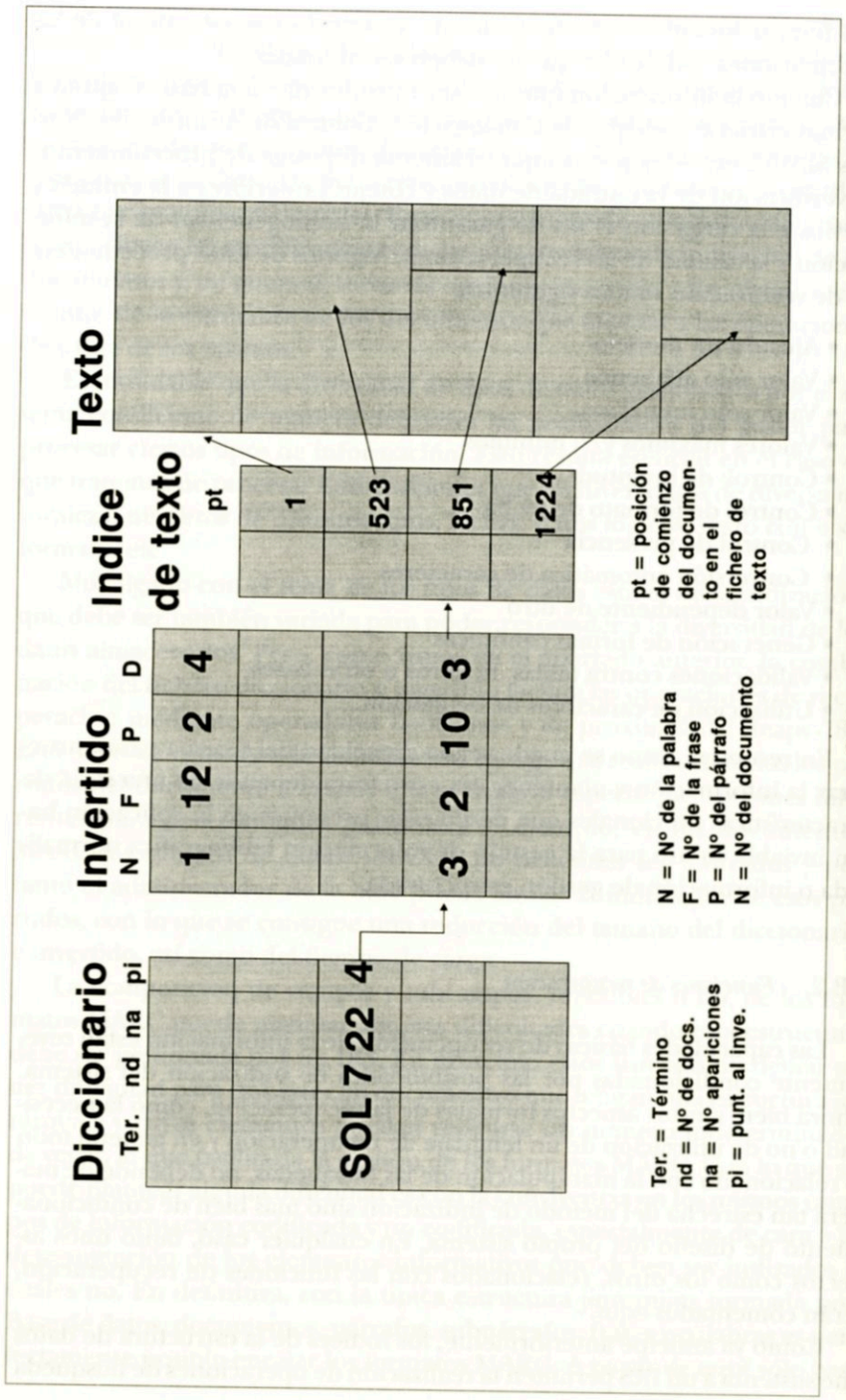


Gráfico 5

tanto del tipo booleano como de proximidad. Esto se debe a dos hechos fundamentales. Por una parte, la respuesta de un sistema con fichero diccionario siempre es, en primera instancia, el número de documentos («nd») y/o el número de apariciones («na») en esos documentos de cada término de búsqueda. Por otro lado, un sistema con fichero invertido puede generar fácilmente matrices con la información posicional relativa a cada entrada del diccionario en todas sus apariciones [Heaps 78]. Partiendo de dichas matrices es relativamente sencillo realizar operaciones como las antedichas —booleanas o de proximidad— sin que esto suponga un excesivo costo en términos de CPU.

Aunque funcionalmente la recuperación de información en un IRS se hace básicamente como acabo de describir, es necesario completar esta información con ciertos aspectos que flexibilizan las operaciones de búsqueda. Un buen ejemplo de lo que estoy comentando es la posible utilización de caracteres de enmascaramiento en sus diferentes versiones, que potencian la generación de los conjuntos de documentos. En general, podemos decir que existe toda una serie de operaciones como ésta que en esencia permiten preprocesar las entradas suministradas al sistema para su recuperación. Otro de los ejemplos más conocidos es la gestión de equivalencias fonéticas o de sinónimos, y uno de los más frecuentes es el procesador de variantes morfográficas de un término, que permite la recuperación de todos los elementos de un paradigma gramatical partiendo de uno de ellos. La dificultad fundamental de algunos de estos preprocesadores está ligada a la necesidad de su desarrollo específico para cada una de las lenguas del texto de los documentos. En cualquier caso la generación automática de variantes de género y de número a partir del término de búsqueda podría servir para mejorar sensiblemente las prestaciones de estos sistemas de recuperación, razón por la cual algunas versiones comerciales de estos sistemas ya disponen de estos preprocesadores.

Otro aspecto del problema de la recuperación desde el punto de vista funcional que me gustaría analizar es el del tratamiento de las frecuencias de aparición de los términos de indización en el diccionario. No es muy frecuente que esta información se utilice para mejorar la recuperación, pero existe algún caso en el que se usan métodos de ponderación de los documentos recuperados basados en la relación de frecuencias de aparición de los términos en cada documento y en el conjunto de la base de datos [Stairs, Sager 76]. Así mismo de manera más reciente se han comenzado a utilizar alguno de los llamados métodos probabilísticos en sistemas comerciales —«Personal Library Software», PLS—. La base teórica y los desarrollos prácticos del modelo probabilístico arrancan de la segunda mitad de los 70 y existe una numerosísima bibliografía al respecto [Robertson 77, Rijsberger 77, Rijsberger 79, Salton 88, Bookstein 85, etc.] que ha

puesto de manifiesto que es posible añadir a los IRS procesos de ponderación de la relevancia de los documentos partiendo del cálculo de la probabilidad de dicha relevancia. Aunque no es objeto de este trabajo entrar en los detalles técnicos del funcionamiento de estos procesos, sino más bien poner de manifiesto sus implicaciones funcionales desde la óptica de los usuarios, me parece oportuno comentar que los IRS, en contra de quienes sostienen que son sistemas de recuperación booleanos y de equiparación —«boolean matching» o «exact matching»—, se puede decir que cuando cuentan con procesos de ponderación probabilística o, como veremos después, de similaridad vectorial, están más cerca de sistemas de equiparación parcial de términos —«partial matching»— [Belkin 87]. Esto abre unas enormes posibilidades a estos sistemas, que son habitualmente criticados por las dificultades que plantean en la estructuración de la información [Macleod 85], pero que son, al mismo tiempo, enormemente específicos para la gestión de información textual. Por lo que sabemos desde los años cuarenta gracias a la psicolingüística y a los primeros trabajos de recuperación de información [Zipft 49, Luhn 58], existe cierta relación entre la frecuencia de utilización de los términos y la amplitud de su significado. Esta idea primigenia que está en la base de los desarrollos incorporados recientemente a algunos IRS y que posteriormente comentaré más en detalle, supone el punto de partida para la realización de las primeras investigaciones relevantes en este campo [Maron 60, Shultz 68].

Como se puede observar, la mera identificación de los términos de búsqueda no es el único ni último método de recuperación de información. Estructuras de datos que incluyen información de frecuencias de uso o de relaciones de contenido, estructuras, en definitiva, que cuentan con la información necesaria para basar la recuperación en el valor de los términos más que en su forma, son necesarias para trabajar con información textual no escalar. Pero en relación con esto es preciso señalar que los IRS son sistemas con una escasa independencia de los datos, lo contrario de lo que ocurre con los DBMS [Date 90]. De tal forma que las posibilidades funcionales en la recuperación no previstas por el propio sistema, originalmente no podrán ser implementadas sin afectar básicamente tanto a la estructura de datos como a la arquitectura de la aplicación, lo que no ocurre con un sistema de modelo de datos interpuesto del tipo RM/V2, Codasyl, etc.

Me gustaría, a continuación, tocar algunos aspectos menos básicos, pero también funcionales, relacionados con la recuperación de información en los IRS. Antes mencioné el hecho de que estos sistemas pueden disponer de un limitado conjunto de comandos con una sintaxis propia, que recibe el nombre de lenguaje de recuperación. Sin entrar en detalles sobre los aspectos formales de esta cuestión sí que me gustaría señalar que

para el caso de los sistemas bibliotecarios empieza a tener una cierta importancia la existencia de una norma internacional —ANS Z39.58 o ISO 8777— que desarrolla un modelo de lenguaje de recuperación denominado «Common Command Language» —CCL—, así como la norma ANS Z39.50 y sus equivalentes ISO 10162 y 10163 para la búsqueda y recuperación de información en bases de datos remotas. La generalización del uso de la norma ISO 8777 entre los OPAC's de muchas bibliotecas, la está convirtiendo de hecho en un estándar para el acceso a la información catalográfica, lo que podría significar una cierta constricción en el desarrollo de las aplicaciones bibliotecarias, especialmente cuando éstas pretenden funcionar en red, puesto que el sistema previsto consiste en transferir cadenas de caracteres ASCII formadas por comandos CCL con sus correspondientes parámetros entre las distintas bibliotecas interconectadas, de tal forma que sea posible, por ejemplo, lanzar la misma demanda informativa a todo un conjunto de sistemas en red con una sola operación de búsqueda. Este tipo de tareas, que anticipan el futuro de sistemas que parecían destinados a desaparecer con la aparición del modelo relacional, empiezan a tener cierta vigencia en temas tales como el del control de autoridades [McCallum 86] en Estados Unidos o la consulta de «multi-catálogos» vía Internet [Farley 91].

Existen otras características funcionales relacionadas con la recuperación que me limitaré a enumerar por ser de cierto interés en los sistemas bibliotecarios.

La posibilidad de manipular las llamadas, de forma un tanto pretenciosa, «ecuaciones de búsqueda» es una función muy necesaria para obtener unos mejores resultados en la relación del sistema con el usuario. Dicha manipulación puede consistir en tareas tales como reordenación de las búsquedas, reselección de las combinaciones, salvaguarda de las ecuaciones, etc.

La existencia de ficheros históricos que guardan referencia de la actividad recuperadora del sistema es una herramienta definitiva para evaluar el uso que se hace del mismo, especialmente desde los terminales de acceso público, lo que nos permitirá mejorar aquellos aspectos deficientes de su funcionamiento que puedan ser detectados, tanto en el software como en los contenidos de la base.

V.B.3. *Actualización de la información*

Convencionalmente bajo esta denominación se recogen todas aquellas funciones relacionadas con la introducción de nuevos documentos en la

base de datos y la modificación de los existentes. Estas tareas, en primer lugar, se realizan mediante la utilización de formatos de entrada que pueden ser definidos y modificados por los usuarios. Dichos formatos deben permitir la realización de algunos de los procesos de verificación comentados anteriormente, además de facilitar la introducción de los datos mediante las funciones de edición estándar. La carga de los documentos se realiza normalmente en un momento posterior al de la edición ya que ésta se realiza en «tiempo real» mientras que la carga suele ser un proceso «batch» o «pseudo-batch». Esto permite la utilización de sofisticadas funciones de edición muy necesarias en la introducción de información MARC. Como se ha puesto de manifiesto recientemente [Hípola 91], la utilización de normas como las ISO 8879 e ISO 9069 —«Standard Generalised Mark-up Language», SGML— facilitará la intercambiabilidad de la información en contextos aun más amplios que el puramente bibliotecario, lo que hace necesario facilitar la generación de entradas en el proceso de edición que se ajusten a normas de este nivel. Este tipo de hechos dan aún mayor valor a la posibilidad de utilizar editores «ad-hoc» en el momento de la introducción de los documentos.

Existen, sin embargo, dos dificultades importantes en relación con el proceso de edición que es necesario subsanar. La utilización de editores convencionales hace difícil la validación de las diferentes entradas contra bases de datos de autoridades u otras informaciones tabulares. Esto sólo se puede resolver por la vía de la utilización de programas de captura de información específicos, por lo que las funciones de edición avanzadas deberán ser implementadas expresamente. Por otro lado, cuando se realiza la modificación de documentos de gran tamaño este proceso suele llevar algún tiempo, lo que en entornos multiusuario podría dar lugar a inconsistencias en la actualización si cada documento está permanentemente disponible para modificación por todos los usuarios. La única forma de resolver este problema es que el sistema realice un control de bloqueos de los registro para modificación —no para consulta— de los restantes usuarios cuando uno de ellos ha entrado a modificarlo. Es, por último, necesario recordar aquí que las modificaciones de documentos ocasionan problemas de crecimientos desproporcionados de los ficheros invertidos y diccionarios en las bases de datos, lo que hace necesario disponer de herramientas software de análisis de los índices y de mantenimiento de los mismos; sobre este tema trataré en un punto posterior de este mismo capítulo.

V.B.4. *Control de entradas*

Aunque el tema del control de la forma de las entradas está íntimamente ligado a los problemas de la actualización de la información me ha parecido necesario comentarlo por separado debido a su especial trascendencia en el caso que nos ocupa y la cantidad de matices que abarca.

El procedimiento más eficaz para controlar las entradas en un catálogo bibliotecario es la herramienta denominada «gestor de thesaurus». Hablar de este tipo de aplicación significa que existe la posibilidad de integrarla en el sistema de tal forma que sea posible utilizar un thesaurus en línea, tanto en el momento de la entrada de los datos, como en el de la recuperación. Esta integración total permitirá también la gestión de las relaciones existentes entre las diferentes entradas pertenecientes al thesaurus —véase normas ANS e ISO al respecto—, lo que deberá incrementar la eficacia en las recuperaciones y el control de las entradas, especialmente por lo que afecta a los llamados campos de autoridades. Subfunciones tales como la sustitución automática de los términos no aceptados por los aceptados, la recuperación a partir de un término del thesaurus definiendo un ámbito de términos relacionados para realizar la búsqueda con todos ellos, la realización de cambios en las relaciones entre los términos sin que esto afecte a los documentos ya indizados, la posibilidad de realizar modificaciones globales controladas de los documentos a partir de los términos del thesaurus, etc., forman parte de las tareas requeridas corrientemente de un gestor de thesaurus.

Existen, sin embargo, funciones menos frecuentes pero imprescindibles para realizar una gestión adecuada de la información en los entornos bibliotecarios. La necesidad de controlar la información por campos e incluso subcampos MARC en el caso de la información bibliográfica normalizada obliga a que el gestor de thesaurus controle la actualización de determinados campos, ofreciendo la opción de seleccionar un thesaurus u otro según el campo de que se trate. Esta operación es inevitable si se pretende realizar un control de autoridades preciso.

En resumen, el gestor de thesaurus puede ser una herramienta adecuada para la realización del control del vocabulario en el momento de la edición de los documentos siempre que satisfaga ciertas exigencias relativas al control de las entradas de autoridades y de sus referencias cruzadas [Gare 84].

Para concluir este apartado quiero hacer mención de una solución de control terminológico utilizada con cierta frecuencia en los entornos IRS, aunque al utilizar recursos ajenos al propio sistema se aparta ligeramente del criterio que vengo manteniendo de no entrar en lo que podríamos llamar soluciones mixtas. En este caso haré una excepción por tratarse de una solución muy extendida. Me estoy refiriendo a la utilización de fiche-

ros «planos» para realizar el control terminológico. Esta solución se diseña e implementa al margen del propio sistema pues esos ficheros no forman parte de la estructura del IRS. Su ventaja fundamental es la sencillez de mantenimiento y operatividad, su inconveniente la falta de integración con el resto del sistema, que podría plantear ciertas dificultades de gestión especialmente en la fase de recuperación de la información.

V.B.5. *Salida de la información recuperada*

La práctica totalidad de los sistemas comerciales del tipo IRS disponen de utilidades para el formateo previo al envío hacia el dispositivo de salida. Estas utilidades son manejables por los usuarios, aunque algunas de ellas requieren ciertos conocimientos de programación para su manejo. La versatilidad que estos medios ofrecen garantiza la adecuación de los formatos de salida a las características de los usuarios finales que serán sus destinatarios. La dificultad estriba más en la determinación de la mejor forma de establecer esa adecuación. Existen en esta línea algunos trabajos [Mathews 87, Yee 91] que hacen recomendaciones respecto de los formatos de visualización y/o impresión de los documentos recuperados en las bases de datos catalográficas. Incluso existen estudios sobre el «comportamiento» de los usuarios frente a los distintos formatos para poder mejorar su diseño [Prasse 91]. Aunque estos trabajos introducen serias dudas sobre la conveniencia de utilizar los formatos bibliotecarios tradicionales porque son poco claros para los usuarios, resulta esencial desde el punto de vista del administrador del sistema que éste ofrezca la posibilidad de realizar de forma amigable todo tipo de modificaciones en los formatos de salida. En este aspecto podemos decir que los IRS son sistemas que se adecuan a las exigencias del entorno bibliotecario.

Aunque he mencionado antes que debe ser posible enviar los formatos de los documentos seleccionados a cualquier dispositivo de salida, es preciso recordar que básicamente los dispositivos utilizados son de visualización, de impresión y de almacenamiento masivo. Este último es especialmente útil para garantizar la reutilización de las imágenes de los documentos. Esta reutilización suele producirse al remitirlos a través de sistemas de correo electrónico, al enviarlos a dispositivos de edición electrónica o al ser usados para realizar difusión selectiva de la información en entornos locales o remotos.

Por último, aunque lo trataré de manera más específica en un apartado posterior, es preciso decir aquí que la transformación de los documentos en algunos de los formatos estándar exige la utilización de rutinas de «mapping» que se apartan por su complejidad de las posibilidades de las utilidades que estoy comentando en este apartado.

V.B.6. *Relaciones con el exterior*

En cualquier sistema de recuperación documental —IRS— existen tres aspectos funcionales que condicionan de manera definitiva la interacción del usuario con el sistema:

a) El interface de menús que permite el uso de subfunciones básicas.
b) El sistema de comandos que permite a los usuarios la gestión de forma «nativa».
c) La interacción del sistema con otros de igual o distinta naturaleza.

El conjunto de funciones y subfunciones agrupadas en estos tres apartados conforman lo que denomino las relaciones del sistema con el exterior. De manera más específica me parece importante resaltar que en un sistema bibliotecario todo aquello que hace referencia a la interacción con el usuario —profesional o no— tiene especial interés porque los usuarios —especialmente en los OPAC— son eventuales, y cuando no lo son —usuarios profesionales— la eficacia de su actividad está muy condicionada por la forma en que se relaciona con la aplicación que manejan, como han puesto de manifiesto recientes estudios [Prasse 91].

Para tratar de manera detallada los apartados antes citados es preciso anticipar que, en el caso de las aplicaciones bibliotecarias desarrolladas utilizando sistemas de este tipo, tanto los interfaces de menús como el sistema de comandos, aunque tengan el carácter de herramientas de desarrollo, podrán ser consideradas parte específica de la aplicación bibliotecaria. Esto quiere decir que aquí sólo serán tenidos en cuenta en la medida en que puedan servir como punto de referencia para valorar posibilidades funcionales operativas en la aplicación de gestión bibliotecaria.

Hecha esta observación empezaré por hacer algunos comentarios sobre el desarrollo de los interfaces de usuario en gestores de bibliotecas desarrollados a partir de IRS. La arquitectura de los IRS permite con facilidad desarrollar interfaces de usuario específicos para cada aplicación que manejen recursos ajenos al propio sistema: entornos gráficos tipo WIMP —Windows, Icons, Menus, Pointers—, DBMS del tipo relacional con 4GL, programas convencionales de captura de datos —«Data entry»—, etc. Esta facilidad, que supone ventajas evidentes tanto para el usuario como para el desarrollador, es posible por el hecho de que la carga de los documentos se hace mediante procesos batch o pseudobatch en tiempos de proceso diferentes a los de la edición de los documentos, lo que permite preparar un documento con una aplicación desarrollada con unos determinados recursos para, a continuación, entregar el documento preparado al programa de carga para efectuar la indización del mismo. Según esto, bajo

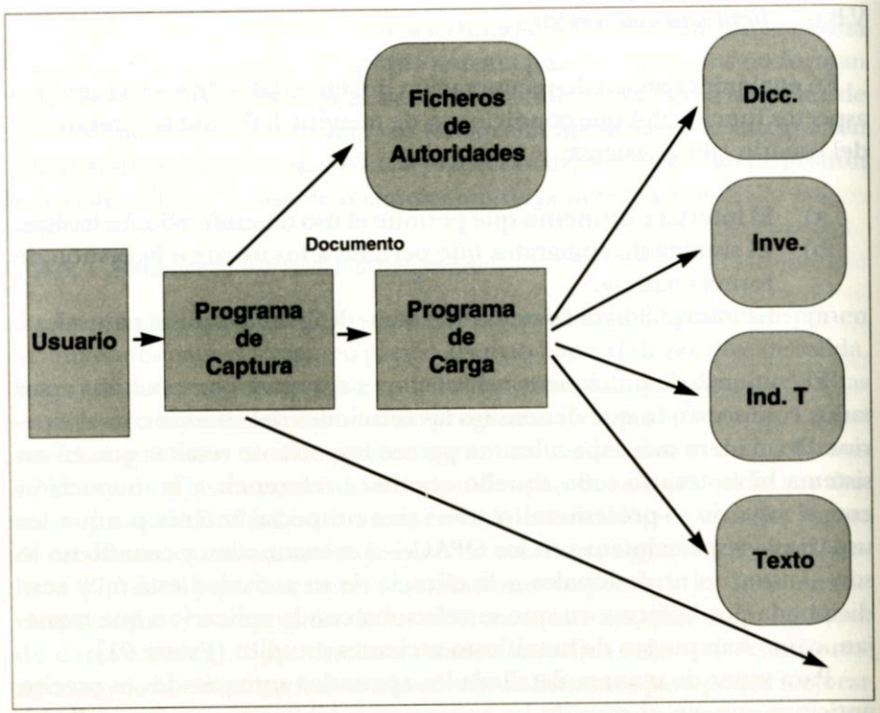


Gráfico 6

apariencias muy diferentes podría haber estructuras de datos del tipo IRS sin que el usuario se percatara de ello. Este el caso de programas tan distantes en el tiempo, el espacio y la apariencia como Notis, Absys y Unicorn, que utilizan el mismo producto del tipo IRS para la indización de la información bibliográfica, BRS/Search, y sin embargo sus usuarios probablemente no encontrarían en ellos muchos elementos comunes. Esta circunstancia se debe a la existencia de diferentes programas de captura de la información aunque el programa de carga sea común (véase figura).

Desde esta perspectiva, la forma concreta del interface, siendo una cuestión vital como dije antes, se convierte en un problema abordable con gran libertad, al contar con la relativa autonomía de la aplicación de captura respecto de la de carga. Algunas de las investigaciones más recientes en el campo del diseño de interfaces para los gestores bibliotecarios [Vizine-Goetz 91] ponen de manifiesto la insistencia en la utilización de técnicas informáticas ligadas a los entornos gráficos —WIMP—. Esto ha hecho que determinadas bibliotecas hayan modificado los programas de captura de datos de sus aplicaciones introduciendo estos nuevos elementos sin que para ello tengan que producirse modificaciones sustanciales en sus estructuras de datos o en sus procesos de carga.

Al comienzo de este apartado aludí también a la interacción del sistema con otros sistemas como función esencial en el ámbito de las relaciones con el mundo exterior. Habida cuenta de las limitaciones funcionales que los IRS tienen cuando tratan de manejar información estructurada, la posibilidad de desarrollar aplicaciones que integren otros recursos de gestión de información junto con un IRS se convierte en una necesidad. Para hacer operativa esta posibilidad es necesario conocer su viabilidad técnica, es decir, si se pueden hacer llamadas a procesos externos al IRS en cualquier momento que sea necesario, o bien, si desde otros procesos se pueden ejecutar programas que formen parte del IRS. Ambas posibilidades, juntas o separadas, pueden permitirnos la utilización de forma compartida de recursos software heterogéneos que mejoren las prestaciones del conjunto en términos de gestión de la información en su conjunto. Como he repetido en diversas ocasiones a lo largo de este trabajo, la heterogeneidad formal de la información procesada en una biblioteca obliga al desarrollo de sistemas automáticos de gestión muy flexibles, no tanto en cuanto a los procesos que son capaces de realizar, sino en cuanto a las estructuras informativas que admiten para su tratamiento posterior.

La viabilidad técnica de la realización de llamadas a procesos externos viene dada por la posibilidad de definición, por parte del usuario, de los flujos de trabajo en el conjunto del sistema como parte del proceso de «tailoring» que el usuario realiza en el momento de su instalación. Esta posibilidad debe haber sido prevista por el diseñador del sistema. De no ser así difícilmente podrá obviarse la dificultad. En algunos de los IRS que se utilizan hoy día de forma comercial sus elementos esenciales son adaptables por los usuarios y fácilmente integrables en otras aplicaciones [Ftrs 89]. En cualquier caso también hay que decir que estos aspectos de relación entre recursos software son muy dependientes de las características de los entornos operativos bajo los que se ejecutan.

V.B.7. *Administración del sistema y mantenimiento*

En sistemas de gestión de bases de datos las funciones relacionadas con el mantenimiento y la administración del sistema son imprescindibles debido a la posibilidad de que se produzcan inconsistencias informativas en las bases, provocadas por funcionamientos deficientes del propio sistema o de forma accidental. Uno de los procedimientos más frecuentes para paliar este tipo de problemas es incluir entre los procesos de arranque del sistema uno de reconstrucción automática de ficheros. Este proceso puede

funcionar gracias a la existencia de ficheros históricos —«journals»— que guardan la información de las peticiones realizadas al sistema hasta la realización del último cierre, con lo que sería posible, en el caso de que se detectara alguna inconsistencia, recuperar la información perdida o deficiente. En cualquier caso estos sistemas siempre deben incluir procedimientos de chequeo capaces de suministrar información sobre el estado de los ficheros de las distintas bases de datos que gestionan.

Otros aspectos funcionales importantes relacionados con la administración del sistema son los que hacen referencia a la seguridad. En este sentido lo habitual es que existan controles mediante el uso de perfiles de usuarios basados en códigos de identificación. Estos controles serán eficaces en el caso de que sea posible cruzar tipos de funciones, con tipos de usuarios y tipos de informaciones a la hora de generar los perfiles de autorización de los usuarios.

Además de éstas hay otras funciones de administración y mantenimiento que no voy a desarrollar aquí y que me limitaré a enunciar:

- Funciones estadísticas.
- Funciones de control de accesos.
- Funciones de ajuste funcional.
- Funciones de monitorización y depuración.

V.C. Conclusiones

a) Los IRS mantienen una estructura física de datos que permite la definición de estructuras lógicas complejas, sin limitación en la extensión de sus elementos, con múltiples repeticiones en cada uno de ellos, por lo que, en este sentido, son sistemas apropiados para la gestión de información textual como la formada por las referencias bibliográficas normalizadas de los catálogos bibliotecarios.

b) Aunque dentro de la clasificación de Belkin y Croft [Belkin 87] los IRS, de los que fundamentalmente he estado hablando aquí, son los que utilizan técnicas del tipo «exact match», ya mencioné anteriormente que al indexar elementos característicos de cada campo podrían utilizar técnicas del tipo «partial match», lo que permitirá mejorar considerablemente las prestaciones de recuperación de la información de los sistemas bibliotecarios.

c) Aunque sólo valoráramos las posibilidades de recuperación mediante técnicas booleanas y de proximidad de estos sistemas habría que admitir que, respecto de otros, éstos ya aportan la utilización de los operadores de proximidad, que en el tratamiento de campos con informaciones como títulos o abstracts resultan de gran utilidad [Keen 92]. Por otra parte el hecho de que los IRS trabajen con conjuntos de documentos facilita la respuesta rápida de las búsqueda mediante operadores y, por lo tanto, permite desarrollar ecuaciones más complejas.

d) Las operaciones de cálculo realizadas sobre campos con información numérica son muy limitadas, lo que dificulta considerablemente la gestión de los llamados procesos no bibliográficos de una biblioteca. Esta dificultad, que viene siendo tan tradicional como incomprensible en los IRS, hace necesario el desarrollo con recursos ajenos al sistema de todos los procesos en que estén implicados cálculos. Es cierto que las dificultades técnicas para la gestión sincrónica de información estructurada y no estructurada quedaron resueltas desde el punto de vista de la investigación, hace tiempo. A pesar de que se anuncia casi de manera permanente la aparición de productos comerciales que sean capaces de gestionar simultáneamente imágenes, textos e información estructurada [Reid 90], de momento estas capacidades de gestión las encontramos en productos distintos.

e) La facilidad que aportan estos sistemas de indización de múltiples claves por cada elemento de la estructura documental lógica es idónea para la gestión de referencias catalográficas. Esta facilidad es consecuencia de la estructura de datos en la que se apoya el IRS y sin ella sería imposible efectuar una recuperación de información en base a los elementos significativos —palabras clave o similar— de cada campo. Por el contrario, este tipo de indización multiclave ocasiona los consabidos problemas lingüísticos en la recuperación —homografías, sinonimias, variantes gramaticales, etc.— que sólo pueden ser obviados mediante técnicas no disponibles aún en la mayoría de los productos comerciales disponibles o mediante el control exhaustivo de las entradas.

f) Aunque diversos autores han puesto de manifiesto la falta de control de las entradas de punto de acceso en los procesos de actualización de las bases de datos catalográficas gestionadas por IRS [Kemp 88], desde que estos sistemas disponen de gestores de thesaurus perfectamente integrados, el control de las autoridades en los catálogos se viene realizando por medio de estos gestores. Esta

es la razón por la que resulta tan efectivo el control de las referencias cruzadas en los procesos de actualización y recuperación. Por otra parte, la posibilidad de hacer llamadas a otras bases de datos desde la que se está actualizando en un momento dado permite consultar bases de datos de autoridades en MARC que por contener una gran cantidad de datos de cada autoridad no resulta cómodo estar consultándolas constantemente pero sí eventualmente.

g) Una de las dificultades más importantes para la utilización de los IRS en el desarrollo de sistemas bibliotecarios es la relacionada con la diversidad de entidades informativas que es necesario gestionar de forma concurrente en una biblioteca. Los IRS por norma son capaces de gestionar múltiples bases de datos pero no de forma concurrente, incluso pueden gestionar varias bases de datos de forma encadenada, pero esto no permitiría la gestión de estructuras de datos en las que convivieran diversas entidades en el curso de la realización de la mayor parte de los procesos de gestión de la información. Esta circunstancia convierte a los IRS en muy adecuados para la gestión del catálogo de una biblioteca y muy poco adecuados para la gestión de funciones básicas como las adquisiciones, el control de las publicaciones periódicas o el préstamo, porque en ellas se hace imprescindible la interrelación de elementos informativos pertenecientes a varias entidades básicas.

h) Otra de las críticas que tradicionalmente se ha hecho a los IRS tiene que ver con su incapacidad para controlar relaciones de distinta naturaleza entre los documentos. El tratamiento de los documentos como unidades independientes ha sido una constante en los IRS, si bien se trata de algo que ha empezado a cambiar recientemente. La posibilidad de establecer relaciones entre dos o más documentos y recuperar los relacionados a partir de cualquiera de ellos es hoy una facilidad disponible en algunos sistemas de recuperación de información. Esta función es especialmente útil en la gestión de catálogos para controlar las relaciones existentes entre referencias que habiendo sido generadas de forma independiente deben mantener entre sí una relación tal que sea posible la remisión de una a la otra para que el usuario tenga una información lo más completa posible de unidades documentales compuestas —obras multivolumen, catalogaciones analíticas, referencias de «holdings», etc.—.

VI

LOS SISTEMAS DE GESTIÓN DE FICHEROS

Como dije anteriormente, el desarrollo de los sistemas integrados de gestión bibliotecaria demuestra que sus orígenes están ligados al uso por parte de los informáticos de sistemas de gestión de ficheros —File Management Systems, FMS—. Estos sistemas de gestión de datos empezaron a utilizarse en las bibliotecas con el comienzo de la automatización bibliotecaria [Tedd 87], aunque, de la forma en que trataré de ellos aquí, empiezan a existir cuando se realizan los primeros intentos de gestión integrada de las bibliotecas. La identificación, por tanto, del uso de estos modelos de gestión en los entornos bibliotecarios resulta sencilla porque son las primeras aplicaciones con vocación de sistemas integrados de gestión bibliotecaria, desarrolladas desde principios de los años setenta, las que se sirven del nivel de conocimientos que los informáticos tenían en ese momento en el campo de la gestión de datos para aplicarlo a la gestión de la información producida y procesada por un centro bibliotecario. En alguno de los buenos trabajos realizados a comienzo de los setenta sobre la situación de la tecnología de herramientas software para la organización de la información en grandes bases de datos [Prywes 72] se pone de manifiesto la complejidad que entrañaba la utilización de los llamados Data Description Languages —DDL—, que actuaban como intermediarios entre la estructura lógica de los datos diseñada y la estructura física de los mismos en los dispositivos de almacenamiento secundario. Los nombres que reciben los sistemas desarrollados con estos lenguajes son diversos, pero indudablemente uno de los más usados es Sistemas de Gestión de Ficheros o simplemente Sistemas de Ficheros [Kruglinski 83]. Estas denominaciones revelan que en la base de estos sistemas se encuentra la peculiar interacción que en cada caso se produce entre la aplicación y el método de acceso a los datos permitido por el sistema operativo bajo el que se ejecuta la aplicación. Dicha interacción siempre se produce haciendo asequible la lógica de la información procesada a un conjunto físico de ficheros —de ahí su nombre— gestionados directamente mediante recursos software de nivel muy básico, y mediante recursos software de nivel secundario: el sistema de gestión de ficheros desarrollado.

Si bien ésta es la filosofía general de funcionamiento de este tipo de aplicaciones, trataré en este caso como en los anteriores, de describir los aspectos comunes a estos sistemas cuando han sido desarrollados para resolver los problemas de la gestión integrada de los centros bibliotecarios.

VI.A. Conceptos básicos

Antes de que aparecieran los primeros Sistemas de Gestión de Bases de Datos —DBMS— los desarrolladores de aplicaciones tenían que enfrentarse con la tediosa tarea de hacer aplicaciones que manejaban estructuras de datos en ficheros informáticos cuya estructura física no siempre tenía mucho que ver con la lógica de los datos que se pretendía procesar. Esta diferencia, que está en la base de cualquier sistema de información automatizado —SIA—, en el caso de los FMS, debido a la dependencia que los DDL tienen del hardware y software básico bajo el que funcionan, son las propias aplicaciones las que tienen que hacer la imprescindible conversión de la estructura lógica de los datos a su estructura física, lo que se traduce en el diseño y mantenimiento de estructuras de ficheros capaces de atender los problemas lógicos de los datos procesados. Mientras que en los DBMS o IRS son los propios sistemas de gestión los que realizan esa conversión, por lo que a nivel de las aplicaciones sólo se define la estructura lógica de los datos, en los FMS es necesario definir estructuras físicas de datos en los dispositivos de almacenamiento correspondientes y trabajar con ellas desde la aplicación con el fin de hacer asequibles esos datos a los usuarios, convirtiéndolos desde su estructura física a lógica. Esta característica capital de los FMS es lo que se viene definiendo como dependencia del hardware y software por parte del sistema y tiene importantes implicaciones en el desarrollo de las aplicaciones que luego trataré de concretar. Pero ahora me gustaría abundar algo más en la raíz misma de esta dependencia que, como puse antes de manifiesto, está ligada al hecho de que los DDL han sido diseñados para operar en entornos hardware/software concretos, por lo que un DDL sólo tiene sentido ligado a un cierto tipo de ordenador y un determinado sistema operativo. Esto, de entrada, quiere decir que la portabilidad de las aplicaciones desarrolladas utilizando este tipo de recursos es muy escasa. Aunque la informática trató de paliar este problema definiendo lenguajes de programación estándar cuya sintaxis era común en todos los entornos, lo que variaba de un entorno a otro era el compilador. Aun así el nivel de los recursos específicos de sistema operativo utilizados —método de acceso fundamentalmente— era tal que se dificultaba constantemente la portabilidad.

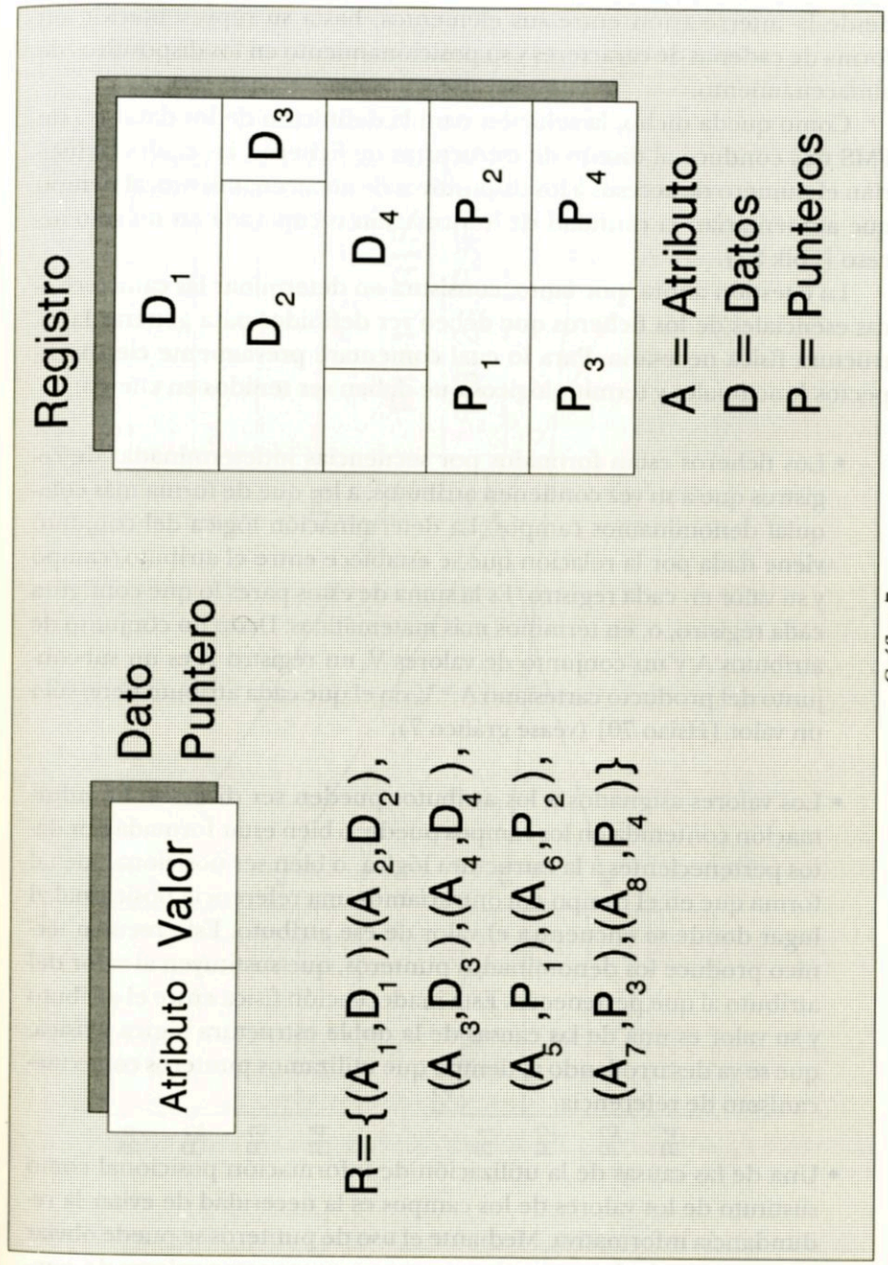


Gráfico 7

En definitiva, en un FMS, se realiza mediante un lenguaje de programación la descripción de todos los aspectos de una estructura de datos, desde la interrelación entre sus elementos, hasta su representación en forma de cadenas de caracteres y su posicionamiento en los dispositivos de almacenamiento.

Como queda dicho, la solución para la definición de los datos en un FMS nos conduce al diseño de estructuras de ficheros, las cuales reducirían el número de accesos a los dispositivos de almacenamiento, al tiempo que aumentarían la cantidad de información recuperada en un solo acceso [Folk 87].

La cuestión ahora, por tanto, consistirá en determinar las características esenciales de los ficheros que deben ser definidos para generar la estructura física necesaria. Para lo cual comentaré previamente ciertos aspectos funcionales y terminológicos que deben ser tenidos en cuenta:

- Los ficheros están formados por secuencias indeterminadas de registros que a su vez contienen atributos, a los que de forma más coloquial denominamos campos. La determinación lógica del conjunto viene dada por la relación que se establece entre el atributo/campo y su valor en cada registro. Es la suma de estos pares lo que configura cada registro, o, en términos más matemáticos: Dado un conjunto de atributos A y un conjunto de valores V, un registro R es un subconjunto del producto cartesiano A * V, en el que cada atributo tiene sólo un valor [Hsiao 70] (véase gráfico 7).

- Los valores asignados a los atributos pueden ser diversos. La información contenida en los campos puede, o bien estar formada por datos pertenecientes a la estructura lógica, o bien ser posicional, de tal forma que en el campo encontraríamos una referencia posicional al lugar donde se encuentra el valor de ese atributo. Este recurso técnico produce los denominados punteros, que sustituyen al valor del atributo al que pertenecen. Esta inadecuación física entre el atributo y su valor es una de las causas de la doble estructura lógica y física, que se va desarrollando al tiempo que utilizamos punteros como mecanismo de referencia.

- Una de las causas de la utilización de información posicional como sustituto de los valores de los campos es la necesidad de evitar la redundancia informativa. Mediante el uso de punteros se puede obviar la repetición de datos que tienen que aparecer como valores de campos en varios registros pertenecientes a distintos ficheros de la base.

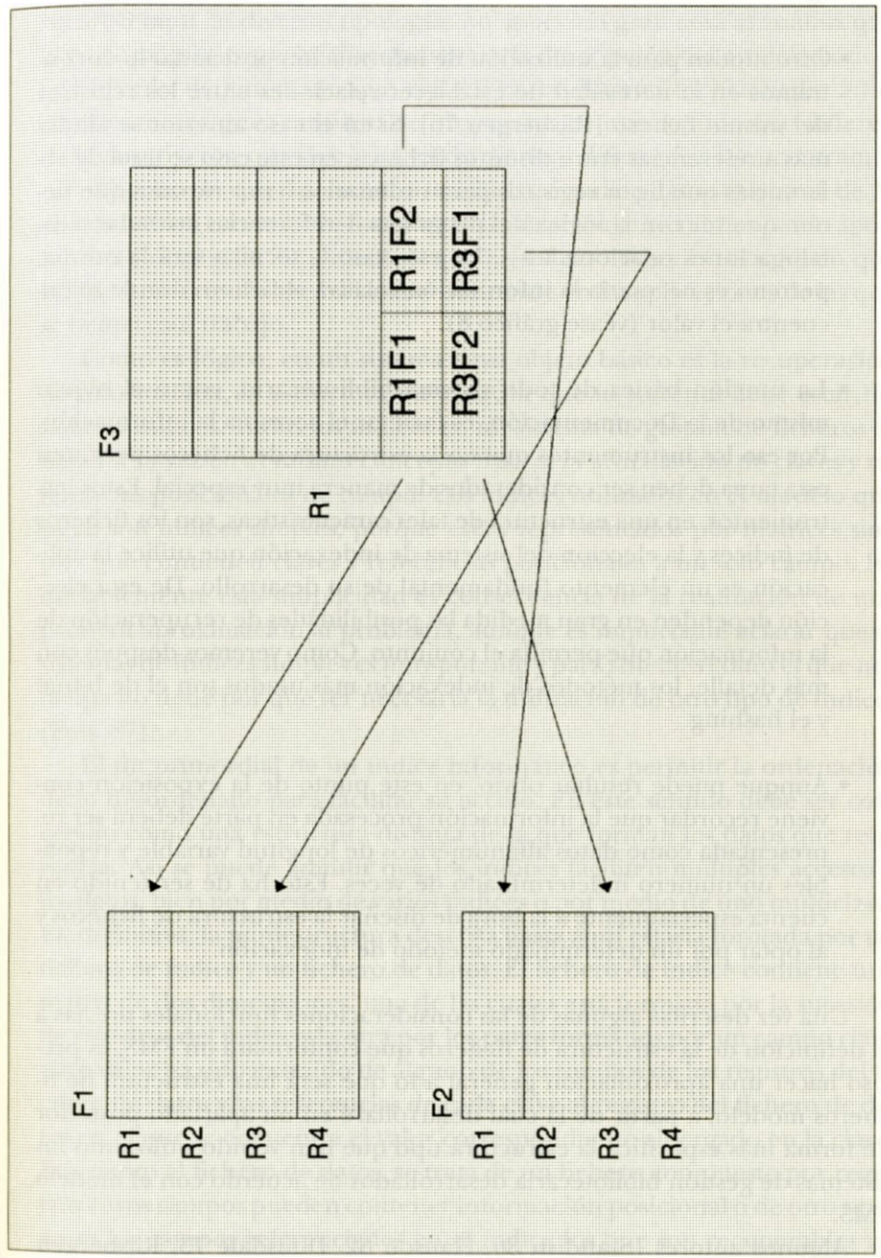


Gráfico 8

Este aspecto tiene una especial importancia en la gestión catalográfica.

- Otro motivo para la utilización de información posicional lo encontramos en la necesidad de establecer relaciones entre los registros del mismo fichero [Rijsbergen 76]. Si en el caso anterior se aludía más a referencias entre distintos ficheros, en este caso se trata de referencias que ligan registros para ordenarlos o por razones que tienen que ver con la lógica del programa —referencias cruzadas o catalogaciones relacionadas—. En este caso la técnica será la misma, pero no es necesaria la información relativa al fichero donde se encuentra el valor (véase gráfico 8).

- La función básica de todo sistema bibliotecario, como el objeto mismo de la Documentación, es facilitar el acceso a la información. Por eso los instrumentos que en la estructura de ficheros permiten esta tarea deben ser considerados de manera muy especial. Estos instrumentos, en una estructura de tales características, son los ficheros de índices y la elección del sistema de indexación que utilice la aplicación es un elemento fundamental de su desarrollo. De esta elección dependen en gran medida las posibilidades de recuperación de la información que permita el conjunto. Como veremos después con más detalle, los métodos de indexación más usados son el de b-tree y el hashing.

- Aunque puede resultar obvio, en este punto de la exposición conviene recordar que la información procesada en parte deberá ser representada como datos alfanuméricos de longitud variable y repetibles un número indeterminado de veces. Esto ha de ser tenido en cuenta especialmente a la hora de diseñar la estructura de ficheros y al optar por un determinado método de indexación.

Una vez descritas algunas de las consideraciones funcionales previas a la definición de la estructura de ficheros que conformará un FMS, es preciso hacer una aproximación general a lo que será una estructura de ficheros modelo, a partir de la cual desarrollaré en un apartado posterior de forma más específica la estructura tipo que han venido utilizando los sistemas de gestión bibliotecaria desarrollados de acuerdo con el modelo FMS.

Algunos autores [Standish 80, Hanson 82, Dimsdale 73, Rijsbergen 76, etc.] realizan una detallada tipología de estructuras de ficheros que van desde los más simples —secuenciales— hasta agrupaciones complejas —multilistas o de almacenamiento distribuido–, pasando por las modalidades clásicas —secuenciales indexados o invertidos—. En todo caso, recorrer aquí las diversas tipologías no aportaría gran cosa al análisis que nos ocupa, por lo que iré directamente a la descripción del modelo básico, advirtiendo de antemano que existen en las fuentes mencionadas anteriormente una gran cantidad de propuestas en este sentido que difieren de la descripción que sigue. La razón de estas diferencias la encontramos en el hecho de que las fuentes reseñadas estudian las estructuras de ficheros, o bien tratando de hacer una clasificación, o bien intentando aportar una nueva solución, mientras que en mi caso describo el modelo que la industria del software ha venido utilizando en las aplicaciones de las que se ocupa este trabajo.

Como es lógico, en un sistema cuyo objeto básico es la recuperación de información, la estructura informativa que lo sustenta debe estar basada en índices. Todos los índices en un sistema de gestión de ficheros se basan a su vez en el mismo concepto básico: la articulación de claves y referencias posicionales de campo. Los índices utilizados en el modelo que sigue son índices simples, porque están representados por matrices simples que contienen claves y referencias posicionales a un solo campo. Indudablemente esta simplicidad es consecuencia de la realización de una primera aproximación al problema, aunque es importante aclarar que el sistemas de índices simples es muy potente y para sistemas como el que nos ocupa no tiene por qué ser necesaria la utilización de otro tipo de índices [Folk 87].

El fin primordial de un índice informático es permitir la ordenación de lo desordenado para facilitar su acceso. En este sentido debe ser concebido como una estructura distinta de la que forman los datos que referencia. Así se puede admitir que los índices faciliten múltiples accesos a los datos, bien por medio de varios índices o por medio de uno multiclave. En definitiva, la miniestructura descrita hasta aquí estaría formada por un fichero de índice y un fichero de datos. El fichero de índice contiene una matriz de dos dimensiones, una de las cuales está formada por la sucesión de registros del índice y la otra por los pares formados por un campo clave cuyo valor ha sido extraído de alguno de los campos de los registros del fichero de datos y las referencias de cada clave al registro del fichero de datos en el que se encuentra el valor correspondiente albergado en la clave. En cuanto al fichero de datos, se trata de un fichero compuesto por registros cuyos campos pueden contener información posicional o de otro tipo, y son los campos referenciados en el índice los que son recuperables rápidamente, porque el acceso al fichero de datos se hace secuencialmente y, por tanto, lentamente.

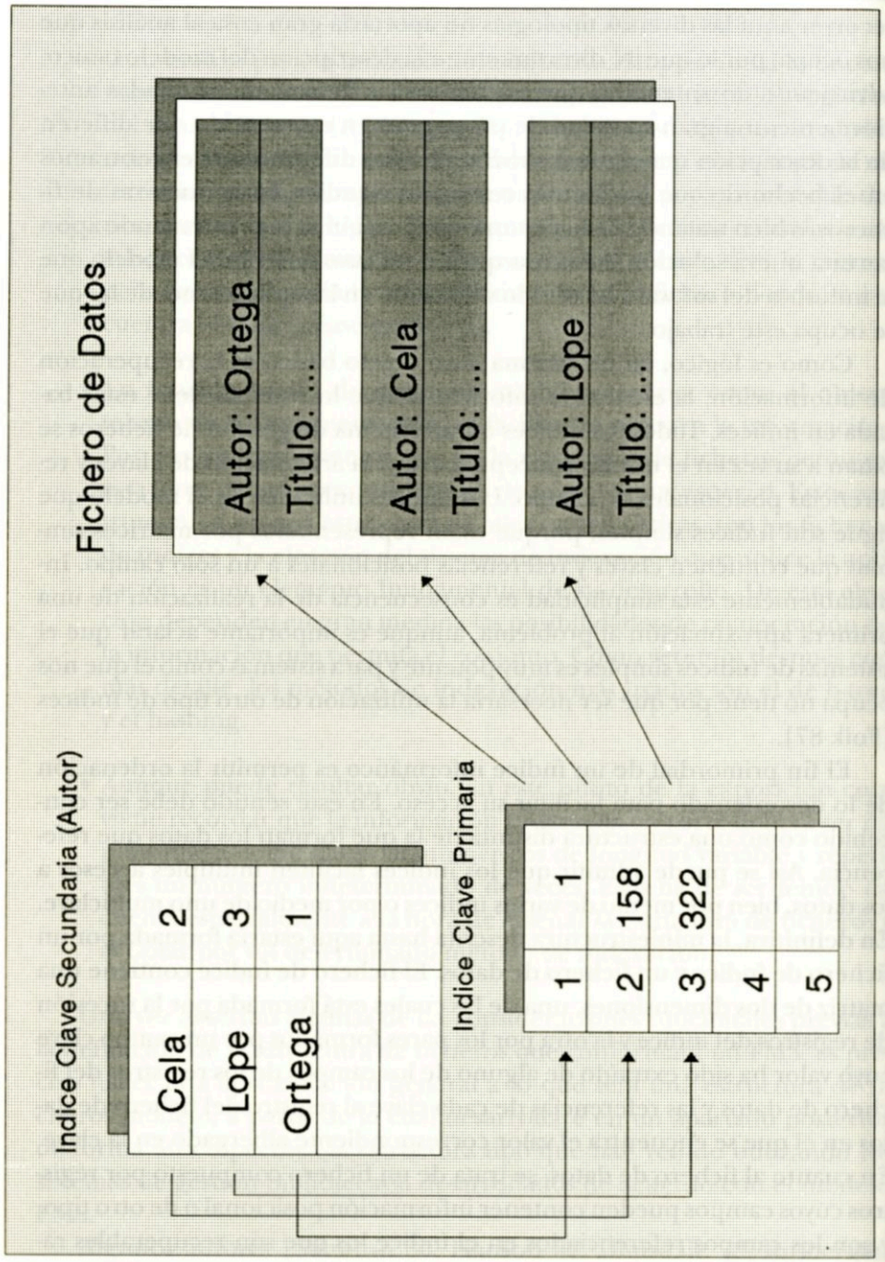


Gráfico 9

En todo sistema de ficheros basado en índices y con registros y campos de datos de longitud variable el acceso a los registros del fichero de datos se realiza a través de una clave primaria, de tal forma que, si son necesarios otros accesos por claves de campos, los índices generados a partir de los valores de estos campos referencian las claves primarias y éstas a su vez los registros del fichero de datos, de tal manera que los accesos normales requieren la consulta de dos índices, uno de claves secundarias y siempre el de las claves primarias —véase el gráfico 9—. Visto de esta forma, el fichero de índice se puede comportar como un fichero invertido. Basta con que el índice deje de ser simple y junto a cada clave pueda haber una lista de referencias de registros de datos en los que se encuentra el valor de dicha clave.

Funcionalmente las únicas dificultades que una estructura como ésta debe resolver son las de la actualización de los índices, asunto que está muy condicionado por el método de indexación, por lo que será necesario hacer algunas observaciones al respecto antes de continuar con el análisis funcional del problema.

Como ya dije anteriormente uno de los sistemas de índices más usados es el denominado b-tree. Este método de indexación, a pesar de estar tan extendido, se desarrolló a finales de los años sesenta. El objeto de su desarrollo fue crear una herramienta de acceso a los datos que fuera eficaz y de bajo costo. Cuando se publicó el famoso artículo de Bayer y McCreight [Bayer 72], a pesar de no aparecer la que luego sería su denominación definitiva, sus fundamentos técnicos conocieron la luz. El problema al que se quería dar solución con el desarrollo de esta nueva técnica era específicamente el de la recuperación en grandes bases de datos cuyos índices no podían ser albergados en la memoria principal de los ordenadores, por lo que se hacía necesario desarrollar algoritmos capaces de gestionar índices en dispositivos de almacenamiento secundario que tuvieran que realizar la menor cantidad posible de accesos al fichero de índice para localizar la clave buscada al tiempo que eran económicos a la hora de añadir o borrar claves.

Parece evidente que el problema fundamental en el acceso a través de ficheros de índice es que los accesos a dispositivos de almacenamiento son lentos. Para obviar este inconveniente se mejoró el procedimiento de las búsquedas binarias introduciendo procedimientos matemáticos que reducían la distancia entre los nodos de los árboles y por consiguiente el número de accesos necesarios para localizar una clave. Esto se complementó con la paginación de las zonas de memoria ocupadas por el índice, de tal forma que al método de acceso le resultara más fácil la localización de las páginas del árbol en las que se encuentran las claves [Held 78, Folk 87].

El resultado de la aplicación de esta técnica de indexación es la existencia de un método de acceso ágil, que consume pocos recursos y que permite la gestión de claves de gran tamaño, lo que resulta tremendamente útil para manejar información textual.

El otro método de indexación muy difundido entre los sistemas de gestión de información textual es el llamado hashing. Este método cumple con una vieja aspiración de la informática clásica, poder realizar accesos directos a los datos. Los métodos que durante años se habían venido utilizando para indexar información eran indirectos, en el sentido de que no eran capaces de localizar una clave en un solo acceso, sino que debían realizar múltiples accesos hasta tener éxito en la localización. El método capaz de generar una determinada dirección de fichero dada una clave sin más información que la forma de la propia clave no apareció hasta que se diseñó el primer hashing. Este método tendrá la virtualidad de permitir accesos a los datos mucho más ágiles, con lo que las aplicaciones del tipo FMS podrán prestar servicios de recuperación de información en tiempo real hasta ese momento impensables. Es cierto que las diferencias en tiempos de acceso entre unos sistemas y otros pueden llegar a ser importantes, pero el tiempo de respuesta no depende únicamente de las características técnicas del método sino que hay otros factores como el tipo de información a indexar, el hardware utilizado, las necesidades de recuperación, etc., que pueden afectar más a los tiempos de respuesta [Leftkovitz 69, Standish 80, Bookstein 72].

Las ventajas fundamentales del hashing son:

- Utiliza poco espacio en disco para la generación de los índices.
- Necesita realizar pocos accesos al disco para localizar una clave.
- Gestiona muy rápidamente la inserción y eliminación de claves.

Su principal inconveniente es que el hashing es únicamente un método de acceso, no un método de ordenación y acceso, porque, como veremos después, no mantiene las claves ordenadas lexicográficamente porque para realizar los accesos en tan corto espacio de tiempo necesita alterar esta ordenación, que resulta más propia de sistemas de indexación como los basados en árboles [Knuth 73, Salton 88, Bookstein 74].

La operativa del hashing consiste en transformar una clave en una dirección mediante una función. La función de hashing fh transforma la clave C en una dirección del fichero de índice en la que se almacenará un puntero que señala la dirección del registro del fichero de datos donde se encuentra la clave C.

fh(C) = posición en el índice

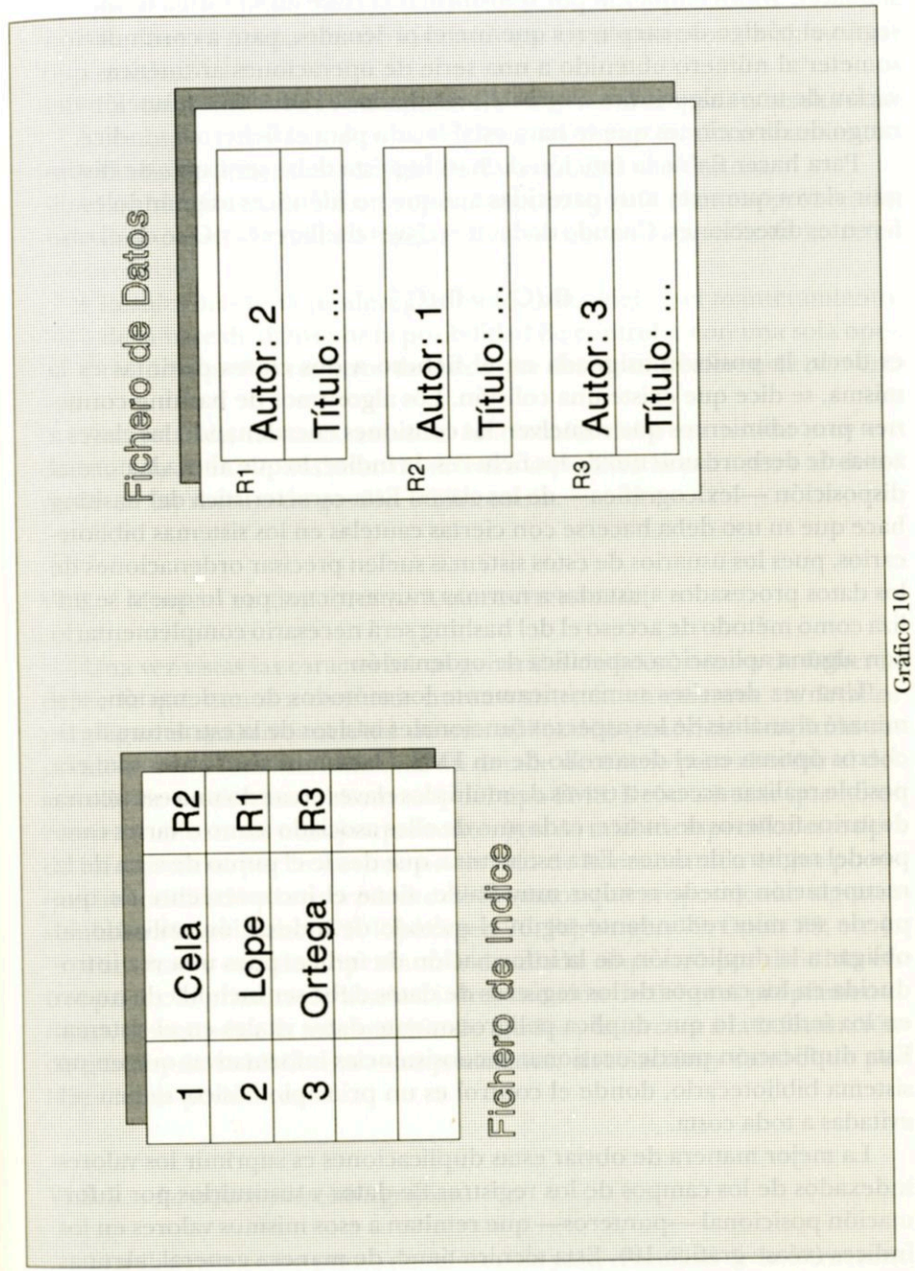


Gráfico 10

Los algoritmos de hashing incluyen funciones basadas en técnicas muy similares. Todos empiezan por transformar la clave en su forma numérica según el código de caracteres que use el ordenador, para a continuación someter al número obtenido a una serie de operaciones aritméticas que varían de unos algoritmos a otros. El resultado se reducirá en función del rango de direcciones que se haya establecido para el fichero de índice.

Para hacer fiable la función de hashing ésta debe ser capaz de distinguir claves que sean muy parecidas aunque no idénticas asignándoles diferentes direcciones. Cuando dadas dos claves similares C_i y C_j se da el caso

$$fh(C_i) = fh(C_j),$$

es decir, la posición asignada en el fichero a dos claves distintas es la misma, se dice que existe una colisión. Los algoritmos de hashing contienen procedimientos que resuelven las colisiones encadenando las claves a zonas de desbordamiento de los ficheros de índice, lo que altera la normal disposición —lexicográfica— de las claves. Esta característica del hashing hace que su uso deba hacerse con ciertas cautelas en los sistemas bibliotecarios, pues los usuarios de estos sistemas suelen precisar ordenaciones de los datos procesados ajustadas a normas muy estrictas, por lo que si se utiliza como método de acceso el del hashing será necesario complementarlo con alguna aplicación específica de ordenación.

Una vez descritos sumarísticamente los métodos de ordenación, terminaré el análisis de los aspectos funcionales básicos de la estructura de ficheros óptima en el desarrollo de un FMS. Habíamos visto antes que era posible realizar accesos a través de múltiples claves creando una estructura de varios ficheros de índice, cada uno de ellos asociado a uno o varios campos del registro de datos. Esta estructura, que desde el punto de vista de la recuperación puede resultar muy fiable, tiene el inconveniente de que puede ser muy redundante según el método de indexación utilizado, al obligar a la duplicación de la información de índice, pues una vez introducida en los campos de los registros de datos debe ser incluida de nuevo en los índices, lo que duplica peligrosamente datos vitales en el sistema. Esta duplicación puede ocasionar inconsistencias informativas que en un sistema bibliotecario, donde el control es un principio básico, deben ser evitadas a toda costa.

La mejor manera de obviar estas duplicaciones es suprimir los valores indexados de los campos de los registros de datos y sustituirlos por información posicional —punteros— que remitan a esos mismos valores en los índices (véase gráfico 10). Esta técnica tiene, de manera general, algunas consecuencias:

- Evita la redundancia y con ello las posibles inconsistencias informativas.
- Reduce la ocupación de los datos en los dispositivos de almacenamiento.
- La visualización de la información y en general su gestión resulta más costosa en términos de acceso a los dispositivos de almacenamiento.
- El mantenimiento de la base de datos resulta más costosa.
- La abundancia de información posicional puede plantear problemas de inconsistencias informativas por errores en la gestión de las referencias.
- Resulta más fácil, desde un punto de vista lógico, el mantenimiento de la base de datos por la posibilidad de controlar con una sola operación aquellos registros de datos que tienen una referencia común.

Esta relación de características funcionales debe ser tenida muy en cuenta a la hora de trasladar de manera específica la estructura al desarrollo de un sistema de gestión bibliotecaria.

VI.B. Modelo de estructura y análisis funcional

Una vez vistas las características generales de la estructura de ficheros desde la que vamos a construir nuestro sistema, entraremos en detalles para ligar el análisis funcional de una aplicación genérica —capítulo segundo— con el análisis funcional en detalle necesario para abordar el desarrollo de un sistema bibliotecario del tipo FMS. Debo advertir que no es corriente encontrar trabajos que contengan de manera específica este tipo de información, por lo que me he servido de la encontrada en los pocos que he localizado y la procedente de los datos que proporcionan algunos suministradores [Salton 75, Rijsbergen 76, Rijsbergen 79, Dobis 85, etc.].

Para hacer esta descripción más detallada empezaré por dividirla en tres partes: la primera dedicada a la información de ficheros propiamente dicha, la segunda a la información de fondos y la tercera a la información de gestión de la biblioteca —adquisiciones, circulación y control de publicaciones periódicas—.

VI.B.1. *La información bibliográfica*

Si utilizamos como punto de referencia para el análisis de la estructura de la información bibliográfica la normativa internacional que la regula [Usmarc 88, Crawford 84, Tannehill 82], habrá que convenir que tras su

complejidad aparente se esconde un entramado informativo que divide todos los elementos que la integran en dos grandes grupos:

- Información de punto de acceso-indexable.
- Cuerpo de la descripción bibliográfica-no indexable.

Este dato nos lleva de entrada a establecer que en cada referencia de un catálogo habrá un conjunto de datos que deben ser indexados y otro conjunto formado por información a través de la cual no se necesita acceder a las referencias. Si utilizamos terminología MARC esta distinción se nos muestra especialmente clara. Existen una serie de bloques funcionales, descritos en los manuales de referencia del formato, que han sido definidos expresamente para albergar la información de punto de acceso:

1XX - Encabezamientos principales
6XX - Encabezamientos de materias
7XX - Encabezamientos secundarios
8XX - Encabezamientos secundarios de serie

Aunque éstos son los campos catalográficos que básicamente recogen la información de punto de acceso, es lógico pensar que el diseñador de un sistema de gestión de catálogo en línea pretenda dar opción de recuperación a los usuarios a través de campos no recogidos entre éstos y sin embargo necesarios para realizar una adecuada gestión de las búsquedas:

020 - ISBN/ISSN
080 - CDU
24X - Títulos
260 - Editores
4XX - Títulos de series

Una vez seleccionados todos los campos MARC cuya información debe ser indexada, podremos proceder a determinar qué índices crear y qué campos deben asociarse a cada índice para su actualización. Para realizar esta tarea será preciso establecer de antemano si es posible realizar agrupaciones de campos por la similitud de los valores informativos que contienen y a partir de aquí establecer los ficheros que se deben crear. Alguna clave en este sentido nos dan otras normas [Gare 84, Usaut 87] al establecer el concepto y la tipología de las autoridades bibliográficas. Las GARE consideran que los nombres de personas e instituciones, los títulos uniformes y, en otro orden, las materias, son las entradas que deben ser consideradas como de autoridades. Como se puede ver, con esta pequeña clasificación no se abarca todo el conjunto de campos reseñados anteriormente, pero sí que se hacen unas agrupaciones de campos diferentes de las de los bloques funcionales, que resultan muy útiles para definir los ficheros de índice. Así todos los campos definidos para contener nombres de personas —X00 en formato MARC— tanto si son encabezamientos principales o secundarios pertenecen al mismo tipo de entrada y desde el punto de vista del control es conveniente que estén juntos para garantizar su consulta en el momento de la validación de una nueva entrada. Esta necesidad de realizar un control en línea de las entradas de punto de acceso es la causa de que haya que hacer un minucioso análisis de los campos para garantizar la normalización de dichas entradas y con ello la calidad de las recuperaciones. Imaginemos por un momento que tratamos de añadir un nombre de autor que previamente ha sido introducido en el catálogo como prologuista de otra obra, por lo que se encuentra en un campo 700 y es un encabezamiento secundario. La forma del nombre introducida previamente debe ser utilizada ahora para cumplimentar un campo 100 —encabezamiento principal de autor personal—, pues de lo contrario podríamos tener en el catálogo formas distintas del mismo autor. Sin embargo, es preciso considerar otro aspecto práctico del problema antes de dar por resuelta la cuestión. La composición de subcampos especificada para los distintos campos X00 existentes en MARC no es coincidente, y lo mismo ocurre con los campos X10 —autores corporativos— y X11 —congresos—. Si utilizamos el ejemplo anterior encontraremos que en el caso del autor como prologuista junto al subcampo a que contiene el nombre del autor en forma invertida aparece un subcampo e que contiene de forma abreviada la relación entre el autor y la obra —en este caso pr— Si indexamos la totalidad del campo, cuando intentemos relacionar esta entrada de índice con un registro perteneciente a una obra en la que el autor en cuestión tenga una responsabilidad distinta, la información relativa a la relación sería incorrecta. Esto nos obliga a determinar no sólo qué campos deben ser indexados y en qué ficheros, sino qué subcampos dentro de cada campo deben ser indexados. Una propuesta en este sentido para el caso de las monografías podría ser la siguiente:

- Índice de nombres:

100 a
110 a
111 a
700 a
710 a

711 a
800 a
810 a
811 a

- Índice de títulos:

130 a
240 a
245 a
440 a
490 a
730 a
740 a
830 a
840 a

- Índice de materias:

6XX todos

- Índice de editores:

260 b

- Índice de ISBN:

020 a

- Índice de CDU:

080 a

Aunque esta relación no pretende ser exhaustiva sino tan sólo suficiente para la descripción de la estructura de ficheros, lo que sí considero necesario resaltar es que las posibilidades de recuperación de la información bibliográfica que permite una relación de índices como ésta son las suficientes desde la perspectiva de las normas de descripción bibliográfica [Isbd 77]. En cualquier caso, habida cuenta que mi objetivo aquí es definir una estructura, cualquier otro campo o conjunto de campos que se necesitara indexar por necesidades de recuperación podría añadirse a los anteriores. El procedimiento está definido.

Me gustaría hacer ahora algunas observaciones sobre los índices descritos anteriormente. Como se puede observar, junto a la denominación del índice se ha consignado la relación de campos y subcampos que lo integran. En el caso del índice de nombres se han incluido en él los mismos contenidos que la LC asigna al fichero de autoridades de nombres —auto-

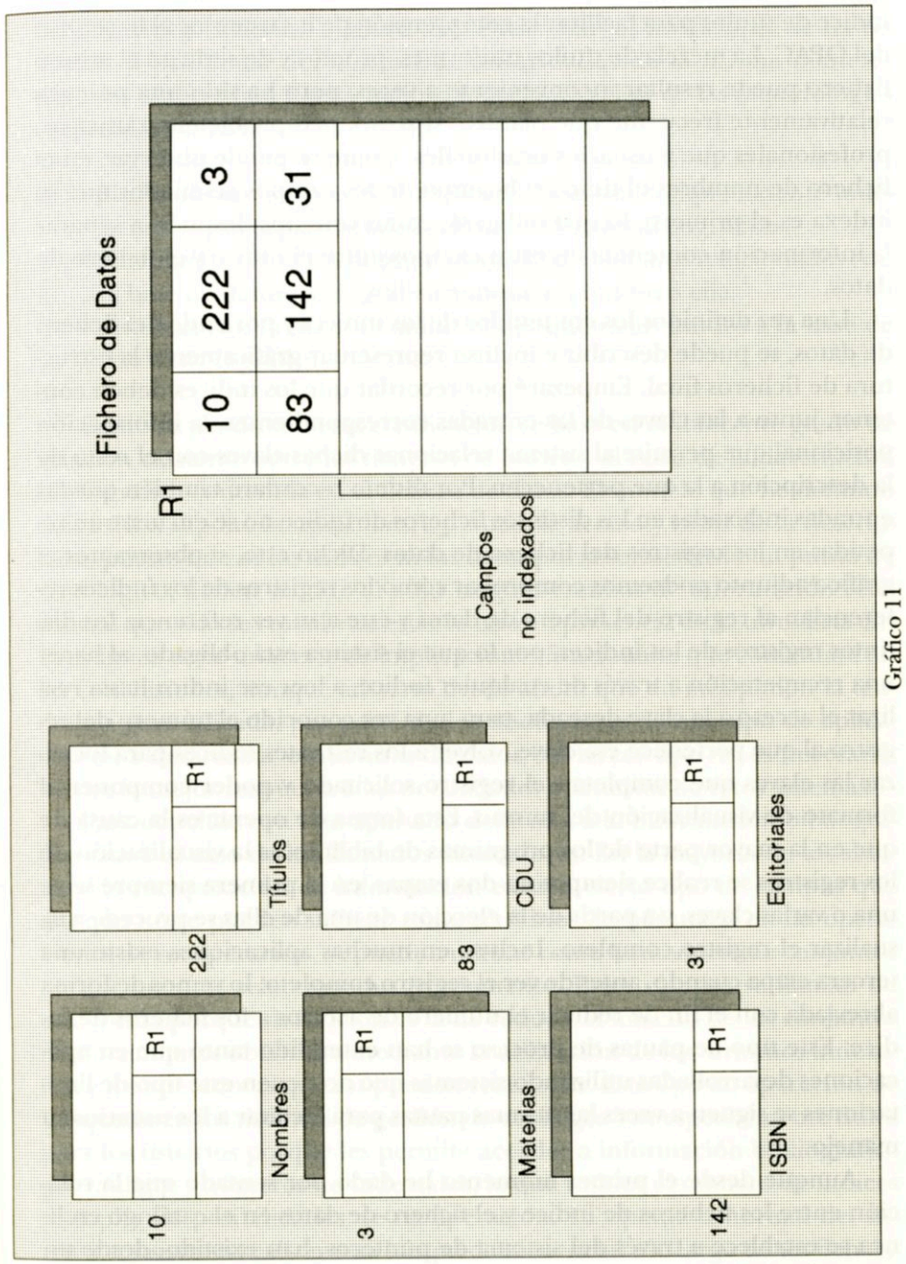


Gráfico 11

res personales, corporativos, nombres de congresos— con la única excepción de los títulos uniformes y las series, que han sido incluidos en el índice de títulos para facilitar la comprensión de los usuarios al hacer uso del OPAC. La mezcla de títulos uniformes, propios y de serie en el mismo fichero puede resultar inconveniente a veces, pero ha sido una práctica relativamente frecuente que plantea, si acaso, más problemas a usuarios profesionales que a usuarios ocasionales. Como se puede observar, en el fichero de nombres el único subcampo de los campos reseñados que se indexa es el primero, lo que obligará, como veremos después, a separar la información contenida en estos campos entre el índice y el fichero de datos.

Una vez definidos los contenidos de los índices y por ende del fichero de datos, se puede describir e incluso representar gráficamente la estructura de ficheros final. Empezaré por recordar que los índices deben contener, junto a las claves de las entradas correspondientes, la información posicional que permite al sistema relacionar dichas claves con el resto de la descripción a la que pertenecen. Por último recordaré también que las entradas indexadas en los distintos ficheros de índice no se encuentran repetidas en los registros del fichero de datos. Dicho esto, si observamos el gráfico adjunto podremos comprobar cómo los registros de los índices referencian al registro del fichero de datos y éste a su vez referencia los distintos registros de los índices, por lo que el sistema está obligado, al hacer una recuperación a través de cualquier índice, a leer ese índice hasta realizar el acceso a la clave deseada, para, una vez conocido el número del registro al que pertenece esa clave, volver a los restantes índices para localizar las claves que completan el registro solicitado y poder componer el formato de visualización del mismo. Esta forma de operar es la causa de que en la mayor parte de los programas de bibliotecas la visualización de los registros se realice siempre en dos etapas, en la primera siempre se ve una o varias claves, y a partir de la elección de una de ellas se procede a visualizar el registro completo. Incluso en muchas aplicaciones existe una tercera etapa cuando, antes de ver el registro completo, lo vemos de forma abreviada con el fin de reducir el número de accesos a los ficheros de índice. Este tipo de pautas de proceso se han extendido tanto que en aplicaciones desarrolladas utilizando sistemas que no tienen este tipo de limitaciones se siguen a veces las mismas pautas para facilitar a los usuarios su manejo.

Aunque desde el primer momento he dado por sentado que la relación entre los ficheros de índice y el fichero de datos en el catálogo en línea se establece a través del sistema de punteros, han existido, desde un punto de vista lógico, diferentes formas de relación entre unos y otro. En realidad la forma de relación condiciona de manera definitiva el tipo de

control de autoridades posible, puesto que las llamadas autoridades bibliográficas están depositadas en tres de estos ficheros de índice —nombres, títulos y materias— [Gare 84]. Sin embargo, no siempre se ha resuelto el problema del control de autoridades de la misma manera. Algunos autores han hecho incluso una clasificación de las metodologías empleadas [O'Neill 88, Burger 85]:

- Sistemas con ficheros de autoridades que son completamente independientes y separados de las bases de datos catalográficas.
- Sistemas con ficheros de autoridades que están muy relacionados con la base de datos catalográfica aunque no unidos a ella.
- Sistemas con ficheros de autoridades que están unidos a la base de datos catalográfica.

Como se puede ver, esta clasificación se basa en el diferente grado de relación existente entre los ficheros de índice y el resto de la descripción bibliográfica contenida en el fichero de datos. El modelo que he definido más arriba supone la unión entre puntos de acceso y resto de la descripción, puesto que ni los índices ni los datos por sí solos, desde un punto de vista lógico, tienen sentido al estar cargados ambos de información referencial que sólo se resuelve al estar lo uno en presencia de lo otro. Esta circunstancia tiene una serie de implicaciones funcionales que comentaré en las conclusiones, aunque advierto desde ahora que en mi opinión la tendencia generalizada, por parte de los diseñadores de este tipo de aplicaciones, a desarrollar estructuras de datos del tercer tipo no es necesariamente la mejor respuesta a las necesidades operativas de estos sistemas.

Para concluir con este apartado dedicado a la información bibliográfica citaré el caso de los llamados limitadores. En la mayoría de los sistemas de recuperación de información existen cierto tipo de campos que aunque no se pueden utilizar como términos de búsqueda en las operaciones de recuperación, se pueden utilizar como limitadores de conjuntos de documentos seleccionados previamente. Estos campos se llaman de esta forma porque su función es la de reducir el tamaño de los conjuntos de documentos obtenidos mediante la utilización de los términos indexados y los operadores que los pueden relacionar. Este tipo de operación de búsqueda en los sistemas de gestión de catálogos son especialmente útiles para los usuarios porque les permite acceder a información bibliográfica por códigos y números no indexados, lo que reduce el número de índices y amplía las posibilidades de recuperación. La operativa normal del sistema consiste en realizar una búsqueda secuencial a lo largo de un fichero virtual formado por el conjunto de documentos seleccionado anteriormente. Este procedimiento, aunque flexible, siempre plantea el problema

de que es tanto más lento cuando que el conjunto de documentos es mayor. Por otra parte, la operativa normal en el proceso secuencial debe limitarse a la comprobación de la presencia o ausencia de la información —normalmente codificada— que se solicita, por lo que este sistema no debe utilizarse con informaciones textuales. Las informaciones que de forma más usual han sido objeto de este tipo de tratamiento son: el código de tipo de material, los códigos de país y lengua de la publicación, y las fechas, éstas últimas sujetas a un tratamiento especial, pues la limitación aquí consiste normalmente en uno de entre cuatro tipos de operaciones aritméticas, igual que (=), mayor que (>), menor que (<) y entre (><). En definitiva los campos de limitación compensaban la ausencia de índices en algunos casos y mejoraban siempre las posibilidades de recuperación de la aplicación.

VI.B.2. *La información de fondos*

El control de la información bibliográfica propiamente dicha se completa en un catálogo en línea con el control de la información relativa a los fondos que cada centro bibliotecario tiene de los documentos descritos en su catálogo. El control de esta información, que es esencialmente topográfica, aunque no exclusivamente [Ushold 89], tiene como principal dificultad que la aplicación debe poder asociar múltiples registros de fondos a cada registro catalográfico, mientras que se facilita el acceso a través de índices de determinadas informaciones topográficas —signaturas y códigos de sección o branch—.

Esta necesidad de recuperación se resuelve por la misma vía que en el caso anterior, creando los índices necesarios asociados mediante punteros al registro de fondos que estará alojado en un fichero de datos que, a su vez, también se relacionará con los índices mediante información posicional. Hasta aquí la estructura es muy similar a la descrita en el caso de la información bibliográfica, pero de nada serviría este nuevo conjunto de ficheros si no se pudieran conectar con los anteriores. Los ficheros de datos que contienen información bibliográfica y de fondos deben mantener conectados entre sí sus registros a fin de que sea posible ir en la recuperación de los datos topográficos a los bibliográficos —de la signatura al título de la obra—, y de los datos bibliográficos a los topográficos —del autor a la ubicación de sus obras—. Esto da lugar a una estructura algo más compleja pero que utiliza los mismos recursos de gestión de datos, por lo que su mantenimiento tiene similares dificultades. Quizá convenga matizar una última cuestión en relación con las similitudes entre los registros bibliográficos y los de fondos. Resulta evidente que un título puede tener

diversos registros de fondos asociados en la medida en que puede haber varios ejemplares del mismo, mientras que a cada registro de fondo sólo le corresponde un título y por tanto un registro bibliográfico. Esta lógica de los datos se traduce en la existencia en los registros bibliográficos de listas de punteros que referencian los registros de fondos en los que sólo existe un puntero que les relaciona con el registro de título correspondiente.

VI.B.3. *La información de gestión*

Muy brevemente y para terminar este apartado me parece necesario hacer alguna referencia a la estructura de ficheros utilizada en los FMS para procesar la información relacionada con las funciones básicas de adquisición, circulación y control de las publicaciones periódicas.

Como ya comenté en el capítulo segundo de este trabajo, la información que, de forma poco propia, se denomina de gestión de la biblioteca está formada por datos que tienen una estructura muy definida, por lo que resulta sencillo adaptar la estructura física a la lógica de los datos. Registros como el que contiene los datos de un pedido, un préstamo o una subscripción siguen un esquema fácil de implementar utilizando los recursos definidos en apartados anteriores —ficheros de índice y de datos—. No entraré en detalles aquí para describir una estructura que físicamente es idéntica a las ya descritas y, lógicamente, por no existir formatos estándar para el procesamiento de estas informaciones las estructuras definidas en las diferentes aplicaciones dependen de las subfunciones que cada aplicación pretenda mecanizar. En definitiva, la estructura de ficheros utilizada en esta parte de la base de datos es perfectamente compatible con el resto, lo que facilita la necesidad de conectar en ocasiones informaciones bibliográficas o de fondos con datos de las funciones básicas de gestión como ya quedó especificado en el análisis funcional de la aplicación bibliotecaria.

VI.C. Conclusiones

a) La recuperación de información posibilitada por estructuras de datos como las que utiliza un FMS tiene dos características fundamentales:

   1. Los accesos se realizan en base a la comprobación de la igualdad entre el término de búsqueda y las claves de izquierda a derecha con truncado implícito o explícito —exact match—.

Esto significa que las técnicas de recuperación basadas en la ponderación de las entradas, en la similaridad o en procesos de retroalimentación de las búsquedas —partial match— no son posibles con esta estructura de datos. En realidad las posibilidades de recuperación de estos sistemas, definidos por algunos como de primera generación [Hilldreth 85, Hilldreth 87], ni siquiera llegan a la recuperación por palabras clave, lo que en el caso de sistemas bibliotecarios resulta especialmente grave. La causa de estas limitaciones es estructural, por lo que habría que introducir cambios importantes en la estructura de la base de datos para facilitar algunas de las funciones de recuperación que acabo de mencionar.

2. La generación de conjuntos de documentos a partir de una búsqueda como paso previo a la utilización de un operador booleano se ve dificultada por la estructura de datos. La generación de un conjunto de documentos sólo es posible mediante la realización de un recorrido secuencial a través del fichero de índice, puesto que los métodos de acceso descritos sólo permiten la localización de la primera clave idéntica al término de búsqueda introducido. Si a partir de esta primera clave existen otras que cumplen la condición especificada será necesario que el sistema proceda secuencialmente. Esto hace que los tiempos de respuesta en operaciones booleanas sean inversamente proporcionales al tamaño de los conjuntos de documentos generados. En realidad la simulación de funciones de recuperación del tipo IRS por sistemas FMS siempre ha tropezado con el problema de la inadecuación de la estructura de datos.

b) Al no existir en los índices más información posicional que la que relaciona cada entrada con el registro de datos al que pertenece, no es posible tampoco la utilización de operadores de proximidad. La búsqueda a través de campos como los de títulos o resumen se ve seriamente perjudicada por no existir índices de palabras clave y, por consiguiente, no se podrán utilizar otros términos de búsqueda que no sean los de comienzo de las distintas entradas. Esto al mismo tiempo perjudicará la recuperación a través de campos como el de materias en el que la limitación de los conjuntos de documentos en función del valor de los subencabezamientos es imprescindible y sin embargo imposible con un sistema de indexación que carezca de permutaciones por términos clave.

c) Una de las críticas que con más insistencia se han vertido sobre los sistemas del tipo FMS ha sido la de su dependencia del hardware y software bajo el que funcionan [Priwes 72]. Esta crítica tiene su origen en las dificultades de mantenimiento que plantean aplicaciones que no disponen de más recursos de mantenimiento de las estructuras de datos que los que los propios diseñadores de la aplicación tengan previstos. Esta dependencia hace necesario resolver por métodos casi artesanales las inconsistencias de datos que puedan darse en la base de datos, lo que hace imprescindible la existencia de personal experto no sólo en la fase de desarrollo sino también en la fase de explotación atendiendo a los usuarios.

d) Las aplicaciones resultantes de la utilización de este sistema son generalmente muy amigables y adaptadas a las condiciones de uso del sistema por parte de los usuarios finales. Esto tiene como contrapartida esencial la dificultad de adaptación de las aplicaciones a las circunstancias funcionales y normativas cambiantes en los entornos bibliotecarios. Esta dificultad de adaptación ha hecho que el grado de obsolescencia de este tipo de aplicaciones sea muy alto. Quizá uno de los aspectos que más acelera esta necesidad de adaptación constante es la renovación tan acelerada a la que se somete a los estándares bibliotecarios de descripción. Al producirse nuevas versiones de los formatos de descripción se hace necesario introducir modificaciones en las aplicaciones, que en el caso de los FMS afectan a la estructura básica de los datos lo que pueden suponer cambios de costosa ejecución por la dificultad de adaptación de los programas en funcionamiento.

e) Este conjunto de problemas ha hecho que las aplicaciones de este tipo hayan ido desapareciendo y en la actualidad no se desarrollen otras de las mismas características. A pesar de todo la estructura de datos que este tipo de aplicaciones desarrolló se sigue utilizando con recursos de programación nuevos —DBMS—. Incluso se han añadido a estas nuevas aplicaciones funciones IRS para mejorar sus posibilidades de recuperación. La persistencia de estas estructuras de datos se debe a lo que es su valor fundamental: la integridad de la información que contienen. En sistemas como los bibliotecarios, que precisan de un gran control en las entradas de datos, es importante que la aplicación garantice la no repetibilidad de la información y, por tanto, su integridad. Esto incluso a costa de que los usuarios profesionales padezcan las dificultades antes descritas.

VII

LOS SISTEMAS HÍBRIDOS

El objeto de este capítulo es hacer algunas propuestas sobre la estructura de datos que más se pueda adecuar a las mejoras funcionales que en materia de recuperación de información bibliográfica es posible introducir hoy en los sistemas bibliotecarios. El punto de partida será la definición de sistema híbrido como soporte de software básico ideal para desarrollar la estructura de datos adecuada a las exigencias de proceso que las nuevas funciones plantean. Para realizar estas propuestas empezaré por revisar algunas de las críticas que con más insistencias han aparecido en los trabajos científicos durante los últimos años sobre las posibilidades de los sistemas basados en el álgebra de Boole. A continuación pasaré revista a algunas de las propuestas técnicas que los investigadores han hecho para mejorar los métodos boolean matching, tratando de probar al mismo tiempo su funcionamiento automático con datos reales para demostrar la viabilidad de su implementación fuera del laboratorio. En esta fase haré algunas sugerencias de modificación de los métodos propuestos, con el fin de adaptarlos mejor a las características de la información catalográfica. Por último, como en capítulos anteriores, haré una valoración general del sistema resultante.

VII.A. Concepto de sistema híbrido

Durante años muchos investigadores han planteado la necesidad de desarrollar sistemas capaces de gestionar con eficacia información estructurada e información textual [Eastman 85, Kemp 88, Reid 90, etc.]. Este tipo de sistemas multivalentes han recibido diversos nombres, quizá uno de los más aceptados ha sido el de DBMIRS —Data Base Management Information Retrieval System— [Schek 81], que mezcla los nombres de los sistemas cuyas funciones combinadas pretende poseer. Sin embargo, la denominación que yo utilizo aquí es la que se ha aplicado a los sistemas bibliotecarios que combinaban los efectos de los IRS y RDBMS, normalmente para aplicar uno a la gestión de la información bibliográfica y el

otro a la gestión de la información estructurada [Kemp 88]. Aunque esta cuestión del nombre no es un tema trascendente en sí mismo, nos pone en situación de plantear una cierta tipología de sistemas híbridos puesto que en ocasiones parecen haber sido el resultado de una mera yuxtaposición de funciones, mientras que otras veces responden a un deliberado intento de dotar a un RDBMS de funciones de IRS o viceversa. En consecuencia existen en mi opinión tres tipos de sistemas híbridos:

- Sistemas basados en la suma de funciones de un IRS y un RDBMS mediante el desarrollo de un software que hace de puente entre uno y otro. En estos casos las aplicaciones desarrolladas ejecutan procesos independientes según que la información tratada sea estructurada o textual, pero en ningún momento en el mismo proceso se gestionan los dos tipos de información [Hopkinson 77, Hickey 89]. No se plantearían problemas de integración entre un RDBMS y un IRS si funciones básicas de la gestión bibliotecaria como la catalogación y la circulación pudieran funcionar de manera independiente, es decir, si sus ficheros no contuvieran información que debe estar relacionada entre si (véase capítulo segundo). Es justamente esta necesidad de constante relación entre la base de datos bibliográfica gestionada por el IRS y las tablas que procesa el RDBMS lo que causa más dificultades de desarrollo en un sistema híbrido de este tipo. La solución suele venir dada por las propias herramientas de desarrollo con las que los sistemas base están dotados [Ftrs 89], lo que convierte a la elección de los productos básicos en la clave del éxito del diseño del sistema.

- Una de las fórmulas más estudiadas en los últimos años para el desarrollo de sistemas híbridos es la que consiste en ampliar funcionalmente los RDBMS, dotándolos de las capacidades necesarias para gestionar información textual [Lynch 87, Sheck 81, Macleod 91]. Pero la solución más extendida es la que consiste en diseñar estructuras de datos que permiten ciertos tratamientos textuales a base de utilizar relaciones no normalizadas [Macleod 90, Crawford 81, Lynch 91, Blair 88, Deogun 88]. En ambos casos nos encontramos con ventajas tales como la de facilitar un alto grado de productividad en el desarrollo de aplicaciones, mayor portabilidad de las mismas, así como menor esfuerzo de mantenimiento de las bases de datos.

- El último tipo es el menos extendido y se genera dotando a los IRS de las funcionalidades necesarias para gestionar información estructurada. El caso más conocido de este modelo es el del producto STAIRS [Stairs] que incluye no sólo funciones de recuperación textual (search) sino funciones de procesamiento de información altamente estructurada (select). Este sistema, y otros [Dobosz 81], aunque están orientados a la recuperación de información organizada jerárquicamente —documento, párrafo, frase, palabra— puede combinar sus funciones más características son compatibles con los RDBMS, pero siempre por el procedimiento de conmutación de modos operativos, lo que supone en el fondo la convivencia de dos subentornos en uno. En cierto sentido se puede decir que este modelo es más similar al primero que el anterior.

Existen algunas constantes en los trabajos de investigación que vengo comentando que me gustaría señalar aquí. La mayoría de estos trabajos ponen de manifiesto la necesidad que existe de disponer de sistemas capaces de gestionar de forma suficientemente integrada información estructurada y textual. Esta necesidad ha sido especialmente patente a partir del momento en que empezaron a aparecer los primeros DBMS, pues las ventajas operativas y funcionales que estos sistemas suponen respecto de las herramientas de desarrollo convencionales han sido el acicate para que se estudiaran soluciones en esta línea. Resultan, por tanto, este tipo de propuestas recientes de la investigación cruciales para la solución del problema básico que me planteé al comienzo de este trabajo: qué tipo de estructura de datos debe usarse para gestionar de forma integrada un conjunto de información que en parte es textual y en parte es información altamente estructurada, o, dicho de otra forma, qué tipo de sistema de gestión de información es el idóneo para realizar la gestión de los flujos informativos que se generan en un centro bibliotecario. Como se puede ver y hasta donde la investigación ha llegado, los sistemas híbridos son la respuesta a la pregunta inicial. Pero estos sistemas mejoran las prestaciones de los gestores de bibliotecas en la medida en que se adaptan mejor a las características de la información que tienen que gestionar, pero no aportan mejoras funcionales sustanciales a estas aplicaciones. En realidad las ventajas de la utilización de los sistemas híbridos en el desarrollo de gestores bibliotecarios son básicamente de prestaciones.

Uno de los ejemplos mas interesantes de lo que acabo de exponer es el proyecto XLS —«Experimental Library System»— de la organización OCLC. En este proyecto se han ligado un DBMS y un IRS para desarrollar en un entorno WIMP un gestor de bibliotecas funcionalmente bastante completo [Hickey 89, Hickey 90]. Según los propios desarrolladores la dificultad fundamental con la que tropezaron en su trabajo fue la de conectar, sin producir repeticiones, las referencias catalográficas con la in-

formación tabular del DBMS. Aunque el resultado satisfizo a sus creadores, el hecho de que la OCLC cancelara el proyecto antes de su conclusión imposibilita la comprobación de su viabilidad. Lo curioso del caso es que el otro gran proyecto —Merlin— de estas mismas características del que existe noticia [Hopkinson 77] tampoco llegó a terminarse debido a su cancelación presupuestaria por parte de la British Library. Sin embargo en ambos casos se logró el desarrollo de prototipos capaces de realizar la gestión integrada de ambos tipos informativos.

Pero a lo largo del trabajo se ha ido deslizando una cuestión que ha quedado planteada y sin respuesta. Si bien es cierto que los sistemas híbridos desde un punto de vista lógico parecen ser la mejor solución para el desarrollo de aplicaciones de gestión bibliotecaria, también lo es el hecho de que es necesario aceptar al menos dos premisas para mantener la afirmación inicial:

a) que funcionalmente, desde el punto de vista de la recuperación de información, lo más que le vamos a pedir a la aplicación son prestaciones del tipo boolean matching.

b) que los tipos informativos procesados por la aplicación serán el estructurado y el textual.

Admitidas estas dos premisas podríamos decir que la búsqueda del modelo de gestión habría terminado. Pero, como he comentado en las conclusiones de los análisis de los modelos estudiados, las nuevas técnicas de recuperación de información —métodos ponderativos— exigen, por ejemplo, la presencia de frecuencias de uso de términos de indización para efectuar las ponderaciones, así como la realización de cálculos en ocasiones complejos para los que la mayoría de los sistemas actuales no están preparados.

La determinación de las características que debe tener un sistema capaz de afrontar la recuperación a través de estas nuevas técnicas de recuperación es el objeto de lo que sigue. Para ello será necesario empezar por hacer un resumen de las más relevantes críticas hechas en los últimos años al modelo de recuperación del boolean matching.

VII.B. La superación de Boole

Las técnicas de recuperación de información utilizadas hoy en los sistemas bibliotecarios se basan en la denominada indización binaria, es decir, los términos de indización o están o no están en cada documento [Larson 92]. No existe por tanto ninguna posibilidad de ponderación de la presencia del término en el documento. Por otra parte, las operaciones realizadas con los términos de indización en el proceso de recuperación son las operaciones de conjuntos del álgebra de Boole —AND, OR, NOT—.

En otro orden de cosas, las tareas de recuperación realizadas por los usuarios en los sistemas bibliotecarios se pueden reunir en dos grandes apartados [Hancock 89, Hildreth 85, Cove 88]:

- La búsqueda específica —queriying—, que se realiza cuando el usuario conoce con cierta precisión lo que está buscando y puede representar su conocimiento del documento buscado en términos asequibles a la base de datos catalográfica, lo que dará como respuesta la representación por parte del sistema de la referencia del documento solicitado. Estos términos de representación pueden ser expresiones o unitérminos, conectables en ambos casos por medio de operadores booleanos o de proximidad.

- El browsing de los ficheros de punto de acceso. Esta es una forma de búsqueda idónea cuando el objeto de la búsqueda no es específico o no se puede expresar con precisión. En este tipo de búsquedas el resultado no se puede anticipar debido a la imprecisión de los conocimientos previos del usuario. El hojeo de los ficheros se puede realizar de forma lineal o multidireccional. En el primer caso el usuario recorre un solo fichero de punto de acceso, mientras que en el segundo «navega» a través de los ficheros utilizando como puente entre unos y otros los propios registros recuperados.

Este marco que acabo de definir conforma las características esenciales de los llamados OPAC de segunda generación [Hildreth 89]. Y es precisamente en este entorno en el que se han desarrollado las críticas más severas por parte de los investigadores para tratar de justificar la introducción de cambios estructurales y funcionales en los sistemas de recuperación de información bibliotecarios.

Una de las primeras críticas fue la planteada a finales de los setenta. De manera muy general pero contundente se estableció una distinción terminológica según la cual a partir de entonces habría que diferenciar entre recuperación de datos —data retrieval— y recuperación de información —information retrieval— [Rijsbergen 79]. Aunque esta distinción estaba basada en la idea de que había diferencias conceptuales y prácticas profundas entre los métodos exact match, que se hacían coincidir con la recuperación de datos, y los métodos partial match, que se asociaban a la recuperación de información, lo cierto es que esta distinción ha tenido un

escaso éxito. En cualquier caso sirvió en su momento para poner de relieve algunos de los problemas que los métodos de recuperación de información tradicionales planteaban:

- Inferencia deductiva frente a inductiva: los sistemas tradicionales son deterministas mientras que los nuevos métodos deberían ser probabilísticos. Las respuestas en el primer caso son todas ciertas mientras que en el segundo caso se ordenan en función del grado de certidumbre.

- Igualación frente a relevancia: los sistemas tradicionales siguen el procedimiento de igualar los atributos de los documentos a los términos de búsqueda, mientras que las técnicas nuevas deben plantearse el problema de la relevancia de los documentos respecto de los términos de búsqueda, lo que permite establecer grados.

Esta crítica tan general fue el primer intento de sistematizar lo que ya en aquella época se consideraba como un proceso irreversible de superación de las técnicas basadas en el álgebra de Boole. Es importante recordar que este proceso que se empieza a teorizar a finales de los setenta comenzó, como casi todos los avances en documentación [Zunde 79], de manera empírica casi veinte años antes [Maron 60].

Ha sido especialmente en la década de los ochenta cuando, de manera mucho más sistemática, se ha procedido a criticar los aspectos más básicos del funcionamiento de los sistemas booleanos. Los autores más críticos han sido, lógicamente, algunos de los que más han trabajado en la investigación de la base teórica y los desarrollos prácticos de las nuevas técnicas [Bookstein 80, 85, Cooper 83, 88, Salton 84, 89, Belkin 87]:

- La lógica booleana no se adquiere por intuición sino que requiere formación por parte de los usuarios, a diferencia de lo que ocurre con el lenguaje natural.

- Los operadores booleanos son unas veces demasiado restrictivos —AND— y demasiado inclusivos otras —OR—. En el caso de la operación Y AND Z todos aquellos documentos que no contengan los términos Y y Z quedarán excluidos de la respuesta, con lo que no se admite la posibilidad de que documentos que contengan sólo el término Y o el término Z sean total o parcialmente relevantes para el usuario. Por el contrario, si la operación se plantea como Y OR Z, la respuesta puede incluir como primer documento del conjunto seleccionado uno que sólo tenga o el término Y o el Z, con lo que no se está asignando ninguna prioridad a los que contienen los dos términos solicitados. Esto, cuando se utiliza la disyunción, puede suponer un grave problema, pues los conjuntos de documentos seleccionados por esta vía suelen ser de gran tamaño.

- En relación con lo anterior también se ha argumentado que el álgebra de Boole es demasiado rígida en cuanto a las posibilidades de entrada y las de salida que ofrece a los usuarios. En las operaciones del párrafo anterior no se ofrecía la posibilidad al usuario de establecer si para él el término Y era más o menos importante en la búsqueda que el término Z y en qué proporción. En cuanto a la rigidez de la salida hay que decir que los documentos en los sistemas booleanos, en tanto que binarios, son recuperados o no recuperados, sin dar opción a la ordenación de la salida en función de su grado de relevancia de acuerdo con la adecuación de los valores de los atributos de los documentos con los términos de búsqueda y su ponderación.

- La incertidumbre y la parcialidad como características inherentes a los procesos de indización y recuperación parecen no existir en los sistemas booleanos, que suministran siempre respuestas ciertas y completas. Hasta tal punto esto es así, que estos sistemas nunca prevén la opción de rehacer las operaciones de búsqueda a partir de las respuestas obtenidas —feedback— con el fin de mejorar los resultados.

- Por último las críticas más duras han sido las relacionadas con la dificultad —imposibilidad en ocasiones— de transformar en operaciones booleanas las necesidades expresadas en lenguaje natural por parte de los usuarios. Esta limitación ha dado lugar al intento de desarrollo de interfaces para la transformación de expresiones del lenguaje natural en operaciones booleanas. En cualquier caso lo que no se ha podido resolver es la creación de una operación que busque cualquier par de términos de entre ocho por ejemplo, lo que demuestra que no sólo resulta difícil para usuarios expertos la generación de las operaciones, sino que, a veces, resulta incluso imposible.

En conclusión, en los sistemas booleanos cada documento recuperado es considerado por el sistema como de la misma utilidad para satisfacer las necesidades informativas de los usuarios. Dicho de otra forma, estos sistemas proporcionan a los usuarios conjuntos, en ocasiones enormes, de documentos sin ninguna ordenación ni indicación del grado de relevancia de los mismos. Lo que significa que sólo en el caso de que los usuarios sean

expertos en las materias de los documentos recuperados podrán servirse de la información proporcionada por el sistema, y esto sólo en el caso de que la búsqueda haya sido realizada por un profesional [Radecki 88].

Para terminar con este apartado de críticas a los sistemas booleanos en general me gustaría comentar algunas de las implicaciones que el uso de estos sistemas está teniendo en las bibliotecas. Pero antes quiero hacer notar que estas implicaciones no sólo son achacables a las características operativas del sistema utilizado, sino que por tratarse de conclusiones extraídas de investigaciones basadas en datos estadísticos de uso, se ven afectadas por todos los elementos que entran en juego en las operaciones de recuperación realizadas por los usuarios —características de los sistemas, comportamiento de los usuarios, características de la información, etc.—.

Desde comienzos de los ochenta se han venido preparando trabajos para analizar el uso que los usuarios realizaban de los OPAC en las bibliotecas. Casi todos estos trabajos han incidido más en el comportamiento de los sistemas cuando los usuarios consultaban a través de materias, debido a que más de la mitad de las búsquedas realizadas contra los catálogos se hacían a través del fichero de materias [Matthews 84]. Por otra parte, estos trabajos también han puesto de manifiesto que se producen de manera constante dos tipos de problemas en las recuperaciones:

- Búsquedas terminadas sin respuesta —search failure—: hasta casi un 50% de las búsquedas realizadas finalizan sin resultado, como consecuencia del desconocimiento por parte de los usuarios de la lógica booleana y/o los ficheros de punto de acceso, errores mecanográficos, etc. [Markey 84, 86, Yee 91].

- Búsquedas terminadas con exceso de información —information overload—: Cuando las bases de datos catalográficas han ido aumentando de tamaño y poniendo a disposición de los usuarios a través de los OPAC grandes cantidades de registros, se ha empezado a producir un fenómeno negativo para los usuarios. El sistema les ofrece como respuesta más información de la que ellos necesitan y de la que están en condiciones de analizar para su depuración [Larson 92, Markey 84, 86].

Esto ha traído como consecuencia fundamental la reducción del número de búsquedas por materias y exige rediseñar las capacidades de recuperación de los OPAC, de tal forma que desaparezcan las deficiencias expresadas [Larson 91].

A partir de aquí trataré de describir los elementos esenciales de un sistema de recuperación de información bibliográfica que mejore las prestaciones de los sistemas booleanos, utilizando algunas de las técnicas de recuperación desarrolladas por los investigadores de este campo junto con alguna modesta aportación personal.

VII.C. La ponderación de las entradas

Como he dicho en repetidas ocasiones hasta aquí, una de las carencias más acusadas de los sistemas booleanos es el hecho de que todos los términos de indización son considerados con el mismo valor. A esto lo han llamado algunos autores carácter binario del sistema. Cuando se realiza la indización de los documentos se produce un fenómeno que consiste en que cada término tiene valor 1 si está asociado al documento y 0 si no lo está. De esta forma, cuando intentamos recuperar los documentos asociados a dos términos distintos, el sistema nos entrega todos los documentos que tienen asociados esos dos términos con independencia del valor o la importancia en la búsqueda que nosotros asignemos a cada uno de ellos. Para el sistema los dos términos tienen la misma importancia, 1, y el resto de los términos de la base de datos, por no pertenecer al conjunto de los solicitados, tendrán en ese momento valor 0.

Esta argumentación puede servir de preámbulo para establecer algunos de los principios teóricos que han regido el desarrollo de los algoritmos de ponderación, que son la base de los estudios de indización automática desarrollados hasta ahora.

En los años cuarenta y en el marco de los entonces incipientes estudios de psicolingüística, Zipf estableció que si ordenamos un conjunto de palabras diferentes pertenecientes al mismo corpus textual de forma que decrezca su frecuencia de aparición en dicho corpus, y multiplicamos cada frecuencia por su rango en la ordenación obtenemos unos valores que son próximos a una constante [Zipf 49]:

frecuencia * rango constante

Esto quiere decir que la frecuencia de aparición de cualquier término multiplicada por su rango da un valor aproximadamente igual, lo que se puede probar con suma facilidad en el lenguaje natural en cualquier lengua [Kucera 67, Juilland 64]. Si representamos gráficamente la constante de Zipf para un conjunto de materias de un catálogo dado obtendremos una campana que representa cómo las frecuencias extremas se apartan del valor de la constante mientras que las intermedias se acercan (véase el grá-

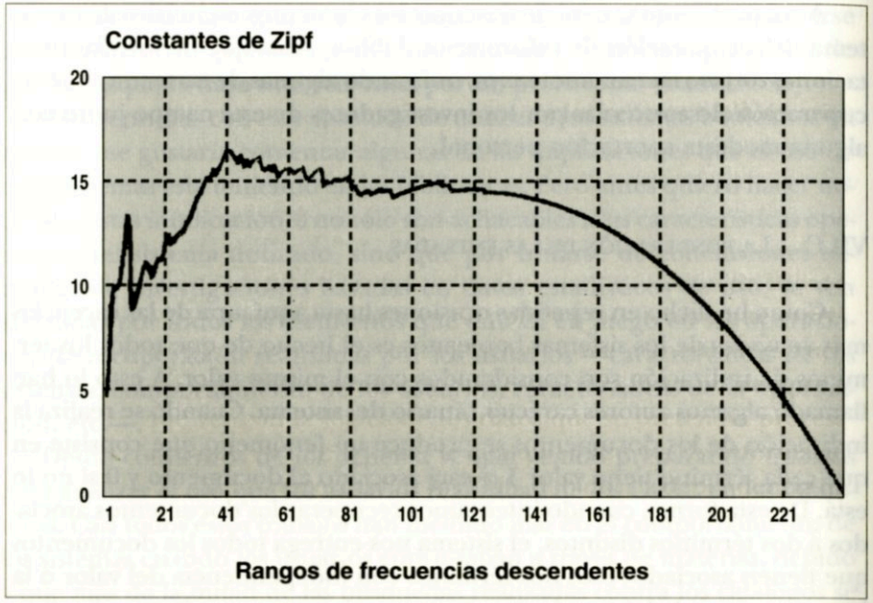


Gráfico 12. *Ley de Zipf*

fico adjunto). Lo que demuestra en primera instancia que la llamada ley de Zipf se cumple también en el caso de términos de indización en lenguaje controlado. Pero la conclusión más importante que se puede extraer ya de las ideas de Zipf es que existe cierta relación entre la frecuencia de utilización de las entradas y la importancia que éstas tienen de cara a la representación del contenido de los documentos a los que están asociados. La ordenación descendente de las frecuencias pone enseguida de manifiesto que las primeras entradas son las que tienen menos importancia en la representación de los contenidos, aunque será necesario desarrollar esta idea algo más para poder extraer de ella conclusiones que puedan ser formalizadas. Por último también se ha concluido de las aportaciones de Zipf que existe una relación inversamente proporcional entre el número de documentos indizados y el número de materias diferentes utilizadas para indizarlos [Dammers 68] (véase gráfico 13).

Retomando los argumentos de Zipf algunos años después, Luhn establece que la significación de cualquier texto está depositada en los términos con frecuencias intermedias y, por tanto, todos aquellos cuyas frecuencias son altas o bajas no tienen valor representativo en la significación del conjunto. Aunque Luhn no desarrolla procedimiento alguno para establecer los límites entre frecuencias altas, medias y bajas es evidente que esta idea está en la base de muchas de las investigaciones en el campo de la recuperación de información que siguieron [Luhn 57]. Como conclu-

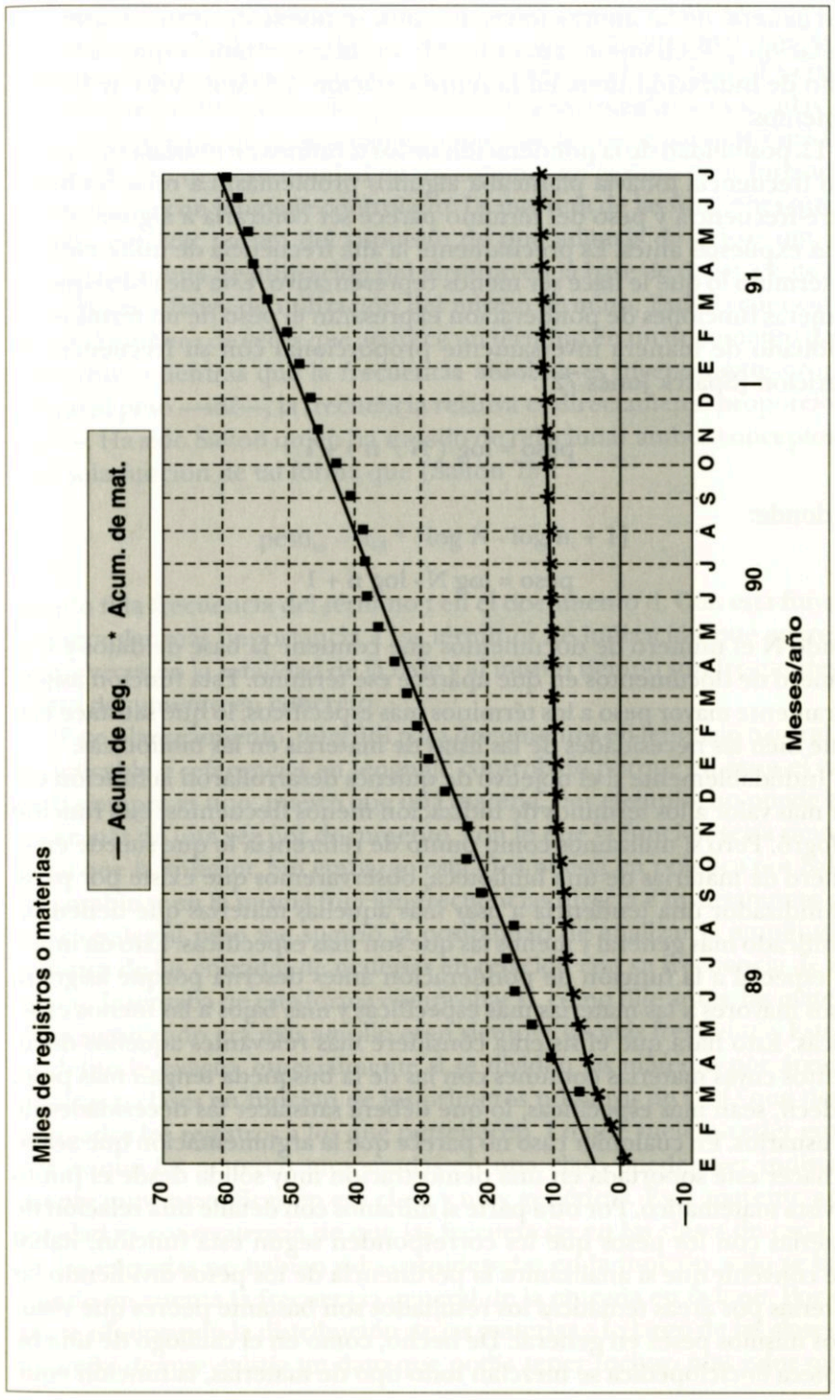


Gráfico 13. *Materias versus registros*

sión general de las aportaciones de Luhn se puede decir que existe una gradación posiblemente cuantificable de la importancia que cada término de indización tiene en la representación del contenido de los documentos.

La posibilidad de la ponderación de los términos de indización en base a su frecuencia todavía planteaba algunos problemas. La relación lineal entre frecuencia y peso del término parece ser contraria a algunas de las ideas expuestas antes. Es precisamente la alta frecuencia de utilización de un término lo que le hace ser menos representativo. Esta idea hizo que las primeras funciones de ponderación expresaran el peso de un término relacionado de manera inversamente proporcional con su frecuencia de aparición [Sparck Jones 72]:

$$\text{peso} = \log (N / n) + 1$$

de donde:

$$\text{peso} = \log N - \log n + 1$$

siendo N el número de documentos que contiene la base de datos y n el número de documentos en que aparece ese término. Esta función asigna claramente mayor peso a los términos más específicos, lo que satisface bastante bien las necesidades de las listas de materias en las bibliotecas.

Indudablemente si el objetivo de quienes desarrollaron la función era dar más valor a los términos de indización menos frecuentes, esta función lo logró. Pero si utilizamos como punto de referencia lo que sucede en el fichero de materias de una biblioteca, observaremos que existe por parte del indizador una tendencia a usar más aquellas materias que tienen un significado más general y menos las que son más específicas. Esto da un valor especial a la función de ponderación antes descrita porque asignará pesos mayores a las materias más específicas y más bajos a las menos específicas. Esto hará que el sistema considere más relevantes aquellos documentos cuyas materias comunes con las de la búsqueda tengan más peso, es decir, sean más específicas, lo que deberá satisfacer las necesidades de los usuarios. En cualquier caso no parece que la argumentación que acabo de hacer esté soportada en una demostración muy sólida desde el punto de vista matemático. Por otra parte si miramos con detalle una relación de materias con los pesos que les corresponden según esta función, habrá que convenir que si analizamos la pertinencia de los pesos dividiendo las materias por áreas temáticas los resultados son bastante peores que vistos estos mismos pesos en general. De hecho, como en el catálogo de una biblioteca enciclopédica se mezclan todo tipo de materias, la función equi-

para entradas por su peso que pueden no ser muy coincidentes desde el punto de vista de su especificidad. Esto me obliga a introducir alguna variante en la función con el fin de adaptarla mejor a la finalidad que persigo.

Aunque las funciones de ponderación desarrolladas y evaluadas durante los últimos veinte años han sido muchas, la mayor parte de estas funciones no son de gran utilidad en los sistemas de gestión de información que trabajan con lenguaje controlado. La mayoría de las funciones que he podido analizar parten del supuesto de que además de existir una frecuencia absoluta de utilización del término en la base de datos, existe una frecuencia relativa de utilización del mismo término. Esta frecuencia expresa el número de veces que aparece un término en un documento dado. Por tanto, mientras que la frecuencia absoluta es inversamente proporcional al peso —idf—, la frecuencia relativa es directamente proporcional —tf—. Ha sido Salton quien ha tratado de relacionar ambos conceptos en una sola función de tal forma que [Salton 73]:

$$\text{peso}_{td} = \text{ft}_{td} * [\log N - \log n_t + 1]$$

siendo ft la frecuencia del término t en el documento d. Con esta función se logra dar más importancia a los términos de indización que aparecen pocas veces en la totalidad de la base y al mismo tiempo son frecuentes en algún documento en concreto.

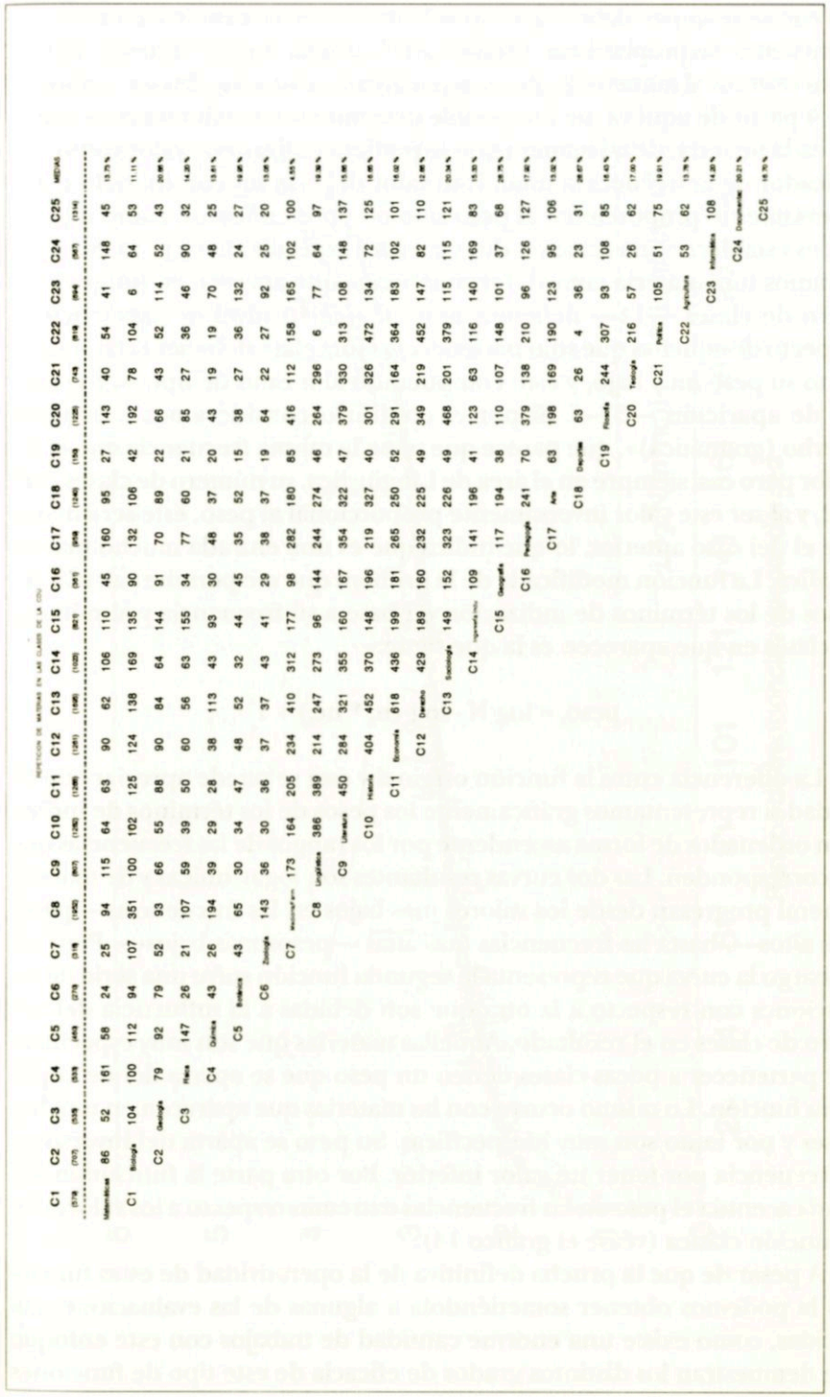
Este planteamiento, muy útil para documentos en lenguaje natural, no es aplicable a referencias en lenguaje controlado porque en éstos el valor de ft siempre es uno, puesto que una materia, por ejemplo, no puede aparecer más de una vez por documento, con lo que la función sería siempre igual que la anterior. Sin embargo esta idea puesta en práctica por Salton de combinar en la misma función frecuencias directa e inversamente proporcionales al peso me sugirió la posibilidad de analizar la amplitud semántica de las entradas de materias en relación con su frecuencia de aparición. Intentaba de esta forma comprobar si, como dije antes, las materias cuyo significado era más amplio eran siempre las más frecuentes. Esto no es del todo exacto, especialmente si se dividen las materias por áreas temáticas o clases en función de los primeros números de CDU que tienen asignados los registros a los que pertenecen. Lo que suele suceder entonces es que las materias muy usadas en una clase pueden ser indistintamente muy específicas en esa clase y muy genéricas. Esta aparente ambigüedad es consecuencia de que las frecuencias en las clases de cada una de las entradas no habían sido consideradas en la función, sólo se había tenido en cuenta la frecuencia general de la entrada en la base. Por otra parte observando la distribución de las materias a lo largo de las clases me apercibí de que existía un dato que podía tener incluso más valor que la

frecuencia de clase en la determinación del peso de esa materia: el número de clases en que aparecía esa materia. Y ello porque este valor, al ser único para cada entrada, permite establecer un único valor de ponderación por materia, mientras que la frecuencia de clase obliga a determinar pesos distintos para cada materia en cada clase.

El procedimiento que seguí para introducir esta variante en la función fue el de seleccionar una serie de áreas temáticas muy amplias que me permitieran dividir la lista general de materias usadas en un catálogo en clases de materias afines por su significado. Una vez seleccionadas las áreas busqué los comienzos de notaciones CDU que las definían. El resultado fue el siguiente:

1. Matemáticas
2. Biología
3. Geología
4. Física
5. Química
6. Botánica
7. Zoología
8. Medicina/Farmacia
9. Lingüística
10. Literatura
11. Historia
12. Economía
13. Derecho
14. Sociología
15. Ingeniería Industrial
16. Geografía
17. Pedagogía
18. Arte
19. Deportes
20. Filosofía
21. Teología
22. Política
23. Agricultura
24. Informática
25. Documentación

A partir de esta selección de veinticinco clases extraje de un catálogo todas las materias que se encontraban en registros cuyas notaciones CDU empezaban por cada uno de los números que identifican según las tablas esas áreas temáticas. De esta forma obtuve veinticinco listas parciales de materias, cada una de las cuales tenía sus correspondientes partes comu-



Repetición de materias en las clases de la CDU

nes con las restantes, debido al solapamiento semántico existente entre las materias y entre las propias áreas. El cuadro de la página anterior muestra algunos datos relativos al número de materias por área y cuántas de ellas son comunes.

A partir de aquí ya me fue posible determinar en cuántas clases aparecía cada materia, de tal manera que se pudiera utilizar este valor como modificador de la frecuencia total. Este valor debería ser considerado como inversamente proporcional al peso si lo que pretendemos, como dije antes, es establecer una relación directa entre especificidad y peso. Si consideramos una materia como «Terminología» que aparece en un gran número de clases —14— debemos admitir que su nivel de especificidad respecto de aquellas que sólo pertenecen a una clase debe ser menor y por tanto su peso muy bajo, y esto computando al mismo tiempo su frecuencia de aparición —37—. Si por el contrario consideramos la materia «Verbo (gramática)», que parece que tiene la misma frecuencia que la anterior pero casi siempre en el área de Lingüística, su número de clases (nc) es 2, y al ser este valor inversamente proporcional al peso, éste será mayor que el del caso anterior, lo que indica que es una entrada mucho más específica. La función modificada de la primera que me permite calcular los pesos de los términos de indización en base a su frecuencia y al número de clases en que aparecen es la que sigue:

$$peso_t = \log N - \log (n_t * nc_t) + 1$$

La diferencia entre la función original y ésta se puede apreciar con facilidad si representamos gráficamente los pesos de los términos de indización ordenados de forma ascendente por los rangos de las frecuencias que les corresponden. Las dos curvas resultantes son logarítmicas y de manera general progresan desde los valores más bajos en las frecuencias —pesos más altos—, hasta las frecuencias más altas —pesos más bajos—. Pero sin embargo la curva que representa la segunda función sufre una serie de oscilaciones con respecto a la otra que son debidas a la influencia del número de clases en el resultado. Aquellas materias que son muy específicas por pertenecer a pocas clases tienen un peso que se aparta del de la primera función. Lo mismo ocurre con las materias que aparecen en muchas clases y por tanto son muy inespecíficas. Su peso se aparta del inverso de la frecuencia por tener un valor inferior. Por otra parte la función modificada acentúa el peso en las frecuencias extremas respecto a los valores en la función clásica (véase el gráfico 14).

A pesar de que la prueba definitiva de la operatividad de estas funciones la podemos obtener sometiéndola a algunas de las evaluaciones conocidas, como existe una enorme cantidad de trabajos con este enfoque que demuestran los distintos grados de eficacia de este tipo de funciones

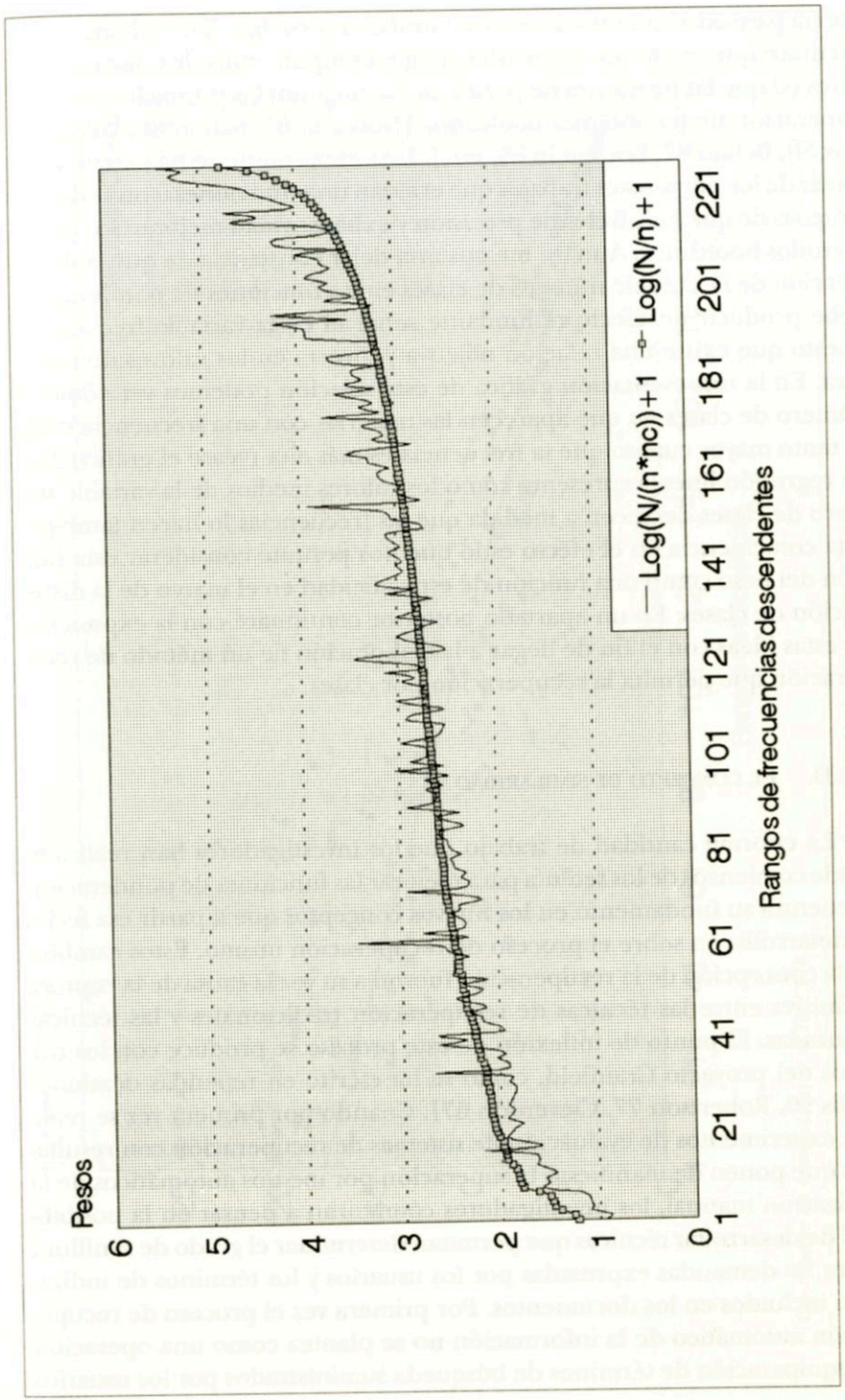


Gráfico 14. *Pesos IDF*

me ha parecido innecesario realizar aquí estas pruebas. Sin embargo sí comentaré que existe una coincidencia generalizada entre los distintos autores en que las funciones de ponderación mejoran las prestaciones de recuperación de los sistemas booleanos [Noreault 81, Salton 88, Wong 92, Fox 88, Belkin 87, Bookstein 85, etc.]. Esta idea común se ha extendido a partir de los numerosos trabajos que evalúan unas funciones u otras dando pruebas de que los niveles de precisión y exhaustividad mejoran los de los métodos booleanos. Aun así me gustaría dejar constancia de que la introducción de la variable número de clases en las funciones de ponderación debe producir un efecto redundante sobre el de la variable frecuencia, puesto que existe una relación objetiva y directa en los valores de una y otra. En la representación gráfica de esta relación podemos ver cómo el número de clases en que aparecen las materias con una frecuencia dada es tanto mayor cuanto que la frecuencia es más alta (véase el gráfico 15). La regresión lineal representa cómo los valores medios de la variable número de clases decrecen a medida que las frecuencias lo hacen también. Esta coincidencia en el efecto es lo que nos permite considerar esta función del peso como una función de especificidad en el marco de la distribución de clases. En un apartado posterior continuaré con la exposición de estas ideas con el fin de llegar a la elaboración de un método de recuperación que permita la recuperación por clases.

VII.D. El concepto de similaridad

La enorme cantidad de trabajo que los investigadores han realizado desde comienzos de los setenta para mejorar las funciones de ponderación encuentra su fundamento en los nuevos conceptos que a partir esa fecha se desarrollaron sobre el proceso de recuperación mismo. Estos cambios en la concepción de la recuperación fueron a su vez la causa de la ruptura definitiva entre las técnicas de recuperación tradicionales y las técnicas avanzadas. El punto de inflexión de este proceso se produce con los trabajos del proyecto Cranfield, como se ha escrito en repetidas ocasiones [Ellis 90, Robertson 77, Cleverdon 67]. Cuando por primera vez se realizan experimentos de evaluación de sistemas de recuperación con resultados que ponen de manifiesto la superación por medios automáticos de la indización manual, los investigadores comienzan a pensar en la posibilidad de desarrollar técnicas que permitan determinar el grado de similitud entre las demandas expresadas por los usuarios y los términos de indización incluidos en los documentos. Por primera vez el proceso de recuperación automático de la información no se plantea como una operación de equiparación de términos de búsqueda suministrados por los usuarios

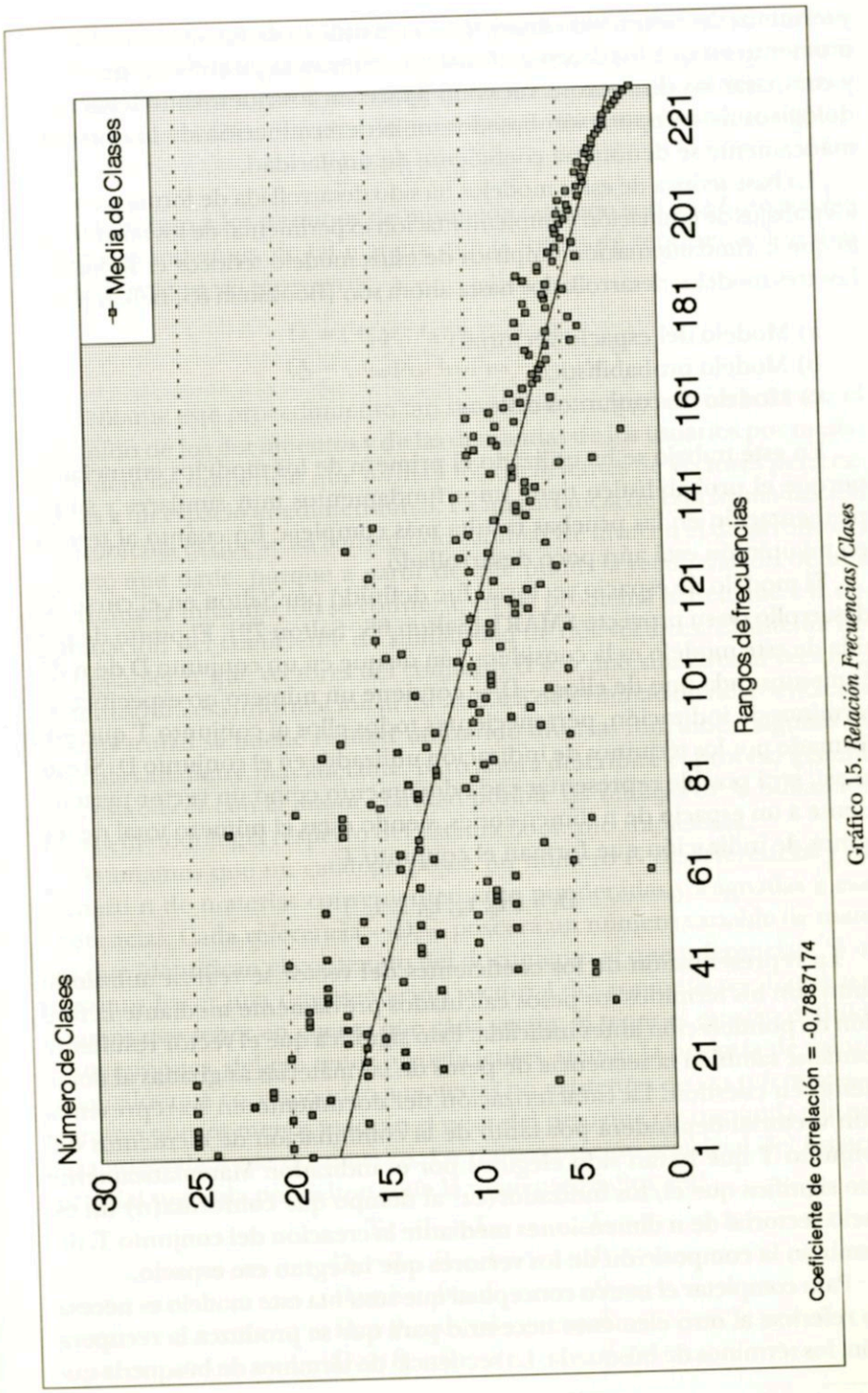


Gráfico 15. *Relación Frecuencias/Clases*

y términos de indización existentes en los índices de los sistemas. Hasta el momento en que los investigadores no tuvieron la posibilidad de evaluar y comparar las distintas técnicas no aparecen los nuevos modelos metodológicos de recuperación basados en la determinación de lo que matemáticamente se denomina coeficiente de similaridad.

La base teórica de estos modelos ha sido desarrollada de forma paralela a los trabajos de evaluación e implementación experimental de los mismos, por lo que la fundamentación empírica de estos modelo teóricos es indudable. Los tres modelos desarrollados hasta ahora son [Bookstein 85, Belkin 87]:

a) Modelo del espacio vectorial
b) Modelo probabilístico
c) Modelo de conjuntos difusos

En este trabajo se ha utilizado el primero de los modelos enunciados, porque el probabilístico tiene unos fundamentos muy similares y su implementación en las pruebas resulta más compleja, En cuanto al tercero en mi opinión está aún poco desarrollado.

El modelo de espacio vectorial fue definido por Salton en el curso del desarrollo de su proyecto SMART [Salton 68, Salton 79]. El punto de partida de este modelo es la consideración de que en un conjunto D de n documentos cada uno de ellos —D_i— contiene un número no específico de términos de indización, pertenecientes todos ellos al conjunto T que está formado por los términos de indización utilizados en el conjunto D. Si esto es así, será posible representar cada documento como un vector perteneciente a un espacio de n dimensiones, siendo éstas el número total de términos de indización que forman el conjunto T:

$$D_i = (t_{ij}, t_{ik}, t_{il}, \ldots t_{in})$$

La representación de los coeficientes del vector se realiza utilizando junto con los términos los pesos calculados previamente mediante la función de ponderación antes indicada. Esto significa que el vector resultante contiene también la secuencia de pesos de las materias asignadas al documento en cuestión. La caracterización del documento en su representación vectorial dependerá por tanto de la combinación de elementos del conjunto T que hayan sido elegidos por el indizador. Matemáticamente esto significa que el/los indizador(es) al tiempo que conforma(n) un espacio vectorial de n dimensiones mediante la creación del conjunto T, determinan la composición de los vectores que integran ese espacio.

Para completar el marco conceptual que sustenta este modelo es necesario referirse al otro elemento necesario para que se produzca la recuperación, los términos de búsqueda. La secuencia de términos de búsqueda que

deben ser comparados con los vectores de los documentos también deben representase en su forma vectorial. Cada interrogación —query— por parte de un usuario dará lugar al vector Q, que estará integrado por los r términos de búsqueda seleccionados de entre los que forman el conjunto T:

$$Q_k = (t_{ka}, t_{kb}, \ldots t_{kr})$$

En este caso a los términos de búsqueda también son añadidos sus pesos correspondientes, de tal manera que la forma de representación más exacta de ambos vectores sería:

$$D_i = (t_{ij},p_{ij}; t_{ik},p_{ik}; \ldots t_{is},p_{in})$$
$$Q_k = (t_{ka},p_{ka}; t_{kb},p_{kb}; \ldots t_{kr},p_{kr})$$

Mediante este procedimiento tan simple hemos logrado caracterizar el contenido de los documentos y de las demandas de los usuarios por medio de secuencias numéricas que forman los coeficientes de vectores pertenecientes a un espacio enedimensional. La importancia de esta formalización del problema es que se ha conseguido cambiar de nivel en el desarrollo del proceso que sigue, porque a partir de este punto la recuperación de una determinada información se convierte en un problema matemático: la determinación del coeficiente de similaridad de dos vectores. El sistema deberá establecer qué vectores del conjunto D son más similares al vector Q y suministrará esa información al usuario. La determinación del coeficiente de similaridad se puede realizar utilizando diversas funciones, algunas de las cuales analizaré más adelante, pero ahora describiré el proceso general para establecer con la mayor claridad posible las ventajas de la utilización de esta metodología respecto del sistema booleano tradicional.

Asumamos que un catálogo posee un número N de referencias y un número n de materias convenientemente ponderadas y asignadas a esas referencias. Cada referencia —D_i— poseerá un número variable de materias que identifican el contenido del documento al que referencian. Si intentamos hacer una representación vectorial del conjunto resultante tendríamos que tener en cuenta que cada vector, al tener el espacio definido n dimensiones, deberá tener n coeficientes, siendo la mayoría de valor 0 —aquellos que correspondan a materias no existentes en esa referencia—, mientras que los menos tendrán el valor del peso que corresponda a la materia asignada a ese documento. La representación matricial del espacio vectorial sugerida por Salton sería la siguiente [Salton 89]:

	T_1	T_2	T_3	T_n
$D_1 =$	0	0	1	0
$D_2 =$	5	6	0	0
$D_3 =$	0	0	1	3
$D_N =$	0	0	0	3

Los valores 0 de los coeficientes de los distintos vectores corresponden a materias no asignadas a los documentos en cuestión. Los valores superiores a 0 corresponden a los pesos de las materias asignadas a esos documentos.

Si hacemos una operación similar con la secuencia de términos de búsqueda entregada al sistema por un usuario, obtendremos una matriz mucho más simple:

	T_1	T_2	T_3	T_n
Q_1 =	0	6	1	3

La pregunta del usuario es relativa a dos materias. Por tanto, los valores mayores que 0 en la matriz se corresponden con los pesos de las tres materias elegidas. En un sistema de recuperación booleano para realizar esta consulta existen dos alternativas que dependen del operador utilizado para relacionar estas materias. Si utilizo la conjunción sólo recuperaré los documentos que contengan todas esas materias —en este caso no hay ninguno en la matriz— y si, por el contrario, utilizo la disyunción obtendré todos aquellos documentos que contienen alguna de las materias solicitadas —los números 1, 2, 3 y N de la matriz—. En cambio, en un sistema de recuperación basado en el modelo de espacio vectorial no sería necesario utilizar ningún operador para ligar las materias. El usuario sólo tendría que seleccionarlas; por otro lado, el sistema realizaría la búsqueda comparando el vector Q_1 con cada uno de los vectores de documentos, y como resultado de esas comparaciones obtendría un valor que expresaría la similaridad existente entre cada par de vectores, de tal manera que la salida de los documentos recuperados por el sistema estaría ordenada de los que tienen un coeficiente mayor a los que lo tienen menor. Por ejemplo, si existen dos materias comunes entre el vector Q_1 y el vector D_3, y los pesos de estas materias comunes son 1 y 3, su coeficiente de similaridad será menor que el del par Q_1 D_2, porque aunque sólo tienen una materia común su peso es mayor que la suma de los otros dos. En definitiva los documentos recuperados mediante la utilización de esta técnica serían suministrados a los usuarios ordenados de los más afines a la búsqueda introducida a los menos afines.

Una vez descrito el proceso general podemos pasar a analizar algunas de las funciones de similaridad propuestas hasta ahora para evaluar la mejor adecuación de cada una de ellas al entorno informativo del que hablamos. Las funciones de similaridad más utilizadas no han sido definidas en el curso del desarrollo de este modelo, sino que se han extraído de las aportaciones que en el campo de la matemática aplicada se habían hecho anteriormente. Estas funciones pueden dividirse en tres grupos: a) las puramente vectoriales, b) las basadas en cálculos de distancias euclidianas en espacios de n dimensiones, y c) las desarrolladas a partir de la teoría de probabilidades. En este trabajo me limitaré a analizar las del primer grupo, ya que, según las evaluaciones consultadas, son las que ofrecen mejores resultados en la recuperación a través de materias [Noreault 81]. Por otra parte fueron éstas las funciones sugeridas por los que desarrollaron el modelo para su implementación en sistemas de recuperación [Salton 83, 88, Rijsbergen 79].

La función más simple de similaridad vectorial conocido es el llamado producto escalar de dos vectores:

$$SIM\ (D_i\ Q_j) = \Sigma\ (x_i * y_i)$$

Siendo x_i cada uno de los coeficientes del vector D_i, mientras que y_i representa a los coeficientes del vector Q_j. Si utilizamos esta función para reproducir el caso anterior, obtendremos:

$$SIM\ (D_1\ Q_1) = 0*0 + 0*6 + 1*1 + 0*3 = 1$$
$$SIM\ (D_2\ Q_1) = 5*0 + 6*6 + 0*1 + 0*3 = 36$$
$$SIM\ (D_3\ Q_1) = 0*0 + 0*6 + 1*1 + 3*3 = 10$$
$$SIM\ (D_N\ Q_1) = 0*0 + 0*6 + 0*1 + 3*3 = 9$$

De tal forma que la función asigna un coeficiente de similaridad más alto a aquellos pares cuyos coeficientes comunes tienen valores mayores. Dicho de otra manera, son los documentos cuyas materias más específicas coinciden con las materias de la búsqueda los que aparecen en la respuesta del sistema en primer lugar.

Otras funciones definidas con el mismo objeto devuelven valores de entre 0 y 1 para el coeficiente de similaridad, aunque, como veremos después, al hacer intervenir en la función el módulo del vector, su resultado es diferente al del producto escalar. Estas funciones son los coeficiente del Coseno, Dice y Jaccard:

Coseno

$$SIM(Q_j\ D_i) = \Sigma(x_i*y_i) / \Sigma x_i+\Sigma y_i$$

Dice

$$SIM(Q_j\ D_i) = 2\Sigma(x_i*y_i) / \Sigma x_i+\Sigma y_i$$

Jaccard

$$SIM(Q_j\ D_i) = \Sigma(x_i*y_i) / \Sigma x_i+\Sigma y_i-\Sigma(x_i*y_i)$$

Las ventajas principales del modelo de espacio vectorial han sido resumidas por Salton recientemente [Salton 89 pág. 317)]:

a) Los documentos pueden ser ordenados en orden decreciente de sus coeficientes de similaridad con los términos de búsqueda, lo que nos permite verlos en orden decreciente de su presunta importancia.

b) El tamaño de los conjuntos recuperados puede ser adaptado a las necesidades de los usuarios. En el caso de usuarios normales sólo sería necesario mostrar los primeros documentos recuperados, mientras que para usuarios especializados se mostrarían mayor cantidad de documentos. En cualquier caso los primeros siempre serían los más importantes.

c) Los documentos recuperados en una búsqueda anterior pueden ser utilizados para señalar los más relevantes. A partir de éstos el sistema ha de generar otra búsqueda más precisa.

Por último es importante señalar que este modelo también ha recibido alguna crítica [Raghavan 86], centrada especialmente en el problema de la ortogonalidad de los vectores de la base. La aplicación en los términos expuestos de las funciones de similaridad exige que los vectores de la base del espacio vectorial definido —T_i— sean linealmente independientes, es decir, su producto escalar debe ser igual a 0, de lo contrario será necesario establecer coeficientes de correlación entre ellos para combinarlos linealmente de tal forma que pierdan su independencia. En términos de recuperación de información en entornos bibliográficos este problema resulta irrelevante puesto que las materias asignadas a los registros son completamente significativas por sí mismas y no suelen mantener relación alguna con otras, en el sentido de necesitarlas para completar su significado. Este problema no tiene conexión alguna con las relaciones semánticas existentes entre los descriptores que conforman un thesaurus o las que pueden existir en una lista de materias. Este tipo de críticas sólo plantean problemas a nivel teórico, porque en la práctica las situaciones sobre las que especulan no tienen referentes posibles, especialmente en el caso que nos ocupa.

Sin embargo, una dificultad importante en relación con este modelo es la elección de la función de similaridad. Sobre este asunto no existe ninguna base conceptual útil y, por tanto, no es posible establecer a priori bajo qué condiciones sería preferible utilizar una función u otra. Por este motivo he tratado de profundizar en el problema para poder fundamentar la elección en el caso de un sistema de recuperación de información catalográfica.

VII.E. El valor de discriminación de los términos

El punto de partida del análisis que vengo haciendo sobre las técnicas no booleanas de recuperación fueron los estudios de Zipf y Luhn. Y fue precisamente Luhn quien introdujo en algunos de sus trabajos el concepto de poder de resolución de un término de indización [Luhn 57]. Con este concepto se pretendía expresar la capacidad que cualquier término de indización tiene para identificar documentos relevantes para el usuario. Esta capacidad debía ser mayor en aquellos términos con frecuencias intermedias, pero Luhn nunca desarrolló ningún procedimiento formal para la determinación cuantitativa de esta capacidad. En realidad esta idea, como otras, de este pionero de la recuperación de información era más el resultado de una intuición que otra cosa. Pero lo que es un hecho es que un método eficaz para la determinación de la capacidad de discriminación informativa de los términos de indización podría ser muy útil para la indización automática de los documentos e incluso para poder evaluar las funciones de similaridad, como luego veremos.

Fue precisamente Salton quien definió por primera vez una función que permitiera la determinación exacta de lo que él llamó el valor de discriminación de un término [Salton 73, 75b]. Para describir el funcionamiento de esta función es preciso analizar previamente ciertos conceptos relativos a la consideración física del modelo teórico desarrollado hasta aquí.

Si consideramos por un momento, y por razones expositivas, el espacio vectorial definido en el ejemplo anterior como un espacio euclidiano de n dimensiones, convendremos que lo que antes llamábamos vectores ahora serán puntos de ese espacio y los componentes de esos vectores serán las coordenadas de esos puntos. De igual forma los coeficientes de similaridad pasarán a ser distancias. Estas distancias expresarán la cercanía o alejamiento de los contenidos de los documentos en cuestión, por lo que existirá siempre una relación inversamente proporcional entre la distancia entre dos puntos y el coeficiente de similaridad de los documentos que representan esos mismos puntos. Expresado de manera simple se puede decir que cuando los puntos de un espacio de estas características están muy cerca unos de otros, los documentos que representan tienen muchas materias con pesos altos comunes, es decir, son documentos temáticamente afines. Por el contrario, cuando los puntos están alejados entre sí, los documentos que representan tienen pocas materias, y de poco peso, comunes, es decir, estos documentos son temáticamente poco afines. En términos físicos esto podría llevarnos a las siguientes conclusiones:

a) La densidad del espacio que conforman los pesos de los documentos indizados en una base de datos es la expresión de la afinidad temática de los documentos, representada ésta por la totalidad de los términos de indización utilizados para expresar sus contenidos.
b) El cálculo de la densidad de ese espacio se realiza mediante la determinación de la suma total de los coeficientes de similaridad

existentes entre todos los pares posibles de documentos existentes en la base de datos.

$$\text{densidad} = \Sigma\ \text{SIM}\ (D_i\ D_{i+1})$$

c) Si calculamos la media de similaridad —ms— de un espacio cualquiera, sería posible utilizar este valor para comparar el grado de afinidad temática que expresa un conjunto dado de materias. La media de similaridad se calcula dividiendo la densidad por el número de pares de documentos existentes en la base de datos:

$$ms = 1\ /\ N\ (N - 1)\ *\ \Sigma\ \Sigma\ \text{SIM}\ (D_i\ D_{i+1})$$

Llegados a este punto y retomando el argumento inicial podemos decir que la influencia de cada materia en el valor que hemos denominado densidad debe ser directamente proporcional a su capacidad discriminativa. Es decir, cuanta más capacidad tenga una materia para establecer diferencias entre lo relevante y lo no relevante, mayor será su aportación al valor densidad. Lógicamente esto siempre estará condicionado por la función elegida para la determinación de la similaridad.

La formalización de esta idea se hizo partiendo del principio expuesto en relación con la aportación que cada materia realiza a la media de similaridad. Para calcular esta aportación se propone inicialmente la realización del siguiente proceso de cálculo:

1) Cálculo ms:

$$ms = 1\ /\ N\ (N - 1)\ *\ \Sigma\ \Sigma\ \text{SIM}\ (D_i\ D_{i+1})$$

2) Eliminación de una materia de la base de datos.
3) Determinación de ms sin la materia excluida.
4) La resta de los valores de ms obtenidos nos proporciona el valor de discriminación —vd— de la materia excluida:

$$vd_i = ms - ms_i$$

Los vd obtenidos mediante este proceso se pueden dividir en tres grupos según que tengan un vd negativo, próximo a cero o mayor que cero. Esta gradación permite evaluar no sólo el comportamiento de la función de similaridad, sino también el uso que los indizadores hacen de la lista de materias utilizada. Los vd puestos en relación con la frecuencia de utilización de las materias nos indicarán, así mismo, de manera inequívoca si, como especuló Luhn, el mayor poder de resolución de un término de indización está asociado a las frecuencias intermedias o no. Y, por último, existe la posibilidad de utilizar este valor para determinar los pesos según una nueva función que es resultado de multiplicar la frecuencia de un término de indización en un documento por el valor de discriminación asignado a este término [Salton 76, 83]:

$$peso_{ji} = f_{ji} * vd_i$$

Lo que, en el caso de las materias usadas en las referencias catalográficas, significa que el peso siempre sería el vd porque la frecuencia del término en el documento —f_{ji}— siempre es 1.

A partir de la publicación del modelo teórico y su formalización matemática empezó un largo proceso, que aún hoy prosigue, de desarrollo de métodos operativos para la determinación del vd. El modelo definido por Salton en lo que afecta a su proceso matemático de determinación fue desde el primer momento tildado de inviable por la enorme cantidad de operaciones que exigía. Téngase en cuenta que en el conjunto de datos estudiados por mí, formado por unos 59.000 documentos con algo menos de 10.000 entradas de materias diferentes asignadas, sería preciso calcular cerca de trescientos cincuenta millones de coeficientes de similaridad para determinar cada uno de los vd que serían necesarios para obtener, por ejemplo, los pesos de todas las materias de esa colección. Esta ingente tarea de cálculo resulta poco viable con medios normales si se realiza mediante procedimientos batch, e imposible con cualquier tipo de medios si se pretende realizar en tiempo real. Estas circunstancias son las que han hecho que se hayan desarrollado muchos trabajos de investigación para tratar de resolver esta cuestión. El primero en aportar una solución fue el creador del modelo [Salton 83], que sugirió la posibilidad de utilizar un documento centroide como elemento de comparación único en el cálculo de la ms. Otros autores han propuesto distintas soluciones [Crawford 75, Willet 85, El-Hamdouchi 88, Biru 89], todas ellas en la línea de reducir los tiempos de cálculo para hacer viable la implementación de este método de ponderación en sistemas reales.

La solución elegida por mí de entre las propuestas es la denominada valor de discriminación aproximado, que ha sido valorada muy positivamente [Crouch 88] por el altísimo grado de compatibilidad que tiene con los algoritmos que resuelven el valor de discriminación exacto, desaconsejándose la utilización de éste por el enorme costo que comporta.

El cálculo realizado pretendía probar las cuatro funciones de similaridad comentadas en el apartado anterior, con el fin de determinar cuál de ellas podría adecuarse mejor a la recuperación de información catalográfica. Para lo cual era necesario ejecutar cuatro programas que calcularan

los vd de todas las materias que formaban parte de la base de datos utilizada para la prueba. El método de cálculo elegido se desarrolla en cuatro fases:

1) En la primera se calculan los componentes del vector del documento centroide. Los componentes de este vector son la media de los componentes de todos los vectores que forman el espacio vectorial. En realidad este documento medio es uno ficticio que se encuentra en el centro geométrico del espacio y cuya representación sería:

$$C = (c_1, c_2, c_3, \ldots c_n)$$

2) A continuación se calcula el coeficiente de similaridad de cada vector de la base con el centroide, lo que reduce considerablemente el número de cálculos. Al mismo tiempo se acumulan todos los coeficientes para calcular la ms de la base con todas sus materias.

3) A partir de aquí se van eliminando materias y recalculando las ms para restar cada una de ellas de la media calculada en la fase dos. El resultado obtenido es el vd de la materia correspondiente.

Estas tres fases se realizan en dos rutinas —las fases dos y tres en la misma— (véase Apéndice), y la segunda se ha ejecutado, con variantes, cuatro veces, una por cada función de similaridad evaluada. El modelo base de estos cuatro procesos ha sido diseñado tratando de optimizar al máximo su funcionamiento, tanto en lo que afecta a la precisión del cálculo como a los tiempos de ejecución. La solución de proceso que encontré más idónea está basada en cálculos de matrices generadas directamente en memoria principal. El proceso base comienza con la carga de las matrices utilizando datos almacenados en ficheros. Las matrices generadas son tres, el centroide —m[a]—, una para los números de las materias —n[b][c]— y otra para los pesos de esas materias —p[b][c]—. A partir de este punto se inicia el cálculo de los coeficientes de similaridad y valores de discriminación para los 527 rangos de frecuencias de materias existentes en la base de datos. Esta parte del proceso se realiza así con el fin de evitar volver a calcular los vd de aquellas materias que tienen la misma frecuencia de uso. La salida de estos procesos se envía a un fichero donde se almacenan los vd obtenidos.

Los resultados de las cuatro ejecuciones han sido representados gráficamente para facilitar la interpretación de los resultados. El sistema de representación gráfico ha sido elegido porque permite visualizar con facili-

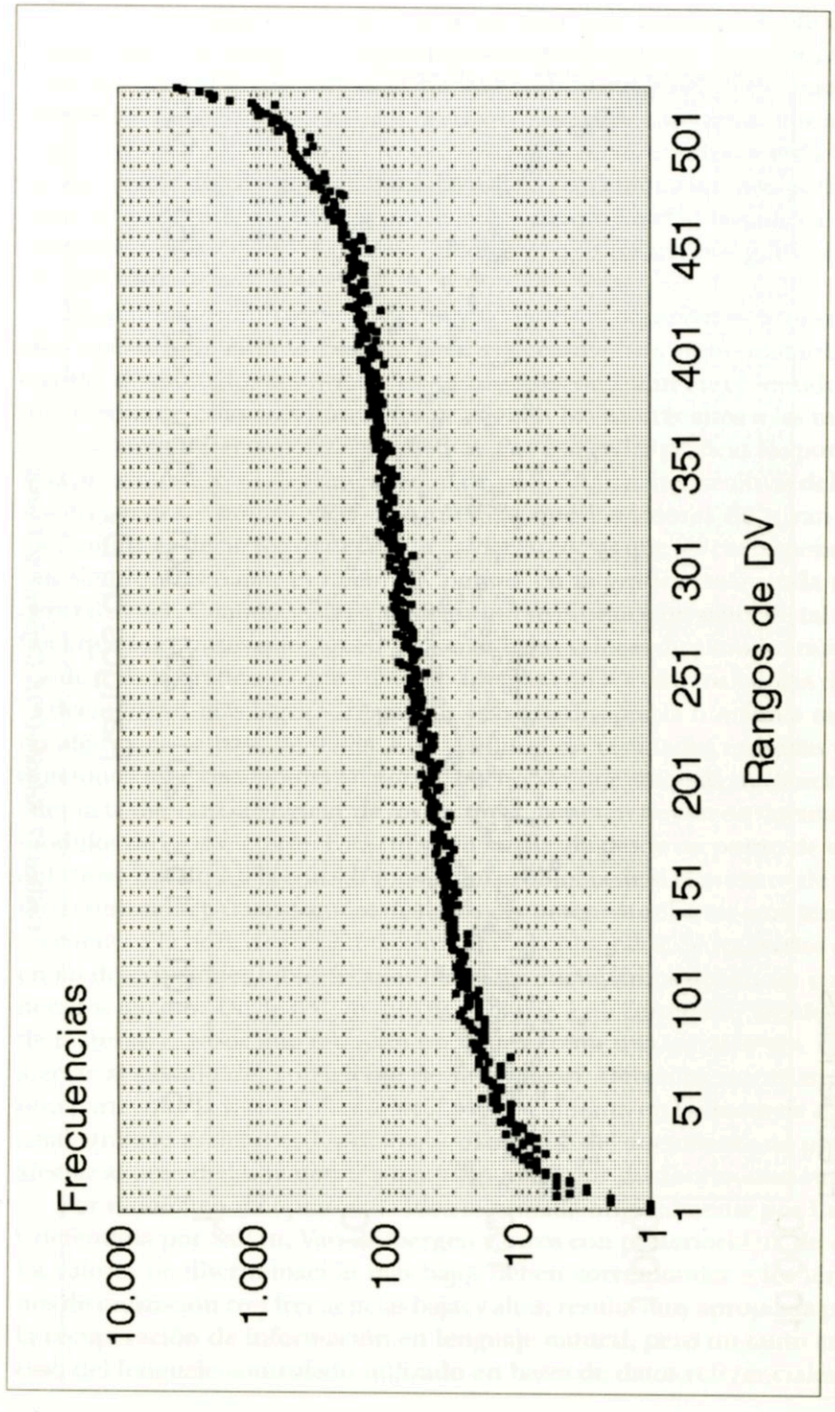


Gráfico 16. *Relación DV/F. Producto escalar.*

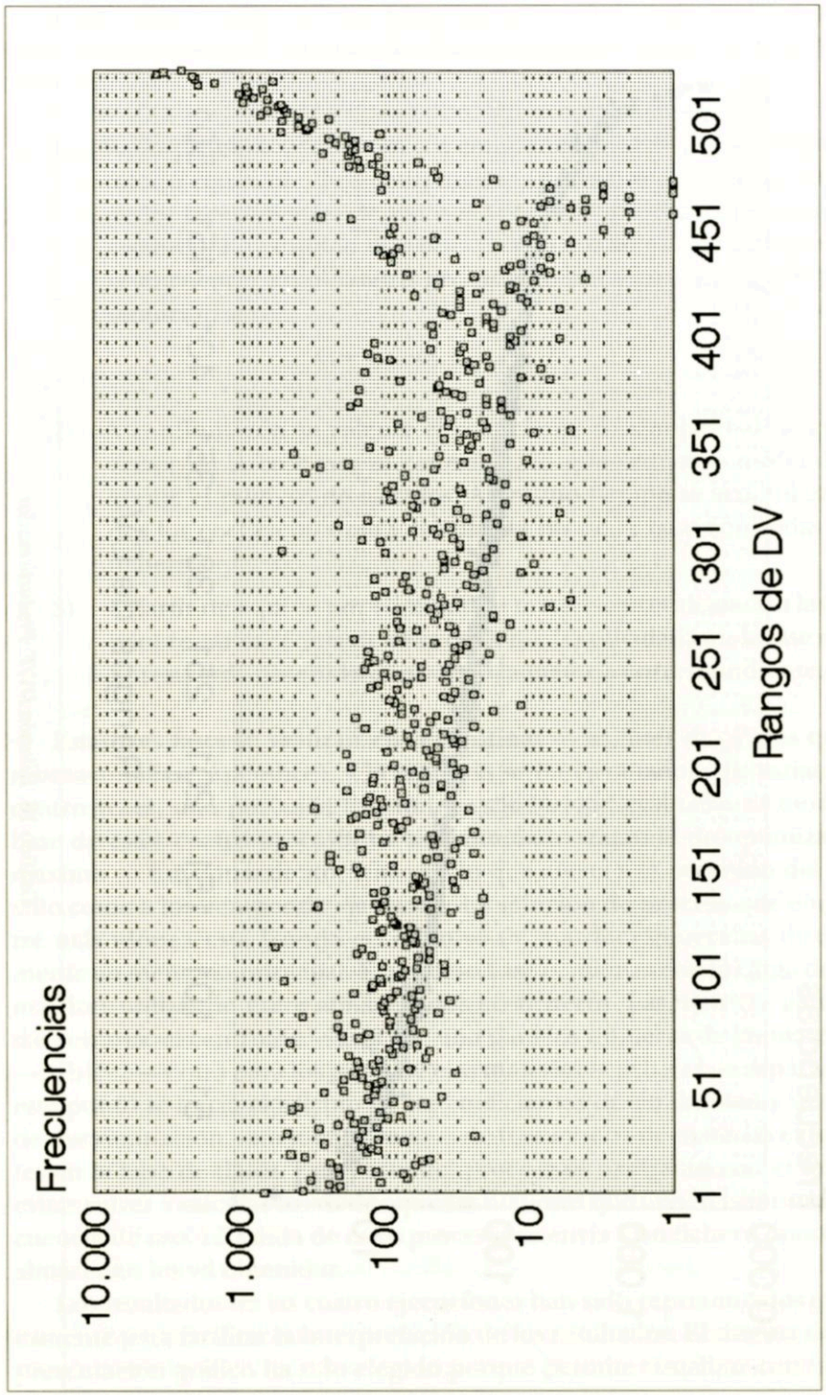


Gráfico 17. *Relación DV/F. Coeficiente del Coseno.*

dad la relación existente entre vd y la frecuencia de los términos de indización, y porque ha sido utilizado con éxito recientemente [Biru 89], aunque con algunas variantes. En el eje x se colocan ordenados de mayor a menor los rangos de los valores de discriminación obtenidos, mientras que en el eje y se colocan las frecuencias que corresponden a cada uno de los rangos de vd del eje x. Esta forma de representación nos permite apreciar para cada uno de los coeficientes de similaridad testados a qué frecuencias se asocian los vd altos, medios y bajos (véanse los gráficos 16, 17, 18 y 19).

Si analizamos en conjunto las cuatro gráficas observamos a primera vista que tres de ellas —Coseno, Dice y Jaccard— son muy similares en cuanto al resultado; confirman las intuiciones de Luhn en el sentido de que mediante estas tres funciones se asignan los vd más altos a las materias que tienen frecuencias intermedias. Por eso en las gráficas los puntos más próximos a x = 1 —primer rango de vd— están a media altura del eje y —frecuencias intermedias—. A medida que los valores de x van aumentando, es decir, los vd decrecen, las frecuencias que les corresponden van siendo más bajas, es decir, los puntos en la gráfica están cada más cerca de y = 1. Cuando se llega a este punto se produce un salto, de tal manera que los puntos que siguen a los más bajos valores de y son los más altos de toda la gráfica, lo que coincide además con los últimos valores de x, es decir, los vd más bajos —negativos— asignados por la función a materia alguna en la base de datos. Esta similitud de resultados entre las tres funciones está condicionada por el hecho de que además de tener en cuenta la diferencia angular de los vectores, también tienen en cuenta los módulos de esos vectores. Esta cuestión analizada desde un punto de vista aplicativo nos lleva a la consideración del problema de si el número de materias utilizadas para indizar los distintos documentos debe ser o no tenido en cuenta a la hora de calcular la similaridad entre ellos. Se ha escrito que en las descripciones bibliotecarias todos los documentos deben ser considerados iguales [Doyle 63], lo que nos llevaría a no considerar el número de materias o pesos que integran un vector como una variable que deba afectar al cálculo del coeficiente de similaridad. Debemos recordar, por otra parte, que la variable frecuencia por documento en las bases de datos catalográficas es siempre 1, por lo que el tamaño del documento no puede afectar al peso de las materias y tampoco a su valor de discriminación.

Por último, en mi opinión, la idea expresada originalmente por Luhn y defendida por Salton, Van Rijsbergen y otros con posterioridad, de que los valores de discriminación más bajos deben corresponder a los términos de indización con frecuencias bajas y altas, resulta muy apropiada para la recuperación de información en lenguaje natural, pero no tanto en el caso del lenguaje controlado utilizado en bases de datos referenciales bi-

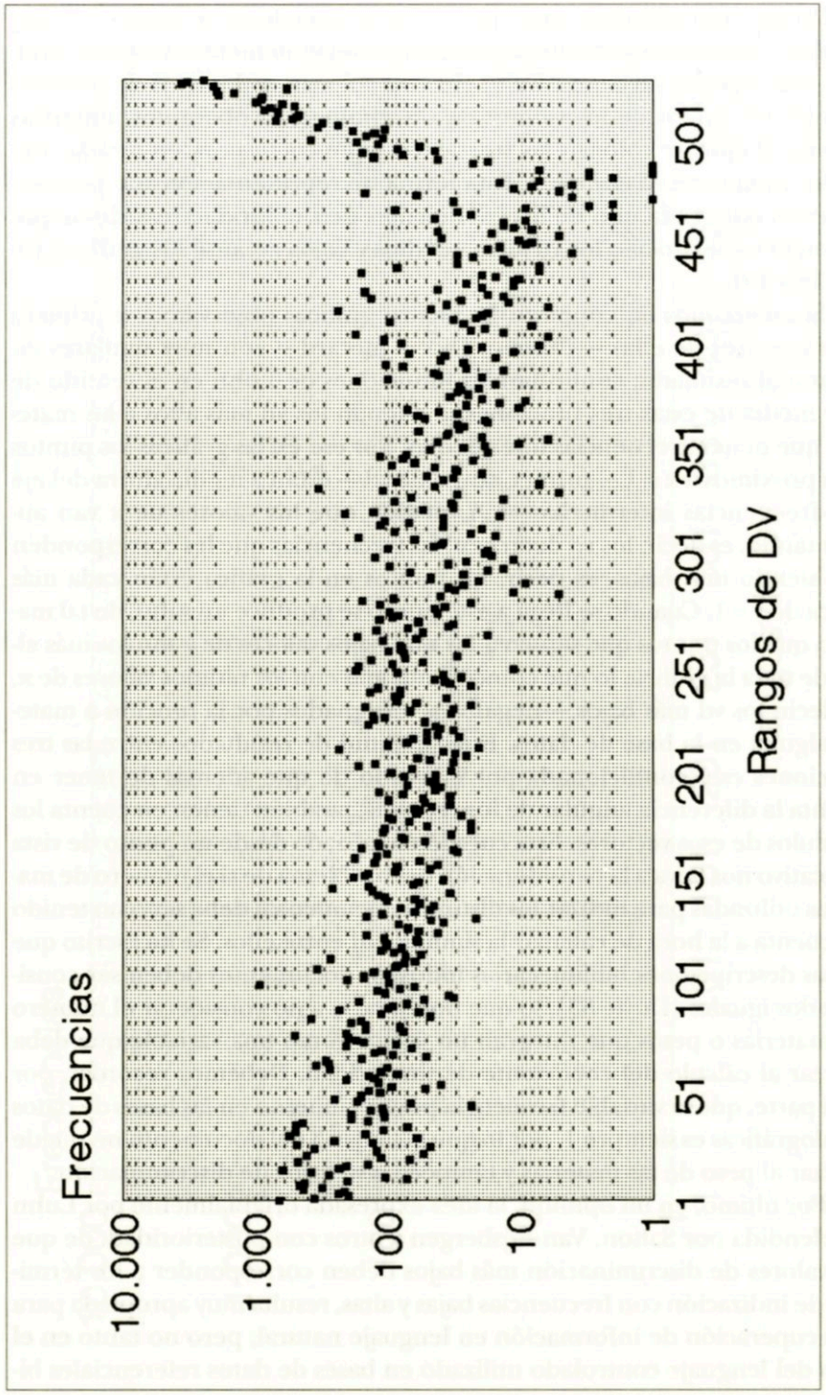


Gráfico 18: *Relación DV/F. Coeficiente de Dice.*

bliográficas. Esta afirmación, sobre la que luego volveré en las conclusiones, es consecuencia del hecho, constatable empíricamente, de que los niveles de especificidad mayores en las listas de materias se encuentran habitualmente en materias que tienen frecuencias muy bajas, si los valores de discriminación terminan siendo directamente proporcionales a los pesos, como dije anteriormente siguiendo a Salton, los usuarios de los catálogos en línea se verían defraudados en sus expectativas de recuperación si la ordenación de los documentos por relevancia que el sistema realiza coloca al final todos aquellos que tienen las materias más específicas de entre las seleccionadas como términos de búsqueda. Y es justamente éste el tipo de ordenación de las entradas que realizan las tres funciones que estoy comentando. Por estos motivos reseñados hasta aquí, considero que la función más simple —producto escalar— es la más apropiada para el tipo de tratamiento del que me ocupo.

Si analizamos la gráfica correspondiente a este coeficiente, observaremos que la relación entre las frecuencias y los vd varía notablemente respecto de los casos anteriores. En este caso son las materias con frecuencias más bajas las que han recibido los vd más altos y, por el contrario, las materias con frecuencias más altas las que han recibido los vd más bajos. Esta relación permite asignar pesos mayores a lo que de antemano hemos considerado como términos de indización más específicos, las materias menos frecuentes. Obsérvese que todo este planteamiento no tendría sentido si no fuera cierta mi observación inicial de que existe una relación inversamente proporcional entre el nivel de especificidad de una materia y su frecuencia de uso, observación que casi he convertido en axioma de manera consciente ante la constatación empírica que se puede hacer de esta relación.

En resumen, en un sistema de gestión de información bibliográfica en el que se utilicen listas de materias, la función de similaridad idónea en el marco del modelo de espacio vectorial es el producto escalar, dados los resultados arrojados por los análisis de los vd calculados según las funciones de los distintos coeficientes. Pero aun así queda sin resolver una cuestión planteada en el apartado anterior, la de los niveles de especificidad de las materias en cada una de las áreas temáticas definidas en el conjunto de la base de datos. Es evidente que esta cuestión no podrá ser resuelta por la vía de la utilización de funciones, por sofisticadas que éstas sean, que se aplican de manera homogénea a la totalidad de las materias. Será necesario recurrir a técnicas de indización de análisis de cluster.

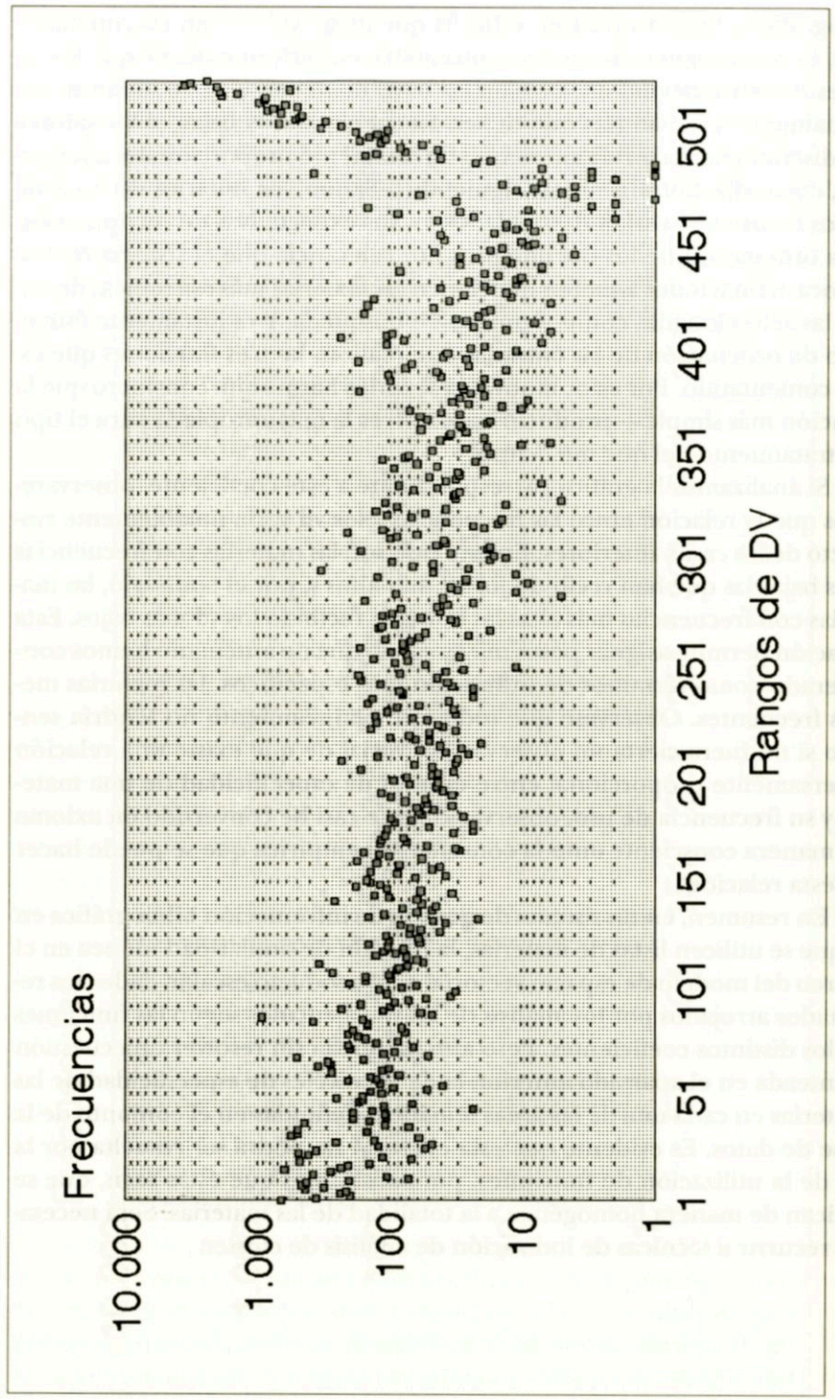


Gráfico 19: *Relación DV/F. Coeficiente de Jaccard.*

VII.F. El método de cluster heurístico

Antes de describir el método en cuestión, me gustaría hacer una breve valoración del estado de la búsqueda por materias en los catálogos en línea, con el fin de resaltar aquellos aspectos funcionales que en este tipo de aplicaciones pudieran ser objeto de mejora y, de esta manera, situar más adecuadamente mi aportación.

Como ya mencioné anteriormente, estudios recientes sobre el comportamiento de los usuarios en relación con los OPAC revelan que la búsqueda por materias está en declive. Aunque los problemas de la búsqueda por materias han sido puestos de manifiesto en repetidas ocasiones, los datos aportados durante toda la década de los ochenta ponían de relieve que un enorme porcentaje de las búsquedas —en torno al 50%— se realizaba a través de los campos de materias, lo que ponía de relieve el enorme interés de los usuarios por disponer de unos eficaces sistemas de recuperación a través de las materias. Los problemas habituales encontrados por los usuarios hacen referencia a falta de familiaridad con el sistema de indización, errores tipográficos, desconocimiento del interface de búsqueda, falta de información de referencia en las materias, pero sobre todo la enorme cantidad de información que el usuario recibe del sistema en respuesta a cualquier demanda. Este estado de cosas ha hecho que en los últimos años hayan proliferado los estudios encaminados a la mejora de la recuperación por materias.

Una de las técnicas más valoradas por los expertos ha sido la del diseño de interfaces de recuperación que permitan a los usuarios la realización de búsquedas vía browsing. Esta técnica en sus distintas modalidades [Hancock 89] permite al usuario recorrer las materias con la esperanza de encontrar alguna útil. Como se ha dicho recientemente, este método es el arte de no saber qué busca uno mientras lo encuentra [Cove 88], y la realidad es que los interfaces que permiten ciertas modalidades de browsing diseñados hasta ahora no facilitan al usuario otras herramientas que las de avanzar o retroceder a través de una lista ordenada alfabéticamente de materias. Son necesarias soluciones que faciliten al usuario la identificación de entradas no previstas por él, porque las que él prevé son, a menudo, las más frecuentes y por tanto las de significado más amplio. Existe una correlación constatable entre la frecuencia de uso de los términos de indización por parte del indizador y la frecuencia de elección de esos mismos términos en las búsquedas por parte de los usuarios. Dicho de otra forma, las materias que más utilizan los bibliotecarios al catalogar son las más usadas por los usuarios de los OPAC [Nelson 88], lo que significa que el problema de la sobreabundancia de información se ve incrementado por la actitud de los usuarios en el momento de realizar su búsqueda. Como se

puede ver, junto a la tendencia del indizador a utilizar con más frecuencia lo más genérico, existe la tendencia del usuario a buscar a través de lo más genérico también. Ante situaciones como ésta, carece de sentido hacer recorrer al usuario las materias porque lo probable es que se detenga en aquellas que tengan significados más amplios, a no ser que su browsing sea del tipo específico, en cuyo caso tendrá una idea concreta de antemano de lo que está buscando y en esta situación el sistema le ayuda simplemente con su agilidad. El problema lo tenemos realmente con el browsing general, porque en éste es preciso aproximar al usuario desde su vaga necesidad informativa a una o varias materias específicas que la satisfacen.

Una vez más la mejora de las posibilidades de recuperación a través de materias pasa por la determinación del nivel de especificidad de cada una de esas materias, y esto sin olvidarnos de que lo específico del significado de un documento puede estar relacionado con la organización en áreas temáticas de la información que antes comentamos. Por otra parte, las operaciones de browsing serán posibles cuando se puedan mostrar los documentos agrupados en base a su pertenencia a áreas temáticas comunes. Por estas razones me planteé la necesidad de estudiar la teoría del análisis de cluster y sus aplicaciones con el fin de utilizar algunos de sus enfoques para el desarrollo de un método de recuperación temática mejorado.

El análisis de cluster lo forman un conjunto de técnicas que permiten la identificación de objetos similares en un espacio multidimensional. En realidad estas técnicas no se aplican únicamente en el campo de la recuperación de información, sino que son de uso corriente en muchas ciencias, especialmente en las biológicas. Quienes se ocupan específicamente de esta disciplina estudian formalmente los algoritmos y métodos para clasificar todo tipo de objetos, para lo que se requiere que hayan sido descritos previamente en base a una serie de características, lo que quiere decir que el objeto de estos análisis es la mejor organización de los objetos a partir de la descripción formal de sus características y de las relaciones entre ellos que éstas implican [Jain 88]. La organización así obtenida supone la generación de una serie de cluster o clases de objetos que no son más que conjuntos de entidades agrupadas en razón de su similaridad [Everitt 74].

En el campo de la recuperación de información la hipótesis en la que se basa el análisis de cluster según Van Rijsbergen se puede exponer así: los documentos muy similares entre sí tienden a ser relevantes para las mismas búsquedas (pág. 45) [Rijsbergen 79]. Según esta hipótesis, aquellos documentos que tienen más materias comunes son potencial respuesta de las mismas demandas de los usuarios, por lo que si estuvieran agrupados antes de su recuperación se facilitarían las búsquedas [Ellis 90]. Cierto es que la mayoría de los trabajos en este campo han perseguido la reducción de los tiempos de respuesta más que la mejora de la calidad de la recuperación [Can 89, El-Hamdouchi 89, Arenas 91], pero tampoco es menos cierto que se han realizado algunos intentos de utilización de estas técnicas para mejorar la efectividad de las recuperaciones [Jardine 71, Croft 80]. Es precisamente en esta línea en la que se desarrolla mi propuesta, la utilización de un procedimiento de cluster para tratar de incrementar la eficacia de las recuperaciones por materias.

En el análisis de cluster se reconocen normalmente dos tipos de algoritmos o técnicas para la generación de los clusters, los jerárquicos y los no jerárquicos [Willett 88]. Unos y otros han sido estudiados por los expertos para su aplicación en la recuperación de documentos, pero todos los autores coinciden en que son los heurísticos, dentro de los no jerárquicos, los que con más frecuencia han sido aplicados a este campo. La característica esencial de este método es el escaso costo informático que supone su aplicación, dado que no es preciso el conocimiento previo a la generación del cluster, por parte del sistema, de los coeficientes de similaridad existentes entre los distintos documentos que componen la base de datos. Los métodos heurísticos se basan en el principio de que el sistema conoce previamente las similaridades posibles entre los documentos. Este conocimiento se hace posible gracias a la existencia de un patrón de similaridades que es utilizado por el método para decidir la inclusión de cada documento en el cluster que le corresponda [Salton 71]. Este método tiene algunas característica singulares que conviene resaltar:

- La clasificación de los documentos resultante de la aplicación del método puede ser arbitraria desde un punto de vista lógico, lo que haría inviable la realización de cualquier tipo de browsing.
- El carácter apriorístico que tiene el patrón de similaridades hace que sea muy probable la distribución irregular de los documentos en los clusters, lo que podría plantear problemas de gestión de las recuperaciones.
- Esta distribución irregular de los documentos en la estructura de clasificación puede terminar conduciendo a la redistribución de los documentos en una nueva organización de cluster.

Estas características del método deberán ser tenidas en cuenta porque algunas de ellas podría impedir su utilización con los fines pretendidos. En cualquier caso, un algoritmo de cluster debe cumplir una serie de condiciones para garantizar su efectividad [Arenas 91, Can 89, Willett 88]:

a) La distribución de los objetos en los cluster debe ser independiente del orden en que éstos sean introducidos en la base de datos.

b) La inclusión de cualquier nuevo documento en la base se realizará de manera rápida y poco costosa.
c) La distribución de los documentos en los clusters debe ser homogénea.
d) La estructura de cluster generada facilitará la recuperación eficaz de los documentos.

En torno a esta última condición me parece necesario hacer alguna precisión antes de continuar. Ya he señalado anteriormente la relación tan estrecha que en opinión de muchos autores existe entre la ponderación de los términos de búsqueda y la eficacia en las recuperaciones. Así mismo, se ha señalado que es posible formalizar una relación estable entre la frecuencia de uso de los términos de indización y su nivel de especificidad. Esta relación, por otra parte, se ha establecido en la mayoría de las propuestas realizadas como inversa entre el nivel de especificidad del término

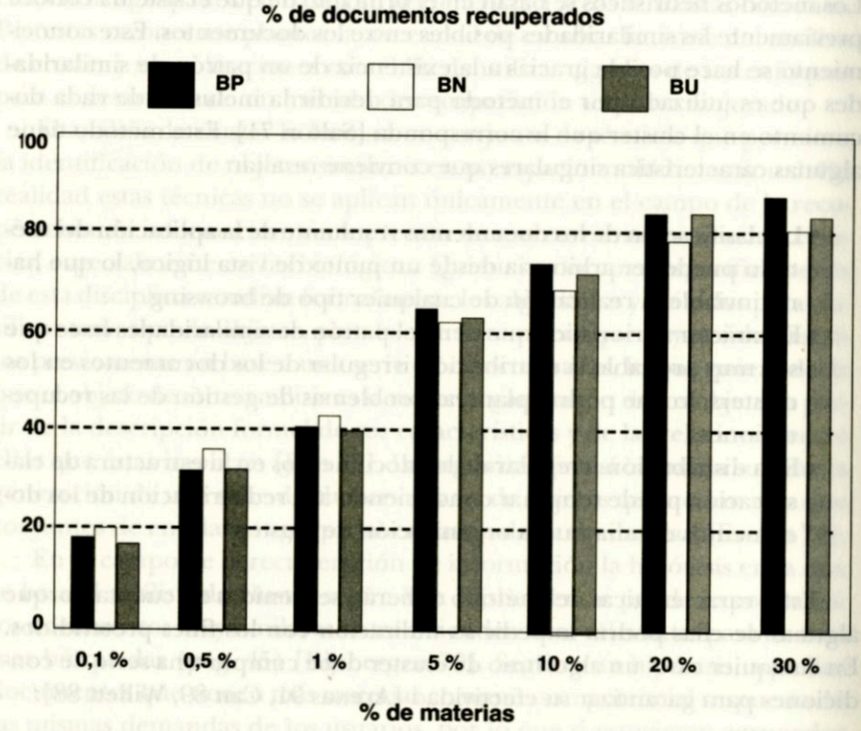


Gráfico 20: *Clases de frecuencias de materias.*

entendido como peso y su frecuencia. Aunque todas estas afirmaciones están suficientemente justificadas tanto desde un punto de vista teórico como práctico, alguno de los problemas que la utilización de estas técnicas pretendían resolver no han sido resueltos todavía.

Por una parte, la excesiva utilización de ciertas materias tanto en las búsquedas como en las indizaciones hace que con un número reducido de ellas se recupere gran cantidad de documentos, y por tanto sea imposible la reducción del tamaño de los conjuntos de documentos recuperados aunque los pesos asignados a estas materias sean muy bajos. En la gráfica donde relaciono los porcentajes de documentos recuperados con los de materias necesarias para recuperarlos, partiendo de las más frecuentes, se puede apreciar la vigencia de este problema en tres bibliotecas españolas, la Biblioteca Nacional, una universitaria y una pública. En los tres casos testados, ya con el 5% de las materias existentes en los catálogos se recuperan más del 60% de los documentos, y con el 20% de las materias sólo dejamos sin recuperar el mismo porcentaje de documentos. Esta situación, que debe de tener su origen en la confluencia de diversos factores, en los que no voy a entrar aquí, debe ser, al mismo tiempo, una de las causas fundamentales de la sobreabundancia informativa —information overload— obtenida por los usuarios de los OPAC que buscan a través de materias, y por consiguiente, del abandono de esta práctica [Larson 92b].

Por otro lado, estas técnicas de ponderación asignan los mismos pesos a todas aquellas materias que tiene la misma frecuencia de uso. Con lo que se consigue una ponderación que, desde el punto de vista del conjunto de la base de datos, es perfecta, puesto que lo más frecuente es menos ponderado sea cual sea su significado. Sin embargo, si consideráramos sólo una parte del catálogo, en el momento de calcular los pesos de las materias es evidente que los pesos serían distintos, y si esta parte no fuera determinada aleatoriamente sino en función de los contenidos de los documentos, la ponderación de las materias resultantes podría reflejar mejor el nivel de especificidad perseguido. En definitiva, lo que pretendo con esta especulación inicial es desarrollar un método que me permita dar una formalización válida al principio tantas veces defendido de que para calcular con eficacia la ponderación es necesario hacer intervenir no sólo frecuencias absolutas sino también relativas [Wong 92].

Para avanzar en el desarrollo de un modelo de recuperación que tuviera en cuenta todos estos aspectos del problema en el marco del análisis de cluster, empecé por estudiar la forma de generación de los cluster en un catálogo de una biblioteca cualquiera. En mi opinión, cualquier sistema automático de generación de cluster en un catálogo debería basarse en la información clasificatoria que los bibliotecarios incluyen en las referencias bibliográficas. Por las siguientes razones:

1) Porque esta información es muy precisa desde un punto de vista lógico cuando es general y pierde parte de su precisión cuando es específica. Es decir, un bibliotecario puede cometer errores al clasificar cuando los números que asigne, cuando utiliza un lenguaje de notación decimal, tengan muchos dígitos, pero los primeros dígitos asignados serán casi siempre correctos. Es la peculiaridad jerárquica de los sistemas clasificatorios lo que nos permite establecer hasta dónde en cada notación vamos a considerar para asignar a un cluster u otro las materias pertenecientes a un documento. Esto vendría a querer decir que es difícil equivocarse al identificar un libro como de matemáticas, pero más fácil al decir que analiza el álgebra de Von Neumann, por lo que la información relativa a lo general puede ser utilizada con más garantías que la relativa a lo más específico en las notaciones de CDU.

2) Porque esta información es suficiente para organizar la estructura de cluster que se quiera, por compleja que ésta sea. En el caso que yo propondré se ha generado un solo nivel de cluster pero en las notaciones de CDU existe información suficiente para organizar estructuras de cluster jerárquicos. Por otra parte esta información no requiere proceso previo alguno para su utilización. Basta con comparar cada notación con los valores de una tabla definida de antemano, para que el sistema sepa el cluster en el que deben ser incluidas las materias asociadas a esa notación. La sencillez del proceso garantiza que la respuesta del sistema sea rápida, aunque esto deberá asegurarse con la elección de una adecuada estructura de datos que soporte esta gestión.

3) Porque la existencia de una o varias notaciones asociadas a cada registro hace que la estructura de datos resultante sea muy flexible desde un punto de vista lógico, dado que las materias asociadas pasarán a formar parte de uno o varios clusters, lo que las hará recuperables desdetodos ellos. Si el sistema estuviera generado con el objeto de agilizar los accesos a los datos simplemente, sin contemplar la mejora de la calidad de las recuperaciones, esta característica sería negativa, pero para conseguir con este método una gestión de browsing que permita al usuario recorrer el conjunto de las materias divididas en subconjuntos temáticos a su elección, es imprescindible que éstas aparezcan incluidas en todos aquellos conjuntos en los que el indizador/clasificador las haya introducido de manera inconsciente. Esta versatilidad da lugar a la superación de la ponderación única para cada materia, puesto que al poder aparecer en varios cluster es lógico pensar que puedan ser ponderadas de distinta manera de acuerdo con el cluster al que pertenecen, lo que deberá mejorar considerablemente la determinación de los niveles de especificidad de las mismas.

4) Al estar el criterio de generación de los clusters determinado por el sistema de clasificación, es el propio usuario quien puede establecer la forma concreta de distribución de las materias confeccionando la tabla de comienzos de notaciones correspondiente. Esta adaptabilidad que el modelo permite hace que los clusters puedan ser generados en función de las características de la colección. Conviene señalar aquí que el modelo, teóricamente, podría funcionar con cualquier tipo de sistema clasificatorio siempre que éste sea jerárquico, pero las pruebas que yo he realizado han sido todas con registros clasificados mediante CDU. Respecto del problema del crecimiento desigual de los cluster, conviene señalar que la tabla-patrón puede ser modificada en cualquier momento y regenerada después la estructura de datos en la que se fundamenta a partir de la nueva tabla, lo que permite reequilibrar los tamaños de los clusters.

5) Muchas y relevantes investigaciones realizadas en los últimos años han puesto de manifiesto la necesidad de ligar la información de materias y la de clasificación con el fin de mejorar la recuperación en los OPAC [Hancock 87, Markey 86, Chan 86, Lancaster 89]. Estos estudios proponen distintos métodos para aprovechar de forma combinada la enorme cantidad de información que sobre el contenido de los documentos se encierra en los encabezamientos de materia, los números de clasificación e incluso las palabras de los títulos. Algunos de ellos incluso se plantean como en este caso la necesidad de utilizar técnicas de browsing ligadas a la gestión de las clasificaciones, aunque de forma bien distinta a como se propone aquí [Huestis 88]. En cualquier caso, la opinión unánime de estos autores es que el enorme esfuerzo de cumplimentación de los formatos estándar no se corresponde con las exiguas posibilidades de recuperación que ofrecen los OPAC que los utilizan.

Todo indica que el método utilizado es una variante del análisis de cluster heurístico, y que necesitamos definir una estructura de datos capaz de

soportar la gestión de la información que será generada y mantenida. Los elementos que intervienen en el proceso son los siguientes:

- Una tabla-patrón donde se encuentran las claves de CDU utilizadas. Esta tabla es idéntica a la utilizada para el cálculo de la variante de ponderación por frecuencias y número de clases.
- Un RDBMS que permita la gestión de una matriz en la que incluiremos por cada fila los datos de una entrada de materias, y por cada columna la frecuencia por cluster de esa materia, además de algunas columnas adicionales para almacenar frecuencias absolutas, número de cluster de cada entrada, pesos, etc.
- Un IRS que permita la gestión de la información bibliográfica calculando la frecuencia de aparición de cada uno de los términos, de tal manera que sea posible alternativamente realizar las recuperaciones tradicionales mediante operadores booleanos, de paso que se proporciona la información de frecuencias a la aplicación de gestión del browsing de materias.

El objeto de este conjunto es dotar al OPAC de un sistema de browsing que permita a los usuarios seleccionar las materias en función de su pertenencia a un área temática u otra que previamente ha sido elegida. Esta función es posible porque cuando se selecciona uno de los cluster el sistema muestra al usuario únicamente las materias que pertenecen a ese cluster. Para facilitar la elección por parte del usuario de una entrada de entre las mostradas, éste puede elegir el criterio de ordenación que prefiere: ponderado o alfabético.

Para terminar con este apartado describiré el procedimiento de ponderación y la forma en que debe ser implementado para facilitar el browsing de materias.

Como veremos a continuación, la función de ponderación que he desarrollado está basada en la existencia de tres variables que se relacionan entre sí para dar lugar a un peso. En este sentido el objetivo general de la función es el mismo de las analizadas anteriormente, permitir la ordenación de las entradas según su nivel de especificidad. Recuérdese, por otra parte, que la objeción fundamental puesta hasta ahora a las funciones del tipo idf es que no toman en consideración más que la frecuencia absoluta y no las posibles frecuencias relativas. Fue precisamente esta objeción la que me animó a considerar la posibilidad de hacer intervenir en el cálculo de la función la frecuencia de aparición del término de indización en cada cluster —fcn a partir de ahora, siendo n el número asignado al cluster—. Esta frecuencia relativa, convenientemente relacionada con la frecuencia total —ft a partir de ahora—, podría permitirnos determinar hasta qué punto una entrada es más específica que otra dentro de un cluster. La forma de relación entre ambas debería permitirnos establecer qué parte del conjunto de ocurrencias representado por la ft supone la fcn. Es decir, qué tanto por ciento de fcn supone la diferencia entre ft y fcn . Formalmente podríamos decir:

$$T = ((ft - fcn) * 100 / fcn) + 1$$

La variable T ordenada de forma ascendente, según lo argumentado anteriormente, podría darnos una primera aproximación de la especificidad de las materias. Pero al haber una gran cantidad de entradas en las que se da una coincidencia absoluta en los porcentajes, será necesario complementar la función a fin de conseguir que los pesos obtenidos fluctúen más. Aunque no es necesaria una exposición detallada de los motivos por los que se da esa coincidencia en el resultado, puesto que al ser la función un porcentaje es lógica la coincidencia, sí puede resultar útil poner un ejemplo: si aplicamos la función anterior a un conjunto cualquiera de materias el resultado será que todas aquellas cuya fcn sea idéntica a su ft, tendrán el mismo peso y éste será el mayor de la lista. Esto quiere decir que la función considera muy específicas todas aquellas entradas que sólo han sido usadas en un cluster, y poco específicas las que han sido usadas en más de un cluster. De cualquier forma, tanto si fcn = ft = 1, como si fcn = ft = 500, la función asignará siempre el mismo peso, lo que no resulta muy lógico, porque lo más probable es que la mayor frecuencia de utilización de una entrada en un cluster, no habiendo sido utilizada fuera de ese cluster, indique más especificidad.

Parece por tanto razonable añadir a la función algo que nos permita establecer diferencias de pesos entre las entradas cuyos valores de T sean iguales. Y asumiendo que a mayor fcn más especificidad, añadiremos esta variable ordenada de forma descendente. Para que la secuencia quede ordenada de forma ascendente y podamos combinarla con la expresión anterior la formalizaremos así:

$$F = Max(fcn) - fcn$$

Por último, la tercera variable que debe ser tenida en cuenta es el número de cluster en que está incluida la entrada. La justificación de la utilización de esta variable está en el hecho de que si tomamos dos entradas que tienen la misma frecuencia total —$ft_i = ft_j$ — y la misma frecuencia de cluster en uno de ellos —$fcn_i = fcn_j$—, pero que sin embargo una de ellas ha sido utilizada en muchas áreas distintas, mientras que la otra sólo en dos —$nc_i > nc_j$ —, a las dos materias, con lo dicho hasta ahora, les corresponderían los mismos pesos, pero en coherencia con el argumento de que el número de cluster en que aparece una materia es inversamente pro-

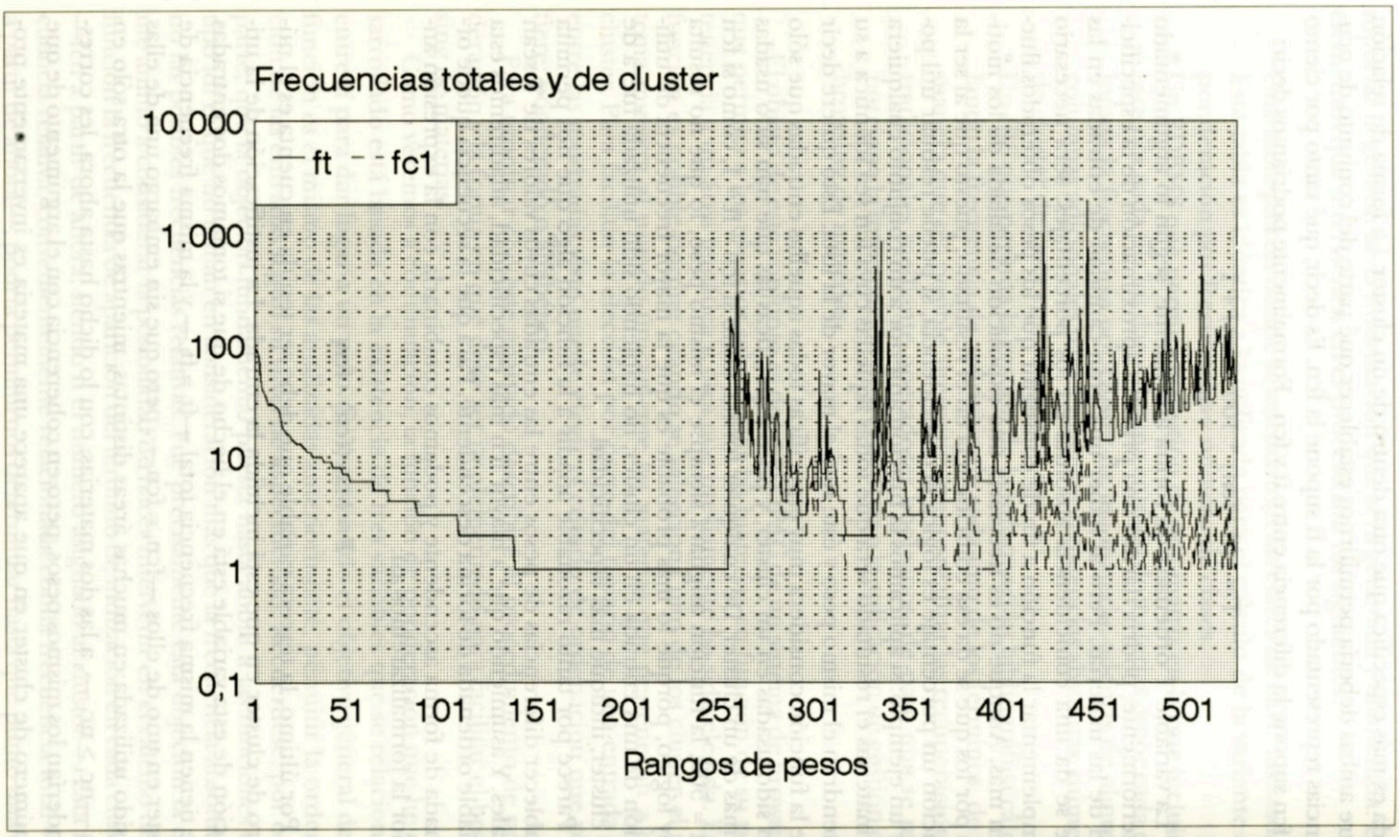


Gráfico 21: *Pesos por cluster (C1-Matemáticas).*

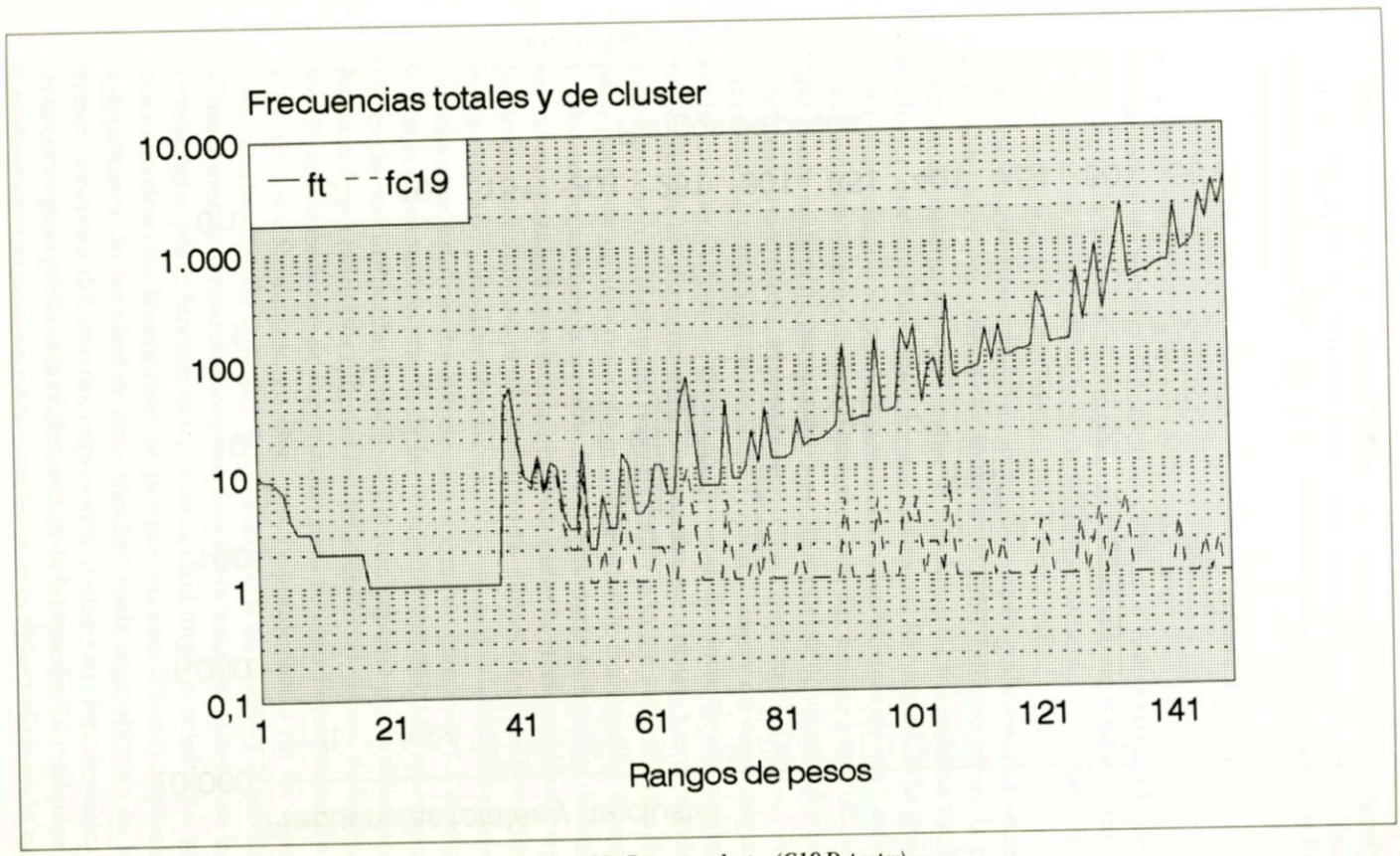


Gráfico 22: *Pesos por cluster (C19-Deporter).*

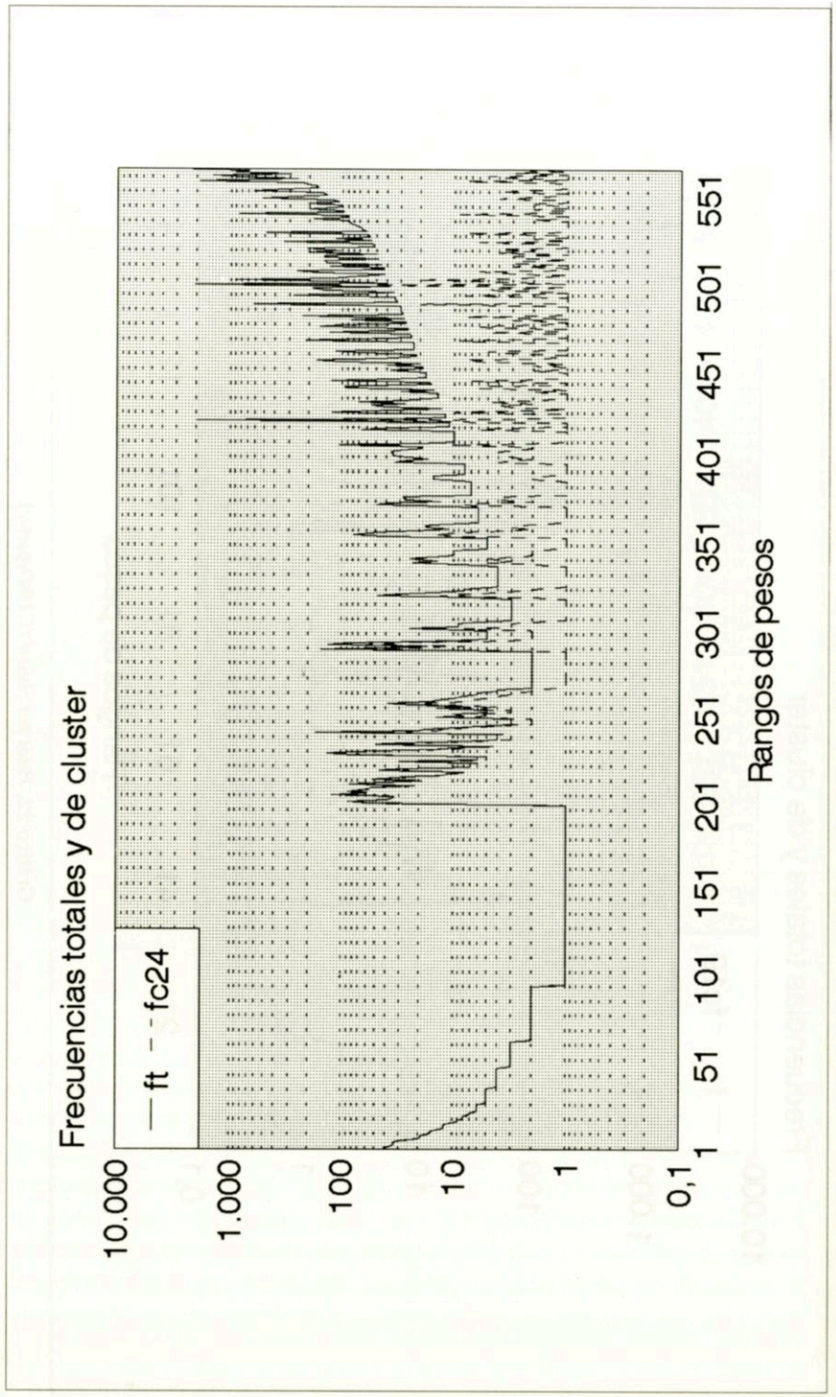


Gráfico 23: *Pesos por cluster (C24-Informática).*

porcional a su nivel de especificidad, las entradas que tuvieran igualdad de pesos según las expresiones anteriores serán ordenadas de forma ascendente en función del valor de la variable nc. Si formalizamos por fin la función completa:

$$p_i = T_i * F_i + nc_i$$

la ordenación ascendente del resultado obtenido nos permitirá hacer un browsing de las materias pertenecientes a distintas áreas temáticas en base a su nivel de especificidad en ese área, lo que deberá contribuir a la mejora de la recuperación de información en los OPAC.

Como en los casos anteriores, he representado gráficamente los resultados obtenidos con el fin de mostrar la relación existente entre la ponderación de las entradas que realiza la función y las frecuencias totales y de cluster que corresponden a estas materias. Representando gráficamente esta relación se pueden valorar las bases teóricas de las funciones anteriormente expuestas y compararlas con las breves especulaciones hasta ahora realizadas. Se aportan aquí únicamente los resultados pertenecientes a tres clusters, Matemáticas —C1—, Deportes —C19—, e Informática —C24—. La representación relaciona rangos de pesos asignados por la función —eje x— con las frecuencias de cluster y totales que corresponden a las materias pertenecientes a ese cluster. Las tres gráficas muestran cómo en la primera parte de cada curva fcn y ft coinciden, hasta que a partir de un determinado punto la variable fcn tiende a disminuir su valor respecto al de la variable ft. Por otra parte la función demuestra que no existe ninguna relación genérica entre la frecuencia total o relativa de una entrada y su peso, puesto que a rangos de pesos muy diferentes les corresponden frecuencias iguales y a rangos muy próximos frecuencias muy distintas. Por último, si tratamos de comparar los resultados de esta representación con la de las funciones del tipo idf, apreciaremos diferencias tan profundas que sólo se pueden explicar por la diferencia de objetivos formulados por quienes las definieron: en el caso de las funciones idf, la ponderación de las entradas para facilitar la ponderación posterior de los documentos; en este caso, la ponderación de las entradas como instrumento de browsing. Sin embargo, en mi opinión, la utilización de funciones de similaridad con vectores cuyos componentes sean los pesos asignados por la función descrita aquí, dará mejores resultados que las funciones descritas anteriormente, porque esta función es más precisa en la determinación del nivel de especificidad desde la perspectiva de los intereses concretos del usuario, expresados al elegir un área temática concreta como paso previo a la realización de la búsqueda. En cualquier caso, la evaluaciones necesarias para corroborar esta última afirmación no han

sido objeto de este trabajo. Quizá en el futuro convendría completar este estudio con test formales que permitan validar lo que desde la teoría parece indudable.

Para finalizar me gustaría hacer algunas observaciones sobre la implementación de esta solución. La formalización de la función de ponderación difiere de las anteriores porque se trata de la unión en una sola función de tres claves funcionales distintas. Por otra parte, formalizada de esta manera puede ser implementada sin problemas como clave de índice en cualquier RDBMS, lo que facilitaría la gestión de la tabla de entradas con sus correspondientes frecuencias de cluster. En realidad, un DBMS nos permitiría definir los veinticinco índices correspondientes a los clusters existentes en un solo fichero de índice multiclave. Todos estos índices actuarán como índices secundarios de la tabla con la matriz de frecuencias de cluster. La relación entre el DBMS que gestiona la información de ponderación de las entradas y los propios documentos se realiza mediante llamadas al IRS desde esta aplicación. Estas llamadas, como puse de manifiesto en el capítulo correspondiente, pueden hacerse de diversas formas, pero según lo expuesto hasta aquí no existirá la posibilidad en un sistema de esta naturaleza, como en proyectos anteriores —XLS, Merlín, Absys—, de establecer un reparto de funciones de gestión de información según los entornos. Esto significa que el sistema generado basará su funcionamiento en la interacción permanente de los recursos aportados por el IRS y los aportados por el RDBMS.

VII.G. Conclusiones

a) Es un concepto comúnmente aceptado que la superación de los métodos booleanos y sus prestaciones de recuperación están ligados a la capacidad de los sistemas automáticos de gestión de la información para ponderar las entradas de índice en base a su nivel de especificidad, lo que permitirá la generación de salidas ponderadas de los documentos según su nivel de relevancia respecto de la búsqueda definida.

b) En los primeros estudios de estas nuevas técnicas se puso mucho énfasis en la relación inversamente proporcional existente entre la frecuencia absoluta de los términos de indización y su peso, probablemente como consecuencia del influjo de los trabajos de Zipf. Sin embargo, muy pronto se comprobó que la ponderación podría ser mas efectiva si se hacía intervenir también en su cálculo la frecuencia relativa de los términos. Este valor, que en el

caso del lenguaje natural equivale a la frecuencia del término en el documento, para el lenguaje controlado puede obtenerse por la vía de la agrupación de los documentos en áreas temáticas. La frecuencia relativa sería la del término en el área.

c) La determinación de las áreas, clases o clusters de documentos se debe realizar por métodos automáticos, en el sentido de que no debe ser una tarea añadida a las ya realizadas por el catalogador en el proceso de actualización de la base de datos catalográfica. Por otra parte, la información clasificatoria incluida en las referencias bibliográficas puede muy bien servir de modelo para la generación de los grupos de documentos en áreas. A partir de estas agrupaciones será posible generar los clusters de las materias integradas en los documentos previamente agrupados.

d) La mayor parte de las funciones de ponderación que utilizan otras variables, además de las frecuencias absolutas, incluyen las frecuencias relativas. Pero en el caso que me ocupa he considerado más significativo desde el punto de vista de la especificidad de los términos el número de clases en que ese término estaba incluido. En todo caso, esta variable será considerada como inversamente proporcional al peso, es decir, su valor incrementa el de la frecuencia total. Y, en definitiva, si bien es verdad que la introducción de esa variable puede mejorar la función de ponderación, ésta no podrá reflejar adecuadamente la especificidad de los términos mientras su cálculo se realice mediante la comparación de cada frecuencia con las demás, utilizando la frecuencia máxima —número de documentos— como base. Planteado de otra forma, el nivel de especificidad de un término no será considerado en función de las características de la colección y de la forma en que ésta ha sido indizada, sino que, como hace el propio usuario al buscar, se utilizará un referente más próximo al término: el área temática a la que pertenece.

e) Para obtener salidas ponderadas de documentos en función de su grado de relevancia es necesario calcular coeficientes de similaridad entre el vector de la query y cada uno de los vectores de los documentos. Estas operaciones, muy costosas en términos de CPU, ofrecen como resultado una ponderación muy exacta de los documentos, que varía según la función utilizada para calcular el coeficiente. Las investigaciones más recientes se han encaminado al diseño de procedimientos que permitan reducir los costes de

estos procesos a fin de que puedan ser implementados en productos de gestión de información.

f) Mediante pruebas formales realizadas sobre el conjunto de materias utilizado por una biblioteca se han comparado las prestaciones de las cuatro funciones de similaridad más corrientes con el fin de determinar cuál de ellas resulta más adecuada para la recuperación de información. La evaluación ha sido realizada utilizando el método de cálculo del valor de discriminación de un término. Los resultados obtenidos me permiten concluir que la función que mejor se adapta a las exigencias de recuperación a través de lenguaje controlado es el producto escalar de dos vectores. Esta mejor adaptación es consecuencia de la prioridad que asigna a los términos cuyas frecuencias son más bajas frente a los que tienen frecuencias intermedias, en contra del comportamiento de las restantes funciones testadas.

g) Las dos críticas de más entidad hechas al modelo de espacio vectorial en el que yo baso mi análisis son la de la ortogonalidad de los vectores de la base y la del tamaño de los vectores al calcular su similaridad mediante producto escalar. En el primer caso, como ya ha quedado dicho, la independencia lineal de los vectores de la base es la expresión formal correcta de la relación que existe entre las distintas materias tal y como las considera una biblioteca cualquiera, por lo que, en mi opinión, no es necesario establecer coeficientes de correlación entre los vectores de la base. Por lo que afecta a la segunda crítica entiendo que los valores de similaridad obtenidos se pueden ver afectados por el hecho de que haya más o menos materias en un documento, pero esto no necesariamente tiene que verse como un defecto del sistema. Lo que tiene la posibilidad de ser más similar por tener más materias debe serlo. En cualquier caso, siempre existe la opción de introducir un factor de normalización que permita igualar la longitud de los vectores [Salton 88b].

h) De todas formas los problemas de la búsqueda por materias, puestos de manifiesto en investigaciones recientes —information overload y search failure—, se resuelven sólo parcialmente con las técnicas expuestas hasta aquí. Los investigadores insisten en la necesidad de mejorar los sistemas de interfaciación del sistema con los usuarios. En relación con esto, las técnicas de browsing parecen ser una buena solución para guiar al usuario en el trámite de

la elección de los términos de indización adecuados. Y esto no porque las técnicas anteriores no sean útiles, sino porque es necesario complementarlas con las correspondientes mejoras en los mecanismos de relación entre el usuario y el sistema.

i) El método propuesto para conseguir el objetivo anterior se desarrolla alrededor de una función que relaciona tres variables, las frecuencias total y de clase, y el número de clases en las que está incluida la entrada. El resultado de la aplicación de esta función es tal que consigue asociar cada entrada a una o varias áreas temáticas ponderándolas de acuerdo con su nivel de especificidad en cada área, lo que, a mi entender, resulta bastante coincidente con las expectativas del usuario. Así por ejemplo, si un usuario decide realizar una búsqueda en el área de Matemáticas, espera que las materias de este área le puedan ser mostradas separadas de las restantes, y es muy probable que agradezca que el sistema le diga en qué orden de especificidad se encuentran esas materias dentro de ese área. Y, en ningún caso, tendrá interés para él si el sistema le informa de que la entrada Ecuaciones diferenciales parabólicas es tan específica como El arte solutrense.

j) La información necesaria para aplicar esta función debe ser almacenada en una tabla-matriz de datos numéricos actualizada en el curso del proceso de verificación de cada uno de los documentos. La actualización se ejecuta mediante la determinación por parte del sistema de los cluster en que van a ser incluidas las materias del documento. Esta determinación se realiza comparando los datos de clasificación con el patrón que el sistema posee a tal efecto. Una vez efectuada la comparación se procede a la actualización de la matriz incrementando los contadores de cada una de las entradas por cluster.

k) El sistema propuesto mejora los procedimientos de browsing en uso y es perfectamente compatible con las técnicas de recuperación tradicionales y avanzadas de las que he hablado aquí. En todo caso, debe ser entendido como un valor añadido a los métodos descritos con anterioridad.

l) Las características descritas de los datos que son imprescindibles para obtener el máximo rendimiento de todas estas técnicas hacen inviable la utilización de forma independiente de ninguno de los tres SIA analizados hasta aquí. La solución más adecuada, da-

das las características de los procesos informáticos implicados y la naturaleza de los datos que se pretende procesar, será la integración en un Sistema Híbrido de un IRS y un RDBMS.

m) La concurrencia de recursos software necesaria para la realización de procesos como el de verificación de los registros hace necesario un altísimo nivel de integración entre los soportes básicos de la aplicación y el entorno. Aunque esto no es siempre un requisito difícil de cumplir dado el nivel de desarrollo de la ingeniería del software hoy día, sí que exigirá un esfuerzo importante de desarrollo hasta lograr la implementación completa del sistema.

APÉNDICE

```
#include <stdio.h> /* CALCULO DEL VD MEDIANTE PRODUCTO ESCALAR
*/
#include <stdlib.h>
#include <math.h>
main(int argc, char *argv[])
{
int a=0, b=0, c=0, d=0, e = 0, y=0;
FILE *in, *out, *in1, *in2;
float p[7][58000], m[10000];
short n[7][58000], x=0;
double pr_esc=0.0, acum_cos = 0.0, sum_1=0.0, sum_2=0.0, cos=0.0, work=0.0;
double sqrt(), cos_medio=0.0;
char linea[30];
                in = fopen(argv[1],"r"); /* fichero de documentos */
                out = fopen(argv[3],"w"); /* fichero de salida */
                in1 = fopen(argv[2],"r"); /* fichero de pesos */
                in2 = fopen(argv[4],"r"); /* fichero de centroide */
for(c=1; c<19733; c++)
                        {
                        fgets(linea,25,in2);
                        if(c % 2 != 0)
                                a = atoi(linea);
                        else{
                                m[a]=atof(linea);}
                                }
for(c=0; c<57543; c++)
                        {
                for(b=0; b<7; b++)
                                {
                        fgets(linea,10,in);
                        n[b][c]=atoi(linea);
                                }
                for(b=0; b<7; b++)
                                {
                        fgets(linea,10,in);
                        p[b][c]=atof(linea);
                                }
                        }
```

```
for (y=1; y<527; y++)
{
fgets(linea,20,in1);
x = atoi(linea);
for(a=0; a<57543; a++)
                {
                for(b=0; b<7 && n[b][a]>0; b++)
                        {
                        if (n[b][a] != x ){
                                pr_esc = pr_esc + (p[b][a]*m[n[b][a]]);}
                        }
                acum_cos = acum_cos + pr_esc;
                pr_esc = 0;
                }
if (y == 1) cos_medio = acum_cos;
printf("%d  %d %f Valor de discriminación: %lf\n",y,x,cos,(acum_cos -cos_me-
dio));
sprintf(linea,"%lf\n",(acum_cos-cos_medio));
fputs(linea, out);
acum_cos = 0;
}
fclose(in);
fclose(out);
fclose(in1);
fclose(in2);
}

#include <stdio.h> /* CALCULO DEL VD MEDIANTE COSENO */
#include <stdlib.h>
#include <math.h>
main(int argc, char *argv[])
{
int a=0, b=0, c=0, d=0, e = 0, y=0;
FILE *in, *out, *in1, *in2;
float p[7][58000], m[10000];
short n[7][58000], x=0;
double pr_esc=0.0, acum_cos = 0.0, sum_1=0.0, sum_2=0.0, cos=0.0, work=0.0;
double sqrt(), cos_medio=0.0;
char linea[30];
                in = fopen(argv[1],"r"); /* fichero de documentos */
                out = fopen(argv[3],"w"); /* fichero de salida */
                in1 = fopen(argv[2],"r"); /* fichero de pesos */
                in2 = fopen(argv[4],"r"); /* fichero de centroide */
for(c=1; c<19733; c++)
                        {
                        fgets(linea,25,in2);
                        if(c % 2 != 0)
```



```
                                a = atoi(linea);
                        else{
                                m[a]=atof(linea);
                                sum_2=sum_2+(m[a]*m[a]);}
                                }
                        work=sum_2;
for(c=0; c<57543; c++)
                        {
                for(b=0; b<7; b++)
                                {
                        fgets(linea,10,in);
                        n[b][c]=atoi(linea);
                                }
                for(b=0; b<7; b++)
                                {
                        fgets(linea,10,in);
                        p[b][c]=atof(linea);
                                }
                        }
for (y=1; y<527; y++)
{
fgets(linea,20,in1);
x = atoi(linea);
sum_2 = work - (m[x]*m[x]);
for(a=0; a<57543; a++)
                {
                for(b=0; b<7 && n[b][a]>0; b++)
                        {
                        if (n[b][a] != x ){
                                pr_esc = pr_esc + (p[b][a]*m[n[b][a]]);
                                sum_1 = sum_1 + (p[b][a]*p[b][a]);}
                        }
                if (pr_esc != 0)
                        cos = pr_esc / sqrt(sum_1 * sum_2);
                acum_cos = acum_cos + cos;
                cos = 0;
                sum_1 = 0;
                pr_esc = 0;
                }
if (y == 1) cos_medio = acum_cos;
printf("%d  %d %f Valor de discriminación: %lf\n",y,x,cos,(acum_cos -cos_me-
dio));
sprintf(linea,"%lf\n",(acum_cos-cos_medio));
fputs(linea, out);
acum_cos = 0;
}
```

```
fclose(in);
fclose(out);
fclose(in1);
fclose(in2);
}

#include <stdio.h> /* CALCULO DE VD MEDIANTE FUNCION DE DICE */
#include <stdlib.h>
#include <math.h>
main(int argc, char *argv[])
{
int a=0, b=0, c=0, d=0, e = 0, y=0;
FILE *in, *out, *in1, *in2;
float p[7][58000], m[10000];
short n[7][58000], x=0;
double pr_esc=0.0, acum_cos = 0.0, sum_1=0.0, sum_2=0.0, cos=0.0, work=0.0;
double sqrt(), cos_medio=0.0;
char linea[30];
                in = fopen(argv[1],"r"); /* fichero de documentos */
                out = fopen(argv[3],"w"); /* fichero de salida */
                in1 = fopen(argv[2],"r"); /* fichero de pesos */
                in2 = fopen(argv[4],"r"); /* fichero de centroide */
for(c=1; c<19733; c++)
                        {
                        fgets(linea,25,in2);
                        if(c % 2 != 0)
                                a = atoi(linea);
                        else{
                                m[a]=atof(linea);
                                sum_2=sum_2+(m[a]*m[a]);}
                                }
                        work=sum_2;
for(c=0; c<57543; c++)
                        {
                for(b=0; b<7; b++)
                                {
                        fgets(linea,10,in);
                        n[b][c]=atoi(linea);
                                }
                for(b=0; b<7; b++)
                                {
                        fgets(linea,10,in);
                        p[b][c]=atof(linea);
                                }
                        }
for (y=1; y<527; y++)
{
```



```
fgets(linea,20,in1);
x = atoi(linea);
sum_2 = work - (m[x]*m[x]);
for(a=0; a<57543; a++)
                {
                for(b=0; b<7 && n[b][a]>0; b++)
                        {
                        if (n[b][a] != x ){
                                pr_esc = pr_esc + (p[b][a]*m[n[b][a]]);
                                sum_1 = sum_1 + (p[b][a]*p[b][a]);}
                        }
                if (pr_esc != 0)
                        cos =2 * pr_esc / (sum_1 + sum_2);
                acum_cos = acum_cos + cos;
                cos = 0;
                sum_1 = 0;
                pr_esc = 0;
                }
if (y == 1) cos_medio = acum_cos;
printf("%d  %d %f Valor de discriminación: %lf\n",y,x,cos,(acum_cos -cos_me-
dio));
sprintf(linea,"%lf\n",(acum_cos-cos_medio));
fputs(linea, out);
acum_cos = 0;
}
fclose(in);
fclose(out);
fclose(in1);
fclose(in2);
}


#include <stdio.h> /* CALCULO DE VD MEDIANTE FUNCION DE JACCARD
*/
#include <stdlib.h>
#include <math.h>
main(int argc, char *argv[])
{
int a=0, b=0, c=0, d=0, e = 0, y=0;
FILE *in, *out, *in1, *in2;
float p[7][58000], m[10000];
short n[7][58000], x=0;
double pr_esc=0.0, acum_cos = 0.0, sum_1=0.0, sum_2=0.0, cos=0.0, work=0.0;
double sqrt(), cos_medio=0.0;
char linea[30];
                in = fopen(argv[1],"r"); /* fichero de documentos */
                out = fopen(argv[3],"w"); /* fichero de salida */
```

```
                in1 = fopen(argv[2],"r"); /* fichero de pesos */
                in2 = fopen(argv[4],"r"); /* fichero de centroide */
for(c=1; c<19733; c++)
                        {
                        fgets(linea,25,in2);
                        if(c % 2 != 0)
                                a = atoi(linea);
                        else{
                                m[a]=atof(linea);
                                sum_2=sum_2+(m[a]*m[a]);}
                                }
                        work=sum_2;
for(c=0; c<57543; c++)
                        {
                for(b=0; b<7; b++)
                                {
                        fgets(linea,10,in);
                        n[b][c]=atoi(linea);
                                }
                for(b=0; b<7; b++)
                                {
                        fgets(linea,10,in);
                        p[b][c]=atof(linea);
                                }
                        }
for (y=1; y<527; y++)
{
fgets(linea,20,in1);
x = atoi(linea);
sum_2 = work - (m[x]*m[x]);
for(a=0; a<57543; a++)
                {
                for(b=0; b<7 && n[b][a]>0; b++)
                        {
                        if (n[b][a] != x ){
                                pr_esc = pr_esc + (p[b][a]*m[n[b][a]]);
                                sum_1 = sum_1 + (p[b][a]*p[b][a]);}
                        }
                if (pr_esc != 0)
                        cos = pr_esc / (sum_1 + sum_2 - pr_esc);
                acum_cos = acum_cos + cos;
                cos = 0;
                sum_1 = 0;
                pr_esc = 0;
                }
if (y == 1) cos_medio = acum_cos;
```



```
printf("%d  %d %f Valor de discriminación: %lf\n",y,x,cos,(acum_cos -cos_me-
dio));
sprintf(linea,"%lf\n",(acum_cos-cos_medio));
fputs(linea, out);
acum_cos = 0;
}
fclose(in);
fclose(out);
fclose(in1);
fclose(in2);
}
```

BIBLIOGRAFÍA

Arenas 91

ARENAS ALEGRÍA, L.: *Efectividad y dinamismo en la recuperación documental mediante análisis cluster*, Tesis doctoral, Universidad de Deusto, 1991.

Ashford 84

ASHFORD, J. H.: *Infromation storage and retrieval systems on mainframes and minicomputers*, Program, 18, 124-146, 1984.

Ashford 88

ASHFORD, J. H. y WILLET, P.: *Text retrieval and document databases*, London : Chartwell Bratt, 1988.

Bayer 72

BAYER, R. y MCCREIGHT, E.: *Organization and maintenance of large ordered indexes*, Acta infromatica, 1, 3, 173-189, 1981.

Belkin 87

BELKIN, N. J. y CROFT, W. B.: *Retrieval techniques*, ARIST, 22, 109-145, 1987.

Biru 89

BIRU, T., EL-HAMDOUCHI, A., REES, R. S. y WILLETT, P.: *Inclusion of relevance information in the term discrimination model*, Journal of documentation, 45, 2, 85-109, June 1989.

Blair 88

BLAIR, D. C.: An extended relational document retrieval model, Information processing and management, 24, 3, 348-371, 1988.

Bookstein 72

BOOKSTEIN, A.: *Double hashing*, JASIS, 23, 402-405, 1972.

Bookstein 74

BOOKSTEIN, A.: *Hash coding with a non-unique search key*, JASIS, 25, 232-235, 1974.

Bookstein 80

BOOKSTEIN, A.: *Fuzzy requests*, JASIS, 31, 3, 241-247,1980.

Bookstein 85

BOOKSTEIN, A.: *Probability and fuzzy-set applications to information retrieval*, ARIST, 20, 117-151, 1985.

Boss 82

BOSS, R. W.: *Automating library acquisitions : issues and outlook*, White Plains, NY : Knowledge Industry Publications, 1982.

Brs

BIBLIOGRAPHIC RETRIEVAL SERVICES, INC.: BRS/Search user's guide, London.

Burger 85
BURGER, R. H.: *Authority work : the creation, use, maintenance, and evaluation of authority records and files,* Littleton, CO : Libraries Unlimited, 1985.

Can 89
CAN, F. y OZKARAHAN, E. A.: *Dinamic cluster maintenance,* Infromation processing and management, 25, 3, 275-291, 1989.

Ceri 84
CERI, S. y PELAGATTI, G.: *Distributed databases : principles and systems,* New York, NY : McGraw-Hill, 1984.

Clayton 91
CLAYTON, M.: *Gestión de automatización de bibliotecas,* Madrid : Fundación Germán Sánchez Ruipérez, 1991.

Cleverdon 67
CLEVERDON, C. W.: *The Cranfield tests of index language devices,* Aslib proceedings, 19, 173-194, 1967.

Codd 70
CODD, E. F.: *A relational model of data for large shared data banks,* CACM, 13, 6, June 1970.

Codd 90
CODD, E. F.: *The relational model for database management : version 2,* Reading, MA : Addison Wesley, 1990.

Cooper 83
COOPER, W. S.: *Exploiting the maximun entropy principle to increase retrieval effectiveness,* JASIS, 34, 1, 31-39, 1983.

Cooper 88
COOPER, W. S.: *Getting beyond boole,* Information processing and management, 24, 3, 243-248, 1988.

Cove 88
COVE, J. F. y WALSH, B. C.: *Online text retrieval via browsing,* Information processing and management, 24, 1, 34-37, 1988.

Crawford 75
CRAWFORD, R. G.: *The computation of discrimination values,* Information processing and management, 11, 249-253, 1975.

Crawford 81
CRAWFORD, R. G.: *The relational model in information retrieval,* JASIS, 32, 51-64, 1981.

Crawford 84
CRAWFORD, W.: *MARC for library use : understanding the USMARC formats,* White Plains, NY : Knowledge Industry Publications, 1984.

Croft 80
CROFT, W. B.: *A model of cluster searchin based on classification,* Information systems, 5, 189-195, 1980.

Crouch 88
CROUCH, C. J.: *An analysis of aproximate versus exact discrimination values,* Information processing and management, 24, 1, 5-16, 1988.

Chan 86
CHAN, L. M.: *Library of Congress Classfication as an online retrieval tool : potentials and limitations,* Information technology and libraries, 5, 3, 181-192, 1986.

Chaumier 86
CHAUMIER, J.: *Systèmes d'information : marché et technologies,* Paris : Entreprise Moderne d'Edition, 1986.

Chen 77
CHEN, P.: *The entity-relationship aproach to logical database design,* Reading, MA : Addison-Wesley, 1977.

Dammers 68
DAMMERS, H. F.: *Aslib CIG research project : progress report,* Aslib proceeding, 20, 4, 218-232, 1968.

Date 87
DATE, C. J.: *A guide to Ingres,* Reading, MA : Addison Wesley, 1987.

Date 89
DATE, C. J.: *A guide to the SQL standard : a user's guide to the standard relational language SQL,* Reading, MA : Addison Wesley, 1989.

Date 90
DATE, C. J.: *An introduction to database systems, Vol. I, 5ª ed.,* Reading, MA : Addison-Wesley, 1990.

Date 90b
DATE, C. J.: *What is a domain?,* en Relational database writings 1985-1989, Reading, MA : Addison Wesley, 1990.

Date 90c
DATE, C. J.: *Defining data types in a database language,* en Relational database writings 1985-1989, Reading, MA : Addison Wesley, 1990.

Date 90d
DATE, C. J.: *Why duplicate rows are prohibited,* en Relational database writings 1985-1989, Reading, MA : Addison Wesley, 1990.

Date 90e
DATE, C. J.: *Referencial integrity and foreing keys,* en Relational database writings 1985-1989, Reading, MA : Addison Wesley, 1990.

Date 90f
DATE, C. J.: *What is a distributed database system?,* en Relational database writings 1985-1989, Reading, MA : Addison Wesley, 1990.

Deogun 88
DEOGUN, J. S. y RAGHAVAN, V. V.: *Integration of information retrieval and database management systems,* Information processing and managemente, 24, 3, 303-313, 1988.

Dimsdale 73
DIMSDALE, J. J. y HEAPS, H. S.: *File structure for an on-line catalogue of one million titles,* Journal of library automation, 6, 37-55, 1973.

Dobis 85
INTERNATIONAL BUSINESS MACHINES CORPORATION: *Dortmund library system : systems guide,* Irving, Texas : IBM, 1985.

Dobosz 81
DOBOSZ, J. y SZYMANSKI, B.: *An implementation of relational interface to an information retrieval system,* Information systems, 6, 3, 219-228, 1981.

Doyle 63
DOYLE, L. B.: *The microstatistics of text,* Information Storage and Retrieval, 1, 189-214, 1963.

Eastman 85
EASTMAN, C. M.: *Database management systems,* ARIST, 20, 91-115, 1985.
El-Hamdouchi 88
EL-HAMDOUCHI, A. y WILLETT, P.: *An improved algorithm for the calculation of exact term discrimination values,* Information processing and management, 24, 1, 17-22, 1988.
El-Hamdouchi 89
EL-HAMDOUCHI, A. y WILLETT, P.: *Comparison of hierarchic agglomerative clustering methods for document retrieval,* The computer journal, 32, 3, 220-227, 1989.
Ellis 90
ELLIS, D.: *New horizons in information retrieval,* London : Library association, 1990.
Evans 83
EVANS, P. W.: *Barcodes, readers and printers for library applications,* Program, 17, 3, 160-171, July 1983.
Everitt 74
EVERITT, B. S.: *Cluster analysis,* New York : John Wiley & Sons, 1974.
Farley 91
FARLEY, L.: *Library resources on the internet : strategies for selection and use,* Documento obtenido vía FTP de la NSF, August 1991.
Folk 87
FOLK, MICHAEL J. y ZOELLICK, Bill: *File structures : a conceptual toolkit,* Reading, MA : Addison- Wesley, 1987.
Fox 88
FOX, E. A. y KOLL, M. B.: *Practical enhanced boolean retrieval experiences with the Smart and Sire systems,* Information processing and management, 24, 3, 257-267, 1988.
Ftrs 89
FREE TEXT RETRIEVAL SYSTEMS : A REVIEW AND EVALUATION: London : Taylor Graham, 1989.
Galacsi 86
GROUPE GALACSI: *Les systèmes d'information : analyse et conception,* Paris : Bordas, 1986.
Gare 84
IFLA: *Guidelines for authority and references entries,* London : IFLA, 1984.
Gredley 90
GREDLEY, E. y HOPKINSON, A.: *Exchanging bibliographic data : MARC and other international formats,* London : Library Association, 1990.
Grieder 78
GRIEDER, T.: *Acquisitions : where, what, and how,* Westport, CO : Greenwood Press, 1978.
Grosch 76
GROSCH, A. N.: *Serial arrival predicting coding,* Information Processing and management, 12, 2, 141-146, 1976.
Hancock 87
HANCOCK, M.: *Subject searching behaviour at library catalogue and at the shelves,* Journal of documentation, 43, 4, 303-321, December 1987.

Hancock 89
HANCOCK-BEAULIEU, M.: *Online catalogues : a case for the user,* en The online catalogue, London : Library association, 1989.
Hanson 82
HANSON, O.: *Desing of computer data files,* Rockville, MD : Computer science press, 1982.
Heaps 78
HEAPS, H. S.: *Information Retrieval : computational and theoretical aspects,* New York : Academic Press, 1978.
Held 78
HELD, G. y STONEBRAKER, M: *B-trees reexamined,* Communication of the ACM, 21, 2, 139-143, 1978.
Hickey 89
HICKEY, T. B.: *The experimental library system (XLS),* Annual review of OCLC research July 1988-June 1989, 1989.
Hickey 90
HICKEY, T. B.: *Experimental library system,* Annual review of OCLC research July 1989-June 1990, 1990.
Hildreth 85
HILDRETH, C. R.: *Online public access catalogs,* ARIST, 20, 233-285, 1985.
Hildreth 87
HILDRETH, C. R.: *Beyond boolean : designing the next generation of on-line catalogs,* Library trends, 35, 4, 647-667, Spring 1987.
Hildreth 89
HILDRETH, C. R.: *General introduction; OPAC research: laying the grounwork for future OPAC desing,* en The online catalog, London : Library association, 1989.
Hípola 91
HIPOLA, P. y MOYA, F. DE: *Proyectos EDI y normalización documental,* Revista española de documentación científica, 14, 408-419, 1991.
Hopkinson 77
HOPKINSON, A.: *Merlin for the cataloguer,* Aslib proceeding, 29, 8, 284-294, 1977.
Hsiao 70
HSIAO, D. y HARARY, F.: *A formal system for information retrieval from files,* Communication of the ACM, 13, 67-73, 1970.
Huestis 88
HUESTIS, J. C.: *Clustering LC Classification numbers in an online catalog for improved browsability,* Information technology and libraries, 7, 4, 381-393, 1988.
Ibermarc
BN: *Especificaciones del formato IBERMARC,* En publicación, Biblioteca Nacional (Madrid), 1991.
Isbd 77
IFLA: *ISBD(G) : general international standard bibliographic description,* London : IFLA, 1977.
Iso 10160
ISO: *Information and documentation - OSI - Interlibrary loan application service definition,* Geneve : ISO, 1991.

Iso 10161
ISO: *Information and documentation - OSI - Interlibrary loan application protocol specification,* Geneve : ISO, 1991.
Iso 10162
ISO: *Information and documentation - Search and retrieve application service definition for OSI,* Geneve : ISO, 1992.
Iso 10163
ISO: *Information and documentation - Search and retrieve application protocol specification for OSI,* Geneve : ISO, 1992.
Iso 2709
ISO: *Documentation - Format for bibliographic information interchange on magnetic tape,* Geneve : ISO, 1981.
Iso 5426
ISO: *Extension of the latin alphabet coded character set for bibliographic information interchange,* Geneve : ISO, 1983.
Iso 646
ISO: *Information processing - ISO 7 bit coded character set for information interchange,* Geneve : ISO, 1991.
Iso 6630
ISO: *Documentation - Bibliographic control characters,* Geneve : ISO, 1986.
Iso 82
ISO: *Concepts and terminology for the conceptual schema and the information base,* ISO/TC97/SC5-N695, March 1982.
Jain 88
JAIN, A. K. y DUBES, R. C.: *Algorithms for clustering data,* Englewood Cliffs, NJ : Prentice Hall, 1988.
Jardine 71
JARDINE, N. y VAN RIJSBERGEN, C. J.: *The use of hierarchical clustering in information retrieval,* Information storage and retrieval, 7, 217-240, 1971.
Jones 88
JONES, R. M.: *A comparative evaluation of two online public access catalogues,* London : British Library Research and Development Department, 1988.
Juilland 64
JUILLAND, A. y RODRÍGUEZ, E. C.: *Frecuency dictionary of spanish words,* London : Monton and C., 1964.
Keen 92
KEEN, E. M.: *Some aspects of proximity searching in text retrieval systems,* Journal of Information Science, 18, 89-98, 1992.
Kemp 88
KEMP, D. A.: *Computer-based knowledge retrieval,* London : Aslib, 1988.
Klug 82
KLUG, A.: *Equivalence of relational algebra and relational calculus query languages having agregate functions,* JACM, 29, 3, July 1982.
Knuth 73
KNUTH, D.: *The art of computer programming. Vol. 3, Searching and sorting,* Reading, MA : Addison-Wesley, 1973.

Kraft 91
KRAFT, H. K. y BOYCE, B. R.: *Operations research for libraries and information agencies,* San Diego, CA : Academic Press, 1991.
Kruglinski 83
KRUGLINSKI, D.: *Sistemas de administración de base de datos,* Madrid : Osborne/McGraw-Hill, 1983.
Kucera 67
KUCERA, H. y FRANCIS, N.: *Computational analysis of present-day american english,* Providence, RD : Brown University Press, 1967.
Lacroix 76
LACROIX, M. y PIROTTE, A.: *Generalized joins,* ACM SIGMOD, 8, 3, September 1976.
Lancaster 89
LANCASTER, F. W.: *Subject analysis,* ARIST, 24, 35-84, 1989.
Larson 91
LARSON, R. R.: *The decline of subject searching : long-term trends and patterns of index use in an online catalog,* JASIS, 42, 3, 197-215, 1991.
Larson 92
LARSON, R. R.: *Evaluation of advanced retrieval techniques in an experimental online catalog,* JASIS, 43, 1, 34-53, 1992.
Larson 92b
LARSON, R. R.: *Experiments in automatic Library of Congress Clasification,* JASIS, 43, 2, 130-148, 1992.
Leftkovitz 69
LEFTKOVITZ, D.: *File structures for on-line systems,* New-York : Spartan Books, 1969.
Luhn 57
LUHN, H. P.: *A statistical approach to mechanized encoding and searching of literary information,* IBM journal of research and developement, 1, 4, 309-317, October 1957.
Luhn 58
LUHN, H. P.: *The automatic creation of literature abstracts,* IBM journal of research and development, 2, 159-165, 1958.
Lynch 87
LYNCH, C. A.: *Extending relational database management systems for information retrieval aplications,* Ph. D. Tessis, Berkeley, CA : University of California, 1987.
Lynch 91
LYNCH, C. A.: *Nonmaterialized relations and the support of IR applications by relational datababase systems,* JASIS, 42, 6, 389-396, 1991.
Macleod 85
MACLEOD, I. A.: *Three aproaches to information retrieval,* Proceeding of the RIAO conference, Grenoble, 1985.
Macleod 90
MACLEOD, I. A.: *Storage and retrieval of structured documents,* Information processing and management, 26, 2, 197-208, 1990.
Macleod 91
MACLEOD, I. A.: *Text retrieval and the relational model,* JASIS, 42, 3, 155-165, 1991.

Markey 84
MARKEY, K.: *Subject searching in library catalogs : before and after the introduction of online catalogs,* Dublin, OH : OCLC, 1984.
Markey 86
MARKEY, K.: *Users and the online catalog : subject access problems,* En The impact on online catalogs, New-York : Neal-Schuman, 1986.
Maron 60
MARON, M. E. y KUHNS, J. L.: *On relevance, probabilistic indexing and information retrieval,* Journal of the ACM, 7, 216-244, 1960.
Martin 86
MARTIN, S. K.: *Library networks, 1986-87 : libraries in partnership,* White Plains, NY : Knowledge Industry Publications, 1986.
Matthews 84
MATTHEWS, J. R. y LAWRENCE, G. S.: *Further analysis of the CLR online catalog project,* Information technology and libraries, 3, 354-371, 1984.
Matthews 85
MATTHEWS, J. R.: *Directory of automated library systems,* New York : Neal-Schuman Publishers, 1985.
Matthews 87
MATTHEWS, J. R.: *Suggested guidelines for screen layouts and desing of online catalogs,* Library trends, 35, 4, 555-570, Spring 1987.
McCallum 85
MCCALLUM, S. H.: *Linked system project in the United States,* Ifla Journal, 11, 4, 1985.
McLeod 89
MCLEOD, D.: *1988 VLDB panel on future directions in DBMS research : a brief, informal summary,* ACM SIGMOD, 18, 1, March 1989.
Nelson 88
NELSON, M. J.: *Correlation of term usage and term indexing frequencies,* Information processing and management, 24, 5, 541-547, 1988.
Noreault 81
NOREAULT, T., MCGILL, M. y MATTHEW, B. K. = *A performance evaluation of similarity measures, document term weighting schemes and representation,* en Information retrieval research, London : Butterworths, 1981.
Osborn 80
OSBORN, A. D.: *Serial publication : their place and treatment in libraries,* Chicago : American Library Association, 1980.
O'Neill 88
O'NEILL, E. T. y VIZINE-GOETZ, D.: *Quality control in online datadases,* ARIST, 23, 125-156, 1988
Pearson 75
PEARSON, K. M.: *Minicomputers in the library,* ARIST, 10, 139-163, 1975.
Prasse 91
PRASSE, M. J.: *Interface desing procedure,* Annual review of OCLC research july 1990-june 1991, 17-18, 1991.
Prywes 72
PRYWES, N. S. y SMITH, D. P.: *Organization of information,* ARIST, 7, 103-158, 1972.

Ra 90
RA, M.: *Technology and resource sharing : recent developments and future scenarios,* Advances in library resource sharing, 1, 141-153, 1990.
Radecki 88
RADECKI, T.: *Trends in research on IR - the potential for improvements in conventional boolean retrieval systems,* Information processing and management, 24, 3, 219-227, 1988.
Raghavan 86
RAGHAVAN, V. V. Y WONG, S. K. M.: *A critical analysis of the vector space model for automatic indexing,* JASIS, 37, 5, 279-287, 1986.
Reid 90
REID, C.: *Comparing text, document, and relational database management systems,* Library software review, 80-82, March-April 1990.
Reynolds 89
REYNOLDS, D.: *Automatización de bibliotecas,* Madrid : Fundación Germán Sánchez Ruipérez, 1989.
Rijsbergen 76
VAN RIJSBERGEN, C. J.: *File-organization in library automation and information retrieval,* Journal of documentation, 32, 4,294-317, 1976.
Rijsbergen 77
VAN RIJSBERGEN, C. J.: *A theoretical basis for the use of co-ocurrence data in information retrieval,* Journal of documentation 33, 106-119, 1977.
Rijsbergen 79
VAN RIJSBERGEN, C. J.: *Information retrieval,* London : Butterwoths, 1979.
Robertson 77
ROBERTSON, S. E.: *Theories and models in information retrieval,* Journal of documentation, 33, 2, 136-148, June 1977.
Saffady 83
SAFFADY, W.: *Introduction to automation for librarians,* Chicago : American Library Association, 1983.
Sager 76
SAGER, W. K. y LOCKEMANN, P. C.: *Classification of ranking algorithms,* International forum on information and documentation, I, 41-46, 1976.
Salton 68
SALTON, G.: *Automatic information organization and retrieval,* New-York : McGraw-Hill, 1968.
Salton 71
SALTON, G.: *The SMART retrieval system,* Englewood Cliffs, NJ : Prentice Hall, 1971.
Salton 73
SALTON, G. y YANG, C. S.: *On the specification of term values in automatic indexing,* Journal of documentation, 29, 4, 351-372, December 1973.
Salton 75
SALTON, G.: *Dynamic information and library processing,* Englewood Clifs, NJ : Prentice-Hall, 1975.
Salton 75b
SALTON, G., YANG, C. S. y YU, C. T.: *A theory of term importance in automatic text analysis,* JASIS, 26, 1, 33-44, 1975.

Salton 76
SALTON, G., WONG, A. y YU, C. T.: *Automatic indexing using term discrimination and term precision measurements,* Information processing and management, 12, 43-51, 1976.
Salton 79
SALTON, G.: *Mathematics and information retrieval,* Journal of documentation, 35, 1, 1-29, March 1979.
Salton 83
SALTON, G. y MCGILL, M. J.: *Introduction to modern information retrieval,* New York : McGraw-Hill, 1983.
Salton 84
SALTON, G.: *The use of extended boolean logic in information retrieval,* SIGMOD record, 14, 277-285, 1984.
Salton 88
SALTON, G.: *A simple blueprint for automatic boolean query processing,* Information processing and management, 24, 3, 269-280, 1988.
Salton 88b
SALTON, G. y BUCKLEY, C.: *Term-weighting approaches in automatic text retrieval,* Information processing and management, 24, 5, 513-523, 1988.
Salton 89
SALTON, G.: *Automatic text processing : the transformation, analysis, and retrieval of information by computer,* Reading, MA : Addison-Wesley, 1989.
Schek 81
SCHEK, H.-J.: *Methods for the administration of textual data in databases systems,* en Information retrieval research, London : Butterworth, 1981.
Shultz 68
SHULTZ, C. K.: *H. P. Luhn : Pioneer of information science - Selected works,* London : Macmillan, 1968
Sibi 89
COSTILLA, C.: *Sistema de Informacion de Bibliotecas científicas Interconectadas y abiertas,* Madrid : SEUI/MEC, 1989.
Sparck Jones 72
SPARCK JONES, K.: *A statistical interpretation of term specificity and its application in retrieval,* Journal of documentation, 28, 1, 11-20, March 1972.
Stairs
IBM: Storage and information retrieval system/virtual storage (STAIRS/VS) Reference manual.
Standish 80
STANDISH, T. A.: *Data structure techniques,* Reading, MA : Addison-Wesley, 1980.
Tannehill 82
TANNEHILL, R. S. y HUSBANDS, C. W.: *Standars and bibliographic data representation,* Library trends, 31, 2, 283-314, 1982.
Tedd 87
TEDD, L. A.: *Computer-based library systems: a review of the last twenty one years,* Journal of documentation 43, 2, 145-165, June 1987.
Tedd 88
TEDD, L. A.: *Introducción a los sistemas automatizados de bibliotecas,* Madrid : Díaz de Santos, 1988.

Tomer 92
TOMER, C.: *Information technology standards for libraries,* JASIS, 43, 8, 566-570, 1992.
Tsichritzis 78
TSICHRITZIS, D.C. y KLUG A. (EDS.): *The ANSI/X3/SPARC DBMS framework : report of the Study Group Data Base Managemente Systems,* Information Systems 3, 1978.
Tuttle 83
TUTTLE, M.: *Introduction to serials management,* Greenwich, CO : Jai Press, 1983.
Usaut 87
USMARC format for authority data: Washington : Library of Congress, 1987.
Usclass 91
USMARC format for classification data: Washington : Library of Congress, 1991.
Ushold 89
USMARC format for holdings data: Washington : Library of Congress, 1989.
Usmarc 88
USMARC format for bibliographic data: Washington : Library of Congress, 1988.
Vizine-Goetz 91
VIZINE-GOETZ, DIANE: *Cataloging productivity tools,* Annual review of OCLC research july 1990-june 1991, 8-10, 1991.
Warheit 69
WARHEIT, I. A.: *File organization of library records,* Journal of library automation, 2, 20-30, 1969.
Willett 85
WILLETT, P.: *An algorithm for the calculation of exact term discrimination values,* Information processing and management, 21, 225-232, 1985.
Willett 88
WILLETT, P.: *Recent trends in hierarchic document clustering : a critical review,* Information processing and management, 24, 5, 577-597, 1988.
Wong 92
WONG, S. K. M. y YAO, Y. Y.: *An information-theoretic measures of term specificity,* JASIS, 43, 1, 54-61, 1992.
X/open 87
X/OPEN: *Relational database language (SQL) portability guide,* X/open, January, 1987.
Yee 91
YEE, MARTHA M.: *System desing and cataloging meet the user : user interfaces to online public access catalog,* JASIS, 42, 2, 78-98, 1991.
Yourdon 89
YOURDON, E.: *Modern Structured Analysis,* Englewood Cliffs, NJ : Prentice-Hall, 1989.
Zipf 49
ZIPF, H. P.: *Human behavior and the principle of least effort,* Cambridge, MA : Addison-Wesley, 1949.
Zunde 79
ZUNDE, P. y GEHL, J.: *Empirical foundations of information science,* ARIST, 14, 67-92, 1979.

Otras obras de Editorial ANABAD

Colección **ESTUDIOS**

— Luis García Ejarque: *La formación del bibliotecario en España.*

— Pedro López Gómez: *Organización de fondos de los Arhcivos Históricos Provinciales.*

— Gratiniano Nieto Gallo: *Panorama de los museos españoles y cuestiones museológicas.*

— George Anderla: *La información en 1985. Necesidades y recursos.*

— M.ª Isabel Morales; Alicia Girón, y Elena Santiago: *Nueva guía de las bibliotecas de Madrid.*

— José M.ª Berenguer Peña: *Guía de innovaciones tecnológicas para archivos, bibliotecas y centros de documentación.*

— Alicia Girón García: *Bibliotecas Populares de Madrid. Ensayo para la planificación de lectura pública en Madrid capital.*

— Isabel Bravo Juega: *Un capítulo fundamental de la museología: La seguridad en los museos.*

— Luis Caballero Zoreda: *Funciones, organización y servicios de un museo: el Museo Arqueológico Nacional de Madrid.*

— Teresa Molina Ávila; Vicenta Cortés Alonso: *Mecanización de protocolos notariales. Instrucciones para su descripción.*

— Félix de Moya Anegón: *Los sistemas integrados de gestión bibliotecaria* (Estructuras de datos y recuperación de información).

COLECCIÓN NORMAS

TÍTULOS PUBLICADOS:

— *Directrices para las entradas de autoridad y referencia.*

— ISBD(S): *Descripción bibliográfica internacional normalizada para Publicaciones seriadas.*

— ISBD(G): *Descripción bibliográfica internacional normalizada General.*

— ISBD (NBM): *Descripción bibliográfica internacional normalizada para materiales no librarios.*

— ISBD(A): *Descripción bibliográfica internacional normalizada para Publicaciones monográficas antiguas.*

— ISBD(M): *Descripción bibliográfica internacional normalizada para Publicaciones monográficas.*

— ISBD(PM): *Descripción bibliográfica internacional normalizada para Música impresa.*

— ISBD(CM): *Descripción bibliográfica internacional normalizada para Material cartográfico.*

— *Directrices internacionales para la catalogación de periódicos.*

— *Pautas para la aplicación de las ISBD a la descripción de partes componentes.*

— ISBD(CF): *Descripción bibliográfica internacional normalizada para Archivos de ordenador.*

— IFLA-FIAB: *Normas para bibliotecas públicas.*

— IFLA-FIAB: *Normas para escuelas de biblioteconomía.*

— ISBD(M): *Descripción bibliográfica internacional normalizada para su publicación monográfica.*

— ISBD(S): *Descripción bibliográfica internacional normalizada de publicaciones seriadas.*

— *Instrucciones para la redacción del inventario general, catálogos y registros en los museos servidos por el Cuerpo Facultativo de Archiveros, Bibliotecarios y Arqueólogos.*

ANABAD