



TESINA DE LICENCIATURA

TITULO: Integrando Roble al mundo Z3950

AUTORES: Diego Chobadindegui, Mauro Pisano, Diego Pochettino

DIRECTOR: Lic. Javier Diaz

CODIRECTOR: Lic. Claudia Queiruga

CARRERA: Licenciatura en Informatica Plan 90

Resumen

Roble es un catalogo colaborativo que concentra la información catalogada de cada una de las bibliotecas que pertenecen a la U.N.L.P. Periódicamente, cada biblioteca transfiere de manera estandarizada, la información que contiene en su catalogo, y mediante un proceso de verificación y actualización, esta información es incluida en Roble. El acceso a Roble para su consulta es mediante su dirección de internet <http://www.roble.unlp.edu.ar>. Mediante el uso de un formulario web se pueden ejecutar búsquedas y ver la información que se encuentra en el catalogo.

Z3950 es un protocolo cliente/servidor estándar de búsqueda, recuperación e intercambio de información catalogada creado por la NISO (National Information Standard Organization). Este protocolo, por su naturaleza, permite la interacción B2B (business to business), es decir, que una aplicación cliente, pueda conectarse al servidor Z3950 y operar sin intervención del usuario final.

El objetivo que se persiguió en el siguiente trabajo de grado surge del interés que existe por parte de la biblioteca de la UNLP de integrar su catálogo colaborativo Roble a la comunidad de servidores Z39.50. Esto permitirá ampliar la comunidad de usuarios de información catalogada que tiene Roble, actualmente limitada a las bibliotecas de la UNLP. El resultado de esta tesis de grado permitirá a cualquier cliente Z39.50 conectado a internet, utilizar la información que se encuentra en Roble.

Líneas de Investigación

La investigación se inició con el estudio del catálogo Roble y las tecnologías inherentes a la implementación del mismo.

Luego se investigó el protocolo Z3950.

El siguiente paso nos llevó a investigar el universo de aplicaciones de Código Abierto que nos permitieran, si existiera alguna, la modificación del Código para nuestro objetivo.

Por ultimo investigamos acerca de implementaciones específicas de librerías de acceso a catálogos CDS/ISIS y de implementaciones de Servidores Z3950

Trabajos Realizados

Se estudiaron los estándares ISO2709, MARC y MARC21, ISO23950, el catálogo CDS/ISIS y su familia de productos de desarrollo y producción. Se estudió, instaló y adaptó el framework Z3950 de código fuente abierto, JZKit.

Se modificó el framework JZKit y se desarrolló una librería de búsqueda específica utilizando las librerías de desarrollo del catálogo CDS/ISIS.

Se realizaron pruebas de integración con el proyecto KOHA-UNLP para que este último pueda incorporar en un futuro este servicio en su catálogo de búsqueda.

Conclusiones

Finalizadas las pruebas, logramos integrar exitosamente nuestro servidor Z3950 con un cliente construido por un equipo especializado de la facultad de informática de la UNLP.

Logramos conectar el catálogo Roble con el mundo Z3950. Adaptamos el servidor Z3950 de código fuente abierto, JZKit, para que atendiera pedidos y los resolviera consultando el catálogo ROBLE en formato CDS/ISIS.

Trabajos Futuros

A partir de aquí, las posibilidades son numerosas dada la naturaleza del código fuente abierto del proyecto, permitiendo las modificaciones y adaptaciones necesarias para implementar las búsquedas más variadas y los procesamientos de las mismas de acuerdo a las necesidades o servicios que se quieran brindar.

Fecha de la presentación: 06 / 2010

Integrando Roble al Mundo Z3950

**Diego Pochettino
Diego Chobadindegui
Mauro Pisano**

Agradecimientos

Queremos agradecer a las personas que nos ayudaron a llegar a esta instancia y que sin su participación no hubiera sido posible.

A Claudia Queiruga por su disposición, consejo y apoyo constante en toda etapa del desarrollo.

A Javier Diaz por su confianza no solo en el proyecto sino también en la posterior integración del mismo a otro proyecto de la UNLP.

A Norma Mangiaterra, Mariana Picchinini y Einar Lanfranco por su gran aporte en nuestro aprendizaje sobre el mundo bibliotecario y de información catalogada.

Por último a nuestras familias que fueron el sostén incondicional para nuestras largas noches y fines de semanas de tesis.

Contenido

Agradecimientos	2
CDS ISIS.....	6
Historia	6
Versiones de CDS/ISIS para DOS y Windows.....	7
CDS/ISIS para DOS	7
CDS/ISIS para Windows	9
WWWISIS	10
JavaISIS	10
Otras versiones.....	10
Generalidades del Sistema	11
La Base de Datos CDS/ISIS.....	11
Funciones del Sistema	12
Estructura de la base de datos	13
Archivos para definir una base de datos	13
Archivo Maestro	14
Archivo Invertido.....	15
Archivo Any	16
Relación Entre Archivos.....	16
Arquitectura del Sistema.....	17
Hojas de trabajo	18
Tabla de Definición de Campos (FDT)	18
Lenguaje de Formateo	19
Selectores de Campos	21
Expresiones	21
Expresiones Numéricas	22
Expresiones Alfanuméricas	23
Expresiones Booleanas.....	24
Tabla de selección de campos (FST).....	26
Ingreso y edición de datos	27
Recuperacion de Datos	28
Tipos de términos de búsqueda.....	28

Operadores de búsqueda.....	29
Operadores a nivel de campo y de proximidad	30
Sintaxis de las expresiones de búsquedas.....	30
Búsqueda en texto libre. (Búsqueda secuencial)	31
ISO 2709	33
Historia	33
Estructura básica	34
Segmento del Rotulo del Registro.....	34
Segmento del Directorio	35
Rotulo del Campo.....	36
Longitud del Campo	36
Posición del Campo	36
Segmento de los campos de datos variables	36
Formato De Archivos ISO Producidos y Aceptados Por CDS/ISIS.....	37
MARC.....	38
Que es MARC?.....	38
Evolución de MARC hacia MARC 21.....	38
Integración de formatos.....	39
UNIMARC.....	41
El Registro MARC.....	42
MARC para diferentes tipos de materiales	42
Normas MARC 21	42
Formato MARC 21 para Autoridades	43
Formato MARC 21 para Registros Bibliográficos	44
Formato MARC 21 para Datos de Existencia.....	48
Formato MARC 21 para Datos de Clasificación.....	48
Formato MARC 21 para Información de la Comunidad	48
Z3950	49
Introduccion	¡Error! Marcador no definido.
Historia	49
Servicios Z.....	50
Atributos Bib-1	51
¿Qué hace la Z39.50?	53
Como trabaja Z39.50.....	54

Arquitectura Cliente y servidor	56
Búsqueda Z.....	57
Otros acuerdos.....	58
Aplicaciones Z.....	59
Ventajas.....	60
ISIS DLL	61
Introduccion	61
Generalidades	62
Java Native Interface (JNI)	63
JZKIT	63
ROBLE	64
Desarrollo de la Tesina.....	65
Objetivos de la tesina.....	65
Primeros Pasos	66
Sistema Roble.....	66
Z39.50.....	69
Definición y puesta a punto.	69
Elección de la versión de JZKit.....	69
Comunicando el servidor con un cliente standard.	70
Aprendiendo a prueba y error.	71
Implementando la búsqueda en el catalogo CDS/ISIS.	72
Refinando el modulo para su integración con proyecto KOHA-UNLP.	73
Prueba de integración con proyecto KOHA-UNLP.	74
Conclusión.....	74
Futuros Trabajos.....	74
Aporte.....	75
Referencias	75
Referencias a sitios.....	75
Referencias Bibliograficas	76

CDS ISIS

Historia

CDS/ISIS es un acrónimo de Computerised Documentation Service / Integrated Set of Information Systems.

Es un paquete de software para sistemas de recuperación y almacenamiento de información generalizada desarrollado, mantenido y diseminado por la UNESCO. Fue liberado por primera vez en 1985 y desde entonces alrededor de 20000 licencias fueron entregadas por la UNESCO y una red internacional de distribuidores.

Es particularmente apropiado para aplicaciones bibliográficas y es usado para los catálogos de muchas bibliotecas de pequeño y mediano tamaño. Este software fue producido en diversos lenguajes como por ejemplo Árabe, Chino, Inglés, Francés, Alemán, Portugués, Ruso y Español entre otros. La UNESCO libero este software para propósitos no comerciales.

En el 2003 fue establecido por las bibliotecas como el estándar para desarrollo de sistemas de información en los países desarrolladores.

El CDS/ISIS original corría en un mainframe IBM y fue diseñado a mediados de los 70^{os} por Giampaolo Del Bigio para el sistema de documentación computarizado de la UNESCO (Computerised Documentation Service).

En 1985 se programó en Pascal una versión para mini y microcomputadores, que corría en una IBM PC bajo el sistema operativo MS-DOS.

En 1995 se conoció Winisis, la versión para Windows, que podía ejecutarse en una computadora sola o en una red de área local.

Luego se conoció el JavaISIS, cuyos componentes cliente/servidor permitían la administración de base de datos remotas a través de Internet y que estuvo disponible para Windows, Linux y Macintosh.

También se conoció a GenISIS, que permite a los usuarios producir formularios HTML para la búsqueda CDS/ISIS en bases de datos.

Durante el surgimiento de este paquete, apareció la reconocida ISIS_DLL que provee una API para el desarrollo de aplicaciones y también OpenISIS, otra librería con algunas mejoras sobre la anterior.

Estas librerías permitieron que la comunidad desarrolladora pudiera, a través de ellas, manipular un catálogo ISIS desde varios lenguajes de programación.

Versiones de CDS/ISIS para DOS y Windows

CDS/ISIS para Windows es un paquete de recuperación de información desarrollado por UNESCO (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura) que se ejecuta bajo plataforma Windows. La versión 1.4 (Revisión 19) se hizo circular en un CD-ROM a los distribuidores, en marzo de 2001 con fecha enero 2001. Esta versión siguió a la versión 1.3 que apareció en enero de 1999 y fue la primera versión del paquete distribuida oficialmente, aún cuando los titulares de licencias existentes han contado con versiones de prueba por algún tiempo. La versión 1.0 se lanzó en 1998 pero sólo fue una versión de prueba.

UNESCO lanzó el CDS/ISIS para microcomputadoras en 1985. Se denominó oficialmente: *Versión CDS/ISIS Mini-Micro*, usualmente conocido como *CDS/ISIS* o simplemente *ISIS*. En Latinoamérica, donde predomina el paquete para minicomputadoras *MINISIS*, (desarrollado en Ottawa, Canadá, por el Centro de Investigación y Desarrollo Internacional) la versión original de DOS se identificó siempre como *Micro-ISIS* y la versión Windows como *Micro-ISIS* o *WINISIS*.

CDS/ISIS para DOS

La primera versión del paquete consistió específicamente en cinco programas que se manejaban en forma separada pero actuaban sobre la misma base de datos. Uno de los programas incluía ingreso de datos y recuperación de la información y los restantes correspondían a las otras versiones del menú principal de las versiones

posteriores de DOS, Ordenación e Impresión, Definición de la Base de Datos, Servicios de Mantenimiento del Archivo Maestro y Utilitarios del Sistema.

En 1998 se lanzó la versión 2.0. Esta versión consistió en poco más que la integración de los diferentes programas en uno, con el agregado de programación Pascal para permitir anexar funciones adicionales al paquete básico.

El siguiente lanzamiento público correspondió a la versión 2.3 la cual incluía mejoras en la velocidad de la indización y en el espacio utilizado por los índices. Esto pudo llevarse a cabo, en parte, estableciendo dos índices, uno para términos cortos y otro para términos largos. Al mismo tiempo el paquete se hizo más elástico; hasta entonces, una base de datos podía dañarse irreparablemente si ocurría un corte de energía mientras se ingresaba un registro. La base de datos debía entonces restaurarse desde una copia de resguardo previa. Esta modalidad cambió ya que, a partir de la versión 2.3, los archivos que contienen una base de datos se cierran luego de que se modifica o agrega un nuevo registro.

Una característica posterior a esta versión fue la inclusión del archivo **syspar.par**, un archivo de parámetro del sistema. El programa, al momento de ejecutarse, revisa valores en dicho archivo. Este archivo permite que diversos grupos de archivos que el programa utiliza, se reubiquen en distintos subdirectorios o carpetas. El programa puede configurarse para que se abra en un idioma predeterminado (para lo cual existe una opción de menú disponible) o comenzar con una base de datos específica. También permite, para el caso de la versión en DOS, que un programa Pascal sea cargado en memoria y ejecutado antes de ingresar al programa principal. Se podrá hallar información adicional acerca de este archivo en la Sección 3.2. Al momento de ser introducido el **syspar.par**, se hizo posible establecer un archivo de parámetros para cada base de datos, de forma tal que los diferentes archivos que componen una base de datos puedan ubicarse en diferentes carpetas, unidades de disco o dispositivos. La razón inicial que motivó la inclusión de esta característica fue la de permitir que CDS/ISIS fuera utilizado como software de búsqueda para bases de datos almacenadas en CD-ROM. El CD-ROM es un dispositivo de sólo lectura y CDS-ISIS requiere que ciertos archivos sean habilitados para escritura. Estos archivos deben almacenarse en el disco rígido de la PC (pueden copiarse desde el CD-ROM): los archivos extensos que contienen datos e índices permanecen en el CD-ROM.

El archivo de parámetros **syspar.par** permitió también que CDS-ISIS se ejecutara en el entorno de una red y ya la versión 3.0 fue lanzada en mayo de 1992

como versión diseñada para trabajar en red. La versión 2.3 habilitaba a que el programa se ejecutase desde un entorno de red, especificando ciertos parámetros en el **syspar.par** como dispositivos de red. Sin embargo, un entorno de red permite acceso a múltiples usuarios, y por lo tanto, los archivos pueden dañarse si dos o más usuarios tratan de efectuar cambios en una misma base de datos al mismo tiempo. Un registro puede corromperse si se lo intenta grabar en forma simultánea. Este problema se subsanó al introducir la función de bloqueo de registros y base de datos a partir de la versión 3.0.

A medida que se desarrollaba CDS/ISIS, se utilizaban nuevas rutinas y funciones Pascal en su compilación las que fueron agregadas a una librería de programación Pascal.

Una importante aplicación del lenguaje Pascal llevó a la mejora del lenguaje de formateo de impresión. Esta característica se ha mantenido en ISIS para Windows. Los usuarios pueden desarrollar sus propios programas en Pascal y ampliar así las funciones que no figuran en el programa fuente.

CDS/ISIS para Windows

Desde 1989, cuando la mayoría de las computadoras eran entregadas con un nuevo sistema operativo llamado Microsoft Windows, fue inevitable que los usuarios de CDS/ISIS demandaran una versión de CDS/ISIS para Windows y así, UNESCO, comenzó a desarrollarla a partir de 1995.

Contrariamente a la versión en DOS, ISIS para Windows no está escrito en Pascal sino en una combinación de lenguajes de programación, principalmente lenguaje C y C++. Continuando con la filosofía de la versión en DOS, se dispone de una biblioteca de funciones del programa para aquellos programas que necesiten utilizar rutinas de los propios usuarios, en forma similar en la que Pascal era utilizado en la versión DOS. BIREME (Centro Latinoamericano y del Caribe de Información en Ciencias de la Salud) ha cooperado junto con UNESCO en el desarrollo de esta „biblioteca de funciones“ y gracias a ello se dispone actualmente de una Biblioteca de Enlaces Dinámicos (ISIS_DLL) que contiene estas rutinas. Las mismas pueden descargarse desde el sitio de BIREME en la dirección: <http://www.bireme.br>.

WWWISIS

La mención de BIREME nos obliga a destacar la existencia de una versión de CDS/ISIS que permite la búsqueda en bases de datos CDS/ISIS en modo cliente/servidor, siendo el „servidor“ un servidor Web y el „cliente“, una PC que acceda a Internet mediante cualquier navegador: Netscape y Microsoft Internet Explorer son los más conocidos. Se requiere poca experiencia por parte del usuario para realizar búsquedas en bases de datos, pero la implementación del WWWISIS en un servidor requiere conocimientos técnicos que usualmente podrán ser encontrados en organizaciones que tienen su propia conexión a Internet y, por lo tanto, cuentan con experticia en sistemas de computación y software dentro de la propia. WWWISIS se basa en un lenguaje de formateo de impresión producto de una combinación del lenguaje CDS/ISIS y del lenguaje de marcas de hipertexto (HTML).

Más allá de esto, la implementación de la base de datos es bastante diferente, aunque opera con datos que, originalmente provienen de una base CDS/ISIS en DOS o Windows. BIREME, incidentemente, reescribió la versión original en DOS en lenguaje de programación C y desarrolló C-ISIS, plataforma sobre la cual basó su trabajo para desarrollar las ISIS_DLL y WWWISIS.

JavaISIS

JavaISIS es otro método de operación cliente/servidor que permite acceder a bases de datos CDS/ISIS instaladas en cualquier máquina conectada a Internet, aún cuando ésta no se comporte como un servidor Web. Esto requiere que el programa WWWISIS servidor, desarrollado por BIREME, se encuentre instalado en la misma máquina.

Mayor información acerca de este tema se puede hallar en:

<http://web.tiscalinet.it/javaisis/>.

Otras versiones

UNESCO ha desarrollado versiones de CDS/ISIS para los sistemas operativos VMS y UNIX / LINUX. VMS ha caído ya en desuso, pero las versiones UNIX/LINUX siguen siendo utilizadas actualmente.

Generalidades del Sistema

La Base de Datos CDS/ISIS

El sistema CDS/ISIS, permite construir y administrar bases de datos estructuradas no numéricas, es decir, bases de datos constituidas principalmente por textos.

Aunque CDS/ISIS maneja con textos y palabras, y por lo tanto ofrece muchas de las características encontradas comúnmente en un procesador de palabras, hace mucho más que un simple proceso de textos. Esto se debe a que los textos que procesa CDS/ISIS están estructurados en datos elementales que define el usuario.

En términos generales se puede pensar en una base de datos CDS/ISIS como un archivo de datos relacionados que fueron generados para satisfacer los requerimientos de información de una comunidad de usuarios. Por ejemplo, un simple archivo de direcciones o un archivo más complejo como un catálogo de biblioteca o un directorio de proyectos de investigación. Cada unidad de información almacenada en la base consiste en un dato elemental, que describe una característica particular de la entidad a la que se refiere el registro. Por ejemplo, un banco de información bibliográfica contendrá datos acerca de entidades como libros, reportes, artículos de revistas, etc. Cada unidad en este caso, estará constituida por datos elementales como: autor, título, fecha de publicación, etc.

Los datos se encuentran almacenados en campos, a cada uno de los cuales se le asigna un rótulo numérico que indica su contenido. Puede pensarse que el número de rótulo es el nombre interno del campo que reconoce CDS/ISIS.

El conjunto de campos que contienen todos los datos de una unidad de información, se denomina registro.

Una característica particular de CDS/ISIS es que está diseñado específicamente para manejar campos (y consecuentemente registros) de longitud variable, permitiendo por lo tanto, una utilización óptima del espacio del disco, así como una completa libertad para definir la longitud máxima de cada campo.

Un campo puede ser opcional (es decir, estar ausente en uno o más registros), puede contener un solo dato, o dos o más datos de longitud variable. En este último caso, se entiende que el campo contiene subcampos, cada uno de los cuales está identificado por un delimitador de subcampo de dos caracteres que precede al dato correspondiente. Además, un campo puede ser repetible, es decir, cualquier registro puede contener más de una ocurrencia de ese campo.

Funciones del Sistema

Las principales funciones previstas por CDS/ISIS permiten:

- Definir bases de datos conteniendo los datos elementales requeridos
- Agregar nuevos registros en una base de datos
- Modificar, corregir o borrar registros existentes
- Construir automáticamente y mantener archivos para acceso rápido a los registros de cada base de datos de modo que haya una recuperación muy veloz
- Recuperar registros por su contenido mediante un lenguaje de recuperación amplio y poderoso
- Visualizar los registros o partes de los mismos de acuerdo a las necesidades del usuario
- Ordenar o clasificar los registros en cualquier secuencia deseada
- Imprimir catálogos completos, parciales y/o índices
- Desarrollar aplicaciones especiales usando las facilidades integradas de programación

Estas funciones se obtienen a través de 8 programas que proveen los servicios correspondientes, clasificados en dos grandes categorías: Servicios para el usuario que emplea bases de datos ya existentes; y servicios del sistema, diseñados para el administrador de las bases de datos, para crear nuevas bases o realizar actividades diversas en relación al sistema. Los servicios para el usuario requieren sólo un conocimiento básico del programa CDS/ISIS, en tanto que los servicios del sistema presuponen un conocimiento más profundo de los componentes del sistema, incluyendo la capacidad para programar computadoras.

Los cuatro servicios previstos para el usuario son los siguientes:

- ISISENT Ingreso y edición de datos
- ISISRET Recuperación de información
- ISISPRT Producción de impresos y reportes, tales como catálogos e índices
- ISISINV Mantenimiento del archivo invertido y funciones utilitarias

Los cuatro servicios del sistema permiten:

- ISISDEF Definir una nueva base de datos y/o modificar las definiciones de una existente
- ISISUTL Funciones utilitarias diversas, tales como crear y/o editar menús o mensajes, definir atributos de pantalla, etc.
- ISISXCH Intercambio de información con otros sistemas y utilitarios del archivo maestro
- ISISPAS Servicios avanzados de programación que permite desarrollar programas de aplicación propios, e integrarlos en CDS/ISIS

Estructura de la base de datos

Aunque las bases de datos CDS/ISIS pueden parecerse a simple vista como un archivo de información, en realidad consisten en varios archivos lógicamente interrelacionados pero físicamente diferentes. La administración de los archivos físicos es responsabilidad de CDS/ISIS y el usuario normalmente no necesita conocer su estructura en detalle para operar una base de datos. No obstante, algunos conocimientos básicos sobre los objetivos y funciones de los archivos principales asociados a una base de datos pueden ayudar a un mejor entendimiento del sistema.

Archivos para definir una base de datos

Antes de que una base de datos pueda ser utilizada, deben definirse ciertas características de la estructura de los registros y de su contenido, para que CDS/ISIS sepa cómo manejarla. El módulo ISISDEF permite crear o modificar la definición de una base de datos.

La definición de una base de datos CDS/ISIS consiste de los siguientes componentes, cada uno de ellos almacenado en un archivo diferente.

Tabla de definición de campos (FDT): La FDT define los campos que pueden existir en la base de datos y sus características.

Hoja(s) de trabajo de ingreso: Comprende una o más páginas que se presentan en la pantalla para crear o actualizar los registros de la base de datos. CDS/ISIS cuenta con un editor especial para crear estas hojas de trabajo.

Formato(s) de visualización: Los formatos definen los requerimientos detallados de presentación para mostrar los registros en la pantalla durante la búsqueda o para la generación de reportes impresos tales como catálogos e índices. CDS/ISIS cuenta con un poderoso lenguaje de formateo que permite visualizar el contenido de un registro en cualquier forma.

Tabla de selección de campos (FST): La FST define los campos de la base de datos que serán recuperables a través del archivo invertido. Definiendo tablas adicionales de selección de campos, pueden establecerse las especificaciones más usuales para la clasificación de los registros de la base de datos.

Archivo Maestro

El archivo maestro contiene todos los registros de una determinada base de datos, cada uno de las cuales consiste en un conjunto de campos de longitud variable. Cada registro se identifica con un número único, asignado automáticamente por CDS/ISIS al ser creado; este número se denomina: Número del archivo maestro o MFN (iniciales de Master File Number).

Con objeto de proporcionar un acceso rápido a cada registro del archivo maestro, CDS/ISIS asocia al archivo maestro un archivo especial denominado "Archivo de referencias cruzadas", que es en realidad un índice que suministra la ubicación de cada registro en el archivo maestro.

Pueden crearse, modificarse o borrarse los registros del archivo maestro por medio de las facilidades de ingreso y edición del módulo ISISENT.

Archivo Invertido

Aunque un registro del archivo maestro puede ser recuperado directamente por su número de MFN, a través del archivo de referencias cruzadas, se necesitan formas de acceso adicionales. Por ejemplo, en la recuperación de registros bibliográficos, es necesario tener acceso a los registros por autor, por materia o por cualquier otro dato que ocurra en el registro. El sistema CDS/ISIS permite tener virtualmente un número ilimitado de puntos de acceso a cada registro con la creación de un archivo especial denominado archivo invertido.

El archivo invertido contiene todos los términos que pueden usarse como puntos de acceso durante la recuperación de registros en una base de datos, y para cada término, la lista de referencias a los registros del archivo maestro de donde se extrajo el término. Al conjunto de todos los puntos de acceso para una base de datos determinada se le denomina diccionario. Se puede considerar el archivo invertido como un índice del contenido del archivo maestro.

Por ejemplo, cuatro registros del archivo maestro (con números MFN 18, 204, 766, y 1039) contienen el descriptor EDUCACION. La estructura lógica correspondiente en archivo invertido sería:

EDUCACION 18 204 766 1039

En este caso, EDUCACION es el punto de acceso (o término del diccionario), y cada referencia al registro donde aparece en el archivo maestro se denomina apuntador (posting)[1]. Debido a que normalmente bajo cada término hay indizados un número diferente de registros, el registro lógico de un archivo invertido es de longitud variable. Aquí también, para tener una recuperación rápida para cada punto de acceso, el archivo invertido, en realidad, consta de varios archivos físicos. CDS/ISIS permite la creación selectiva de archivos invertidos para cada base de datos. Se pueden seleccionar campos, subcampos o partes de ellos. Además, mediante la especificación de opciones apropiadas, se pueden extraer palabras individuales, frases o descriptores de los campos seleccionados.

El usuario define los elementos recuperables de una determinada base de datos por medio de la tabla de selección de campos (FST), que especifica los campos que serán invertidos y la técnica de indizado a usar en cada caso.

A diferencia de otros sistemas de recuperación basados en archivos invertidos, en donde existe un archivo invertido separado para cada campo recuperable, CDS/ISIS utiliza un sólo archivo invertido para cada base de datos. No obstante, debido a la estructura particular de este archivo, es funcionalmente equivalente a un sistema de archivos invertidos múltiples. En realidad cada término en el archivo invertido no sólo contiene los MFN, sino también información adicional que identifica de manera precisa el campo a partir del cual se extrajo el dato, así como la posición de la palabra dentro del campo.

La versión actual de CDS/ISIS prevé términos en el archivo invertido de hasta 30 caracteres de longitud. Los elementos de longitud mayor que la máxima son truncados antes de ser incluidos en el archivo invertido.

Archivo Any

Un tipo opcional de archivo, asociado con el archivo invertido, es el archivo ANY. Se utiliza en la recuperación para agrupar términos asociados. Un término ANY es un nombre colectivo asignado a un grupo de términos de búsqueda. Cuando se utiliza un término ANY en una búsqueda, se recupera el grupo con ese nombre, y los términos individuales asociados al término usado, son recuperados y procesados en forma automática para integrarlos con una operación OR.

El criterio para definir la conveniencia de establecer un término ANY es la probabilidad de que ocurra frecuentemente en las búsquedas. Los agrupamientos geográficos, en muchos casos satisfacen este criterio, pero en otros tipos de términos también es aplicable. Por ejemplo, si se crea el término ANY, se puede usar en la búsqueda el término colectivo: ANY Latinoamérica, que define los nombres de todos los países en Latinoamérica, en vez de tener que especificar el nombre de cada país.

Relación Entre Archivos

La relación lógica entre los archivos principales de una base de datos CDS/ISIS se comprende mejor analizando la forma en que se realiza la recuperación de

información. La recuperación en una base de datos se realiza especificando un grupo de términos de búsqueda que se tratan de localizar en el archivo invertido para obtener la lista de MFN asociada a cada término. Estas listas son manejadas por el programa de acuerdo con los operadores de búsqueda que se especificaron en la formulación de la búsqueda, hasta que al finalizar el proceso se llega a una sola lista llamada lista HIT, que corresponde a los MFN de los registros que satisfacen los requisitos de la expresión de búsqueda. Si en ese momento se solicita la visualización en la pantalla de los registros recuperados, CDS/ISIS lee del archivo maestro cada registro mencionado en la lista HIT, lo formatea de acuerdo con el formato especificado, y lo presenta en la pantalla.

El usuario puede conservar una o más listas HIT para ser usadas posteriormente para imprimir los registros recuperados usando el módulo ISISPRT. Una lista HIT que ha sido guardada en un archivo para uso posterior se denomina archivo SAVE

Arquitectura del Sistema

Un componente básico de CDS/ISIS es su sistema de menús, que permite utilizar los diversos servicios que ofrece el sistema. Sin embargo, para el manejo y operación de una base de datos deben aprenderse algunas técnicas que son específicas de CDS/ISIS, tales como el lenguaje de búsqueda y el de formateo. Las técnicas se han implementado a través de una serie de herramientas incorporadas en CDS/ISIS para este propósito. Por ejemplo, si se desea realizar una búsqueda en una base de datos, debe primero seleccionarse la opción apropiada en los menús, después formular la expresión de búsqueda que debe seguir las reglas del lenguaje de recuperación de CDS/ISIS. Por lo tanto, esta técnica debe conocerse previamente. Para escribir la expresión de búsqueda se utiliza el servicio de otra herramienta de CDS/ISIS denominada editor de campos.

Las técnicas se relacionan con el proceso intelectual empleado para transformar una necesidad particular de un usuario, al lenguaje específico de recuperación de información de CDS/ISIS (tal como sería el caso de recuperar información sobre efectos de la radiación solar sobre la fauna marina). Por su parte, las herramientas son instrumentos más mecánicos y generalmente tienen una aplicación más amplia y generalizada (por ejemplo, el editor de campos no es solamente usado para escribir las expresiones de búsqueda, sino también para crear o modificar registros).

Hojas de trabajo

Una hoja de trabajo define el diseño de la(s) pantalla(s) usada(s) para ingresar los datos. Es equivalente a un formulario prediseñado que debe ser rellenado por el usuario.

Como en un formulario prediseñado, una hoja de trabajo contiene un número de campos, cada uno de los cuales consiste en un nombre (o identificación) y un espacio vacío donde se escriben los datos correspondientes.

En algunos casos, uno o más campos pueden ya contener datos. Los datos que se presentan ya ingresados se denominan valores por defecto. Cuando la pantalla no es suficiente para contener una hoja de trabajo completa, esta se puede dividir en dos o más páginas y el usuario, utilizando los comandos apropiados puede pasar de una página a otra.

Hay dos tipos de hojas de trabajo: Hojas de trabajo del sistema y hojas de trabajo de ingreso.

Las hojas de trabajo del sistema son usadas por CDS/ISIS para solicitar los parámetros requeridos para ejecutar una función determinada.

Las hojas de trabajo de ingreso, se usan para crear o modificar los registros de una base de datos. Este tipo de hoja de trabajo contiene todos los campos que pueden estar presentes en un registro. Debido a que las hojas de trabajo de ingreso deben ser definidas para ajustarse a las necesidades de diferentes bases de datos, CDS/ISIS cuenta con un editor de hojas de trabajo que permite adaptarlas de acuerdo a los requerimientos de cada usuario.

Tabla de Definición de Campos (FDT)

La FDT aporta información sobre el contenido de los registros del archivo maestro de una base de datos dada. En particular, define los diversos campos que pueden estar presentes, y una serie de parámetros para cada campo.

La FDT es utilizada para controlar la creación de hojas de trabajos de ingreso para la base de datos, y para validar el contenido de los campos. La figura 5 muestra un ejemplo de una FDT, tal como la presenta el editor de líneas.

Tabla definición de campos (FDT) Base de Datos: CDS

ETQ	Nombre	Long	Tip	Rep	Delimitador/Patron
12	Entrada principal conferencia	300	X		npdz
24	Título	500	X		z
25	Edición	100	X		
26	Pie de imprenta	300	X		abc
30	Descripción física	100	X		abc
44	Serie	300	X	R	vz
50	Notas	500	X		
69	Palabras clave	1000	X		
70	Autores personales	100	X	R	
71	Entidades corporativas	300	X	R	
72	Conferencias	300	X	R	npdz
74	Título adicional	500	X	R	z
76	Títulos en otros idiomas	500	X	R	z

Figura 5: Ejemplo de tabla de definición de campos

Lenguaje de Formateo

El lenguaje de formateo permite al usuario definir con precisión los requerimientos de formateo de los registros de la base de datos. A través de este lenguaje puede seleccionarse uno o más datos elementales, en el orden que se desee, y en forma opcional, insertar los textos fijos que se deseen, p.ej. para rotular algunos o todos los campos, así como especificar los requerimientos de espaciado vertical y horizontal. A la colección de comandos de formateo en el lenguaje descrito en este capítulo se le denomina un formato. En general, un formato define un subconjunto del

registro de la base de datos, que puede entonces ser usado por CDS/ISIS para realizar una función dada. Aunque los formatos son principalmente usados para especificar la forma en que se visualizan los registros en la pantalla o en la impresora, también son ampliamente usados por el sistema cada vez que se desea realizar una operación sobre uno o más datos elementales. Por ejemplo, en una Tabla de Selección de Campos, se usa un formato para especificar a cuales datos se debe aplicar cierta técnica de indizado. El lenguaje de formateo es por lo tanto el núcleo de muchas operaciones de CDS/ISIS, y un uso eficiente de CDS/ISIS requiere de un conocimiento amplio de esta técnica.

Para un principiante, algunos formatos pueden parecer muy complejos, dando la impresión de que el lenguaje de formateo mismo sea complejo. En realidad, todos los formatos, aun los más complicados, están compuestos de uno o más comandos o declaraciones simples, separados por comas o espacios. La aparente complejidad deriva de que puede haber muchos de estos comandos en un formato. Por lo tanto, la clave para entender los formatos es analizar cada comando en forma individual.

Aunque todos los formatos se definen utilizando el mismo lenguaje de formateo, pueden clasificarse de acuerdo con el uso que se les pretende dar, del modo siguiente:

- **Formatos de visualización:** son usados para visualizar registros en la pantalla o para imprimirlos en una impresora (en este último caso se les conoce como formatos de impresión);
- **Formatos de extracción:** son usados en las FST para definir los datos a ser indizados.

Cuando CDS/ISIS procesa un formato, trabaja con tres objetos: un registro de la base de datos, el formato, y una área de trabajo donde se almacena el producto generado por el formato. Los comandos se ejecutan secuencialmente en el orden en que aparecen en el formato. Algunos comandos generan datos (p.ej. el contenido de un campo determinado), mientras que otros producen acciones (tales como saltar a una nueva línea, dejar una o más líneas en blanco, etc.). Los datos producidos son almacenados como líneas de texto en el área de trabajo, que posteriormente se transfieren al programa relevante para su proceso, por ejemplo, para que sean impresas.

Cuando un formato se usa para visualizar datos, las líneas producidas normalmente están limitadas a cierta longitud máxima (el ancho de línea). Por ejemplo, cuando se presentan datos en la pantalla, CDS/ISIS limita automáticamente el ancho de la línea a 80 caracteres. A menos que se fuerce un salto de línea con comandos explícitos, CDS/ISIS producirá una salida secuencial, tratando de llenar cada línea tanto como sea posible. Si un campo dado excede el ancho de la línea, CDS/ISIS lo partirá en tantas líneas como sea necesario. Cuando CDS/ISIS separa los datos en líneas, el corte siempre ocurre al nivel de palabra, o sea, una palabra nunca será dividida en dos líneas.

Todos los comandos de formateo pueden introducirse en mayúscula, minúscula o en combinaciones de ambas.

Selectores de Campos

Los selectores de campos son comandos usados para extraer un campo o subcampo específico de un registro. Un comando especial permite extraer el MFN del registro, aunque el MFN no es propiamente un campo (el MFN no tiene rótulo y no se define en la FDT).

Expresiones

El lenguaje de formateo permite evaluar y/o comparar valores a través del uso de expresiones. Las expresiones son estructuras que, cuando son ejecutadas, devuelven un valor. Este valor puede ser una cadena de caracteres (p.ej. el contenido de un campo o un literal dado), en cuyo caso la expresión se llama: expresión de cadena; un número, en cuyo caso la expresión se llama: numérica; o puede ser un valor de verdad (Verdadero o Falso), en cuyo caso la expresión se llama booleana. CDS/ISIS también proporciona un conjunto de funciones, que realizan un proceso específico y devuelven un valor, en base a los argumentos que se le proporcionan. Las funciones que devuelven un número se llaman funciones numéricas; aquellas que devuelven una cadena, se llaman funciones de cadena; y aquellas que devuelven un valor de verdad, se llaman funciones booleanas. Sólo las funciones de cadenas pueden ser usadas directamente en como comandos de formateo. Las expresiones numéricas pueden usarse en expresiones booleanas, o como argumentos de funciones.

Expresiones Numéricas

Las expresiones numéricas se forman con operandos que tienen un valor numérico, y operadores que especifican los cálculos a ser realizados. Los operandos que pueden usarse en una expresión numérica son los siguientes:

- **Constantes numéricas:** las constantes numéricas pueden representarse como enteros con un signo opcional, como números con decimales, o en notación científica exponencial, p. ej. 1.5E5 (que significa 1.5 veces 10 a la potencia 5, o sea 150000);
- **Funciones numéricas:** tales como val(v10) (estas se describen bajo "Funciones numéricas");
- **MFN:** el valor del MFN de un registro
- **Expresiones numéricas:** cuando se usan como operandos, las expresiones deben encerrarse entre paréntesis, por ejemplo (val(v20)-5).

Los operadores disponibles son:

- + suma (o + signo +);
- - resta (o - signo +);
- * multiplicación;
- / división.

Al igual que en el álgebra normal, en ausencia de paréntesis los operadores suma y resta se ejecutan en primer lugar, y las multiplicaciones y divisiones se realizan antes que las sumas y las restas. Una serie de dos o más operadores del mismo nivel, se ejecutan de izquierda a derecha. Se pueden usar paréntesis para alterar este orden de evaluación: las expresiones dentro de paréntesis se evalúan antes, y las expresiones entre paréntesis internos a otros, son evaluadas antes que las expresiones externas a los paréntesis.

Nótese que, como los selectores de campo (p.ej. v10 o v20^a) producen una cadena de texto, no pueden usarse como operandos en expresiones numéricas. Sin embargo, la función VAL, puede usarse para convertir el contenido de un campo o subcampo a un valor numérico.

Asimismo, una expresión numérica no puede visualizarse directamente, por lo que debe convertirse primero a una cadena de caracteres usando la función F.

A continuación se presentan ejemplos de expresiones numéricas (donde se supone que MFN=10, v1^a=10, v1^b=20, y v2=30):

Expresión	Valor
-----	-----
0.155e+3	155
1e-3	0.001
2*3+9	15
2*(3+9)	24
10-(4*(2-1))	6
15*0.001	0.015
mfn+100	110
val(v2)+val(v1^a)*7.5	105
(val(v1^a)-val(v1^b))/100	-0.1

Expresiones Alfanuméricas

Las expresiones alfanuméricas se forman con operandos que son cadenas de caracteres. Como CDS/ISIS no provee operadores alfanuméricos explícitos, una expresión de cadena siempre consiste de un operando solo, que puede ser uno de los siguientes:

- **literal incondicional:** tal como 'este es un texto';
- **selectores de campos:** que pueden incluir un comando de desplazamiento/longitud (p.ej. v26^c*2.2);
- **funciones de cadena:** tal como S(v24,v25,v26).

Expresiones Booleanas

Las expresiones booleanas se usan para determinar si un conjunto de una o más condiciones es verdadero o falso, y el resultado de su evaluación es un valor de verdad. Los operandos de una expresión booleana pueden ser cualquiera de los siguientes:

- **Expresiones relacionales:** que comparan dos valores y determinan si existe o no una cierta relación entre ellos (ver más adelante), tal como $mfn < 10$;
- **Funciones booleanas:** tal como $p(v24)$.

Las expresiones relacionales permiten determinar si una relación dada se verifica entre dos valores. La forma general de una expresión relacional es:

expresión-1 operador-de-relación expresión-2

donde:

- **expresión-1** es una expresión numérica o de cadena
- **operador-de-relación** es uno de los siguientes:
 - = Igual
 - \neq No igual (diferente de)
 - < Menor que
 - \leq Menor o igual que
 - > Mayor que
 - \geq Mayor o igual que
 - : Contiene (puede ser usado sólo en expresiones de cadena)
- **expresión-2** es una expresión del mismo tipo que expresión-1, o sea, expresión-1 y expresión-2 deben ser ambas expresiones numéricas o ambas expresiones de cadena.

Los operadores de relación = \neq < \leq > \geq tienen su significado convencional cuando se aplican a expresiones numéricas. Cuando se comparan expresiones de cadena, se aplican las siguientes reglas:

- Excepto por el operador ":" (contiene), las cadenas se comparan exactamente en la forma en que ocurren, o sea, las letras mayúsculas y minúsculas se comparan

de acuerdo con el código ASCII que les corresponde (p.ej. A será considerada menor que a);

- Dos expresiones de cadena no son consideradas iguales, a menos que tengan la misma longitud. Si dos expresiones generan cadenas de diferente longitud que son idénticas, carácter por carácter, hasta el total de la longitud de la más corta, entonces, la más corta será considerada menor que la más larga.

El operador “:” (contiene), busca una cadena de caracteres (definida por expresión-2) en otra cadena (definida por expresión-1). Si el segundo operando existe en cualquier parte del segundo operando, el resultado es Verdadero (TRUE). Este operador es insensible al hecho de que los caracteres se hallen en mayúsculas o minúsculas: por lo que las letras minúsculas se consideran iguales a su letra mayúscula correspondiente. Por ejemplo, el resultado de:

v10 : 'química'

será verdadero (True) si, y sólo si, el campo 10 contiene la cadena química. En caso contrario, el resultado será Falso (False). Nótese que el segundo operando puede ser cualquier cadena o carácter, y no necesita ser una palabra como tal. Por lo tanto, en este ejemplo, el resultado será Verdadero no sólo si el campo 10 contiene la palabra química, sino también si contuviera bioquímica, fotoquímicas, químicamente, etc.

Los operandos de una expresión booleana pueden combinarse con los operadores siguientes:

- **NOT (NO)** Este operador produce el valor Verdadero, si su operando es Falso; y el valor Falso, si su operando es Verdadero. El operador NOT sólo puede usarse como operador signo +, o sea, siempre se aplica a la expresión booleana que le sigue;
- **AND (Y)** Este operador produce el valor Verdadero si ambos operandos son Verdadero. Si cualquiera de los dos operandos es Falso, entonces el resultado será Falso;
- **OR (O)** Este operador realiza una operación O-inclusivo. El resultado es Verdadero si cualquiera de los dos operandos, o ambos son Verdadero. En caso contrario, es Falso.

Al evaluar expresiones booleanas, y en ausencia de paréntesis, CDS/ISIS ejecutará las operaciones NOT en primer lugar, después las operaciones AND, y finalmente las OR. Las series de dos o más operadores del mismo nivel, se ejecutan de izquierda a derecha. Se pueden usar paréntesis para alterar el orden de evaluación: las expresiones dentro de paréntesis se evalúan antes, y las expresiones entre paréntesis internos a otros, son evaluadas antes que las expresiones externas a los paréntesis.

Tabla de selección de campos (FST)

Una Tabla de Selección de Campos (FST) define los criterios para extraer uno o más elementos de un registro del archivo maestro. Dependiendo del contexto en el cual se utilice una FST, estos elementos pueden usarse para crear términos recuperables en el archivo invertido, correspondientes al registro del cual fueron extraídos, para la clasificación de registros en la secuencia deseada antes de producir un reporte impreso, o para reformatear registros durante una operación de importación o exportación.

Un elemento puede definirse generalmente como un fragmento del registro resultante de un proceso particular. Aunque en muchos casos los elementos serán datos elementales, es decir, campos o subcampos, en otros casos pueden ser palabras, frases, o cualquier otra parte de un dato que tenga un sentido particular para una aplicación específica.

Una FST consiste de una o más líneas, cada una de las cuales define tres parámetros:

- 1. un identificador de campo (columna rotulada ID);
- 2. una técnica de indizado (columna rotulado IT); y
- 3. un formato de extracción de datos codificado usando el lenguaje de formateo de CDS/ISIS.

Cuando se solicita a CDS/ISIS extraer elementos usando una FST, el programa leerá los registros relevantes del archivo maestro y llevará a cabo, para cada uno de ellos, y para cada línea de la FST, el siguiente proceso:

- 1. ejecuta el formato para extraer del registro los datos correspondientes

- 2. aplica la técnica de indizado especificada a los datos producidos por el formato; y
- 3. asigna a cada elemento producido el identificador de campo especificado.

El proceso descrito anteriormente es estrictamente mecánico y se desarrolla exactamente del modo que se describe. No hay transmisión de conocimiento entre un paso y el otro, solamente de datos, aunque todos los pasos cooperan para lograr el resultado deseado. Por ejemplo, el hecho de que un campo particular haya sido extraído durante el paso 1 no es conocido por el paso 2: el paso 1 usa al máximo la potencia del lenguaje de formateo para producir una cadena de caracteres y pasarlos al paso 2. Este paso opera sobre esta cadena de caracteres de acuerdo a la técnica de indizado especificada. Las técnicas de indizado están definidas como procesos sobre cadenas de caracteres, no sobre registros ni campos. Esto es debido al diseño generalizado de las FST, que permite usarla para diferentes propósitos, como definir el contenido del archivo invertido, o para especificar los requerimientos de clasificación de un listado impreso, procesos que pueden parecer, a primera vista, como cuestiones no relacionadas.

En su forma más general, puede imaginarse una FST como un dispositivo capaz de producir elementos de datos necesarios para realizar una tarea determinada.

Ingreso y edición de datos

El ingreso de los datos se hace a través de hojas de trabajo, las cuales han sido previamente definidas por medio del editor de hojas de trabajo. Como se mencionó previamente, CDS/ISIS utiliza dos tipos de hojas de trabajo: hojas de trabajo de ingreso, las cuales se utilizan para coleccionar los datos que se almacenarán en el archivo Maestro, y las hojas de trabajo del sistema CDS/ISIS, las cuales se utilizan para definir los parámetros necesarios para realizar una tarea particular, tales como generar reportes impresos. Ambos tipos de hojas de trabajo se llenan exactamente de la misma manera. La única diferencia es que las hojas de trabajo del sistema pueden tener solamente una página, mientras que las hojas de trabajo para ingreso de datos pueden extenderse a varias páginas.

Cada campo de una hoja de trabajo se ingresa o modifica a través del editor de campos. Sin embargo, CDS/ISIS provee adicionalmente un conjunto completo de

facilidades para edición de pantallas, tales como mover un dato de un campo a otro campo en una página, o pasar de una página a otra.

Recuperacion de Datos

El lenguaje de recuperación de CDS/ISIS se basa en el álgebra booleana, que provee una forma conveniente de expresar operaciones lógicas entre conjuntos. Cada término de búsqueda asociado con un registro dado puede, de hecho, considerarse que representa el conjunto de todos aquellos registros asociados con ese término. Por lo tanto, al establecer operaciones lógicas entre términos de búsqueda, se puede definir con precisión el conjunto de los registros a ser recuperados en respuesta a la necesidad planteada.

La forma más eficiente de realizar una búsqueda con CDS/ISIS, es a través del archivo invertido (la razón fundamental para mantener un archivo invertido es precisamente la de permitir una recuperación rápida). Sin embargo, sólo se puede hacer esto si los requerimientos de búsqueda están dentro del ámbito del diccionario de términos de búsqueda definido para una base de datos determinada, o sea, que han sido previstos cuando se definió el contenido del archivo invertido. No importa lo bien pensado que haya sido el diseño de una base de datos, habrá casos en que no sea posible formular una búsqueda en base al archivo invertido. Para estos casos, CDS/ISIS provee un método alternativo de búsqueda, denominado búsqueda en texto libre, que permite satisfacer cualquier requerimiento de búsqueda independientemente del contenido del archivo invertido. Debido a que este segundo método es mucho menos eficiente, normalmente es utilizado en forma asociada a una búsqueda sobre el archivo invertido, para refinar la estrategia de búsqueda, pero puede ser usado en forma independiente si esto es necesario. Si se realizan largas búsquedas en texto libre con frecuencia, puede ser conveniente considerar la posibilidad de redefinir el archivo invertido. Los párrafos siguientes describen el método utilizado para realizar búsquedas a través del archivo invertido.

Tipos de términos de búsqueda

Al formular la expresión de búsqueda, pueden usarse tres tipos de términos: términos exactos, términos truncados a la derecha, y términos ANY. Estos se explican a continuación.

- **Términos exactos:** Un término exacto es cualquier elemento de búsqueda definido para una base de datos dada, tal como un descriptor de materia, palabra clave, frase clave, palabras del título, nombres de autor, etc. El usuario debe estar familiarizado con los términos de búsqueda disponibles en cada base de datos en la que pretenda realizar búsquedas. Cuando se usa un término de búsqueda exacto, este debe ser especificado en la forma idéntica en que es conocido por CDS/ISIS.
- **Términos truncados a la derecha:** En lugar de especificar un término exacto, se puede proporcionar solamente la raíz. Esta técnica, conocida también como búsqueda de raíz o truncamiento a la derecha, permite realizar la búsqueda sobre la secuencia inicial de caracteres. CDS/ISIS realizará automáticamente una operación de O (or) lógico entre todos los términos de búsqueda que tengan la raíz indicada.
- **Términos ANY:** Un término ANY es un término colectivo que representa a un conjunto predefinido de términos de búsqueda. Cuando se incluye un término ANY en la formulación de una búsqueda, CDS/ISIS juntará con una operación O (or), todos los términos de búsqueda de los asociados con el término ANY especificado. Un término ANY consiste de la palabra ANY seguida por un identificador único, generalmente mnemotécnico, asignado al conjunto de términos asociados. Por ejemplo, el término ANY BENELUX puede usarse para recuperar los registros indizados con el nombre individual de cualquier país del grupo Benelux (Bélgica, Holanda y Luxemburgo). Antes de que pueda usarse un término ANY en una búsqueda, su significado debe ser definido especificando el conjunto de términos asociados. Nótese que no es necesario que se defina esto para todas las bases de datos. Por lo tanto, antes de intentar usar un término ANY debe asegurarse que efectivamente existe esta facilidad para la base de datos considerada, y que el término ANY que se desea usar, está efectivamente definido.

Operadores de búsqueda

Se pueden combinar dos o más términos de búsqueda en una expresión, utilizando operadores de búsqueda que indiquen la relación que se desea imponer entre los términos. Los tres operadores básicos, O, Y, y NO (or, and y not)

Operadores a nivel de campo y de proximidad

Estos operadores son tipos más restrictivos del operador lógico AND, y son particularmente útiles en la búsqueda por medio de lenguaje natural.

Los operadores a nivel de campo y de proximidad son los siguientes:

- **(G)** mismo campo (todas las ocurrencias de un campo repetible se consideran como una sola entidad), por ejemplo:

agua (G) suelo

recuperará todos los registros que incluyan tanto agua, como suelo, siempre y cuando ambos términos se hallen en el mismo campo;

- **(F)** mismo campo u ocurrencia individual de un campo repetible, por ejemplo:

agua (F) suelo

recuperará todos los registros que incluyan tanto agua como suelo, siempre y cuando ambos términos se hallen en el mismo campo o en la misma ocurrencia de un campo repetible ((G) y (F) son equivalentes cuando se aplican a campos no repetibles)

- **.** igual que (F), pero con la restricción adicional de que los dos términos no se hallen a **más de** n palabras de distancia, donde n es el número de puntos más uno. Por ejemplo:
 - **A . B** adyacentes
 - **A . . B** máximo una palabra entre A y B
 - **A . . . B** máximo dos palabras entre A y B
- **\$** igual que (F), pero con la restricción adicional de que los dos términos se hallen **exactamente** a n palabras de distancia, donde n es el número de signos \$ más uno. Por ejemplo:
 - **A \$ B** adyacentes
 - **A \$ \$ B** exactamente una palabra entre A y B
 - **A \$ \$ \$ B** exactamente dos palabras entre A y B

Nótese que los operadores **.** y **\$** deben estar precedidos y seguidos por un espacio.

Sintaxis de las expresiones de búsquedas

Al combinar dos o más términos de búsqueda con los operadores descritos, se pueden formar expresiones de búsqueda complejas.

Al igual que en el álgebra normal, se pueden usar paréntesis para alterar el orden de evaluación. En la evaluación de expresiones la prioridad de los operadores es la siguiente:

- (más alta) \$ y .
- | (F)
- | (G)
- v * y ^
- (más baja) +

Si dos o más operadores de la misma prioridad aparecen en el mismo nivel de paréntesis en la misma expresión, serán ejecutados de izquierda a derecha. Por lo tanto, para evaluar la expresión del siguiente ejemplo:

$$A + B * C$$

CDS/ISIS evaluará primero $B * C$ y después evaluará el O lógico entre A y $(B * C)$. Mientras que para evaluar:

$$(A + B) * C$$

primero evaluará $A + B$ y después el Y lógico entre $(A + B)$ y C . Se pueden anidar paréntesis en caso necesario, como lo muestra el siguiente ejemplo:

$$((A + B) * C + (D + E) + F) ^ G$$

Al formular una expresión de búsqueda deben observarse algunas reglas sintácticas simples:

- 1. No puede haber dos operadores lógicos adyacentes, excepto por " . " y " \$ " repetidos (que sin embargo no pueden mezclarse);
- 2. Los paréntesis deben equilibrarse, o sea, el número de paréntesis que abren debe ser igual al número de paréntesis que cierran, y cada paréntesis que abre debe corresponder a uno que cierra.

Búsqueda en texto libre. (Búsqueda secuencial)

Esta técnica permite especificar condiciones de búsqueda relacionadas con campos que no han sido invertidos, y/o especificar condiciones que no sería posible indicar de otro modo a través de las expresiones de búsqueda descritas más arriba, tales

como la comparación de campos, o la comparación de los valores numéricos de los campos.

Para poder distinguirla de una expresión de búsqueda normal, debe precederse con un signo interrogativo, del modo siguiente:

- **? expresión booleana** o
- **? #n expresión booleana**

donde:

- **?** identifica a esta como una búsqueda en texto libre
- **#n** restringe optativamente la búsqueda en texto libre al resultado de una búsqueda previa (n es el número del conjunto de la búsqueda previa: esta puede haber sido una búsqueda a través del archivo invertido, o una búsqueda en texto libre, o una combinación de ambos tipos); si se omite, la búsqueda en texto libre se realizará sobre toda la base de datos;
- **expresión booleana** es una expresión booleana de CDS/ISIS.

En respuesta a una búsqueda en texto libre, CDS/ISIS evaluará la expresión booleana para cada registro del archivo maestro, y construirá una lista HIT que identifica los registros que satisfacen la condición especificada (o sea, todos los registros para los cuales la expresión booleana produce el valor Verdadero [true]). Por ejemplo:

`? v24 : 'Unesco' and val(v26^c) >= 1986`

recuperaría todos los registros que contienen Unesco en el campo 24, con una fecha de publicación (v26^c) mayor o igual a 1986. Lo siguiente:

`? #2 (p(v24) or p(v29)) and v26^b : 'unesco'`

recuperaría, entre los registros recuperados por la expresión #2, sólo aquellos que contienen unesco en el subcampo b del campo 26, siempre y cuando estos registros contengan ya sea el campo 24 o el 29 o ambos.

La expresión booleana puede ser tan compleja como se desee, siempre y cuando no exceda el límite de 250 caracteres (poco más de tres líneas en la pantalla). Al igual que las búsquedas con el archivo invertido, cada búsqueda en texto libre recibe un número de conjunto, que puede posteriormente utilizarse en otras expresiones de búsqueda.

Nótese que aun cuando las expresiones que usan el archivo invertido y las de texto libre, no pueden combinarse en una sola expresión, es posible a través de las referencias retrospectivas combinar cualquier número de ellas en expresiones por separado, tal como se muestra en el ejemplo siguiente:

- conj 1 AGUA * SUELO
- conj 2 ? #1 val(v26^c) > 1985
- conj 3 ANY LATINOAMERICA
- conj 4 #2 * #3

Debido a que CDS/ISIS debe leer cada registro del archivo maestro para poder verificar si se cumplen las condiciones de la búsqueda se satisfacen, el tiempo de proceso de búsquedas en texto libre puede ser muy grande, especialmente si se aplica a una base de datos grande.

ISO 2709

Historia

En 1960 se desarrolló un formato de intercambio bibliográfico de la información, desarrollado bajo la dirección de la Biblioteca del Congreso de los Estados Unidos, para codificar la información impresa en las fichas catalográficas. Primero fue creado como el estándar ANSI Z39.2, uno de los primeros estándares en tecnologías de la información, también llamado *Formato de intercambio de información* (en inglés *Information Interchange Format*). La última edición de ese estándar es Z39.2-1994 (ISSN: 1041-5653). La norma ISO reemplaza el estándar Z39.2. El estándar ISO 2709 fue creado en 1981, modificado en 1996, y la última modificación fue hecha en el año 2008 (ISO2709:2008).

Estuctura básica

Un registro ISO consiste de tres segmentos lógicos:

- **Rótulo del registro** Longitud fija para cada registro
- **Directorio** Consiste en un número variable de elementos de longitud fija, cada uno de los cuales proporciona la identificación, la longitud y la ubicación de cada campo variable en el registro. Termina con un separador de campo.
- **Campos de datos** Contiene los datos propiamente dichos en la forma de información alfanumérica de longitud variable. Cada campo termina con un separador de campos.

Tanto el rótulo del registro como el directorio, son segmentos de control que se usan para el proceso de los datos contenidos en el tercer segmento. Un registro puede ser de cualquier longitud deseada. Termina con un separador de registro. Como cada registro del archivo es de longitud variable, no se requiere espacio excedente para guardar los datos en el archivo.

Segmento del Rotulo del Registro

El rótulo del registro está formada por elementos de longitud fija que proporcionan información de control al programa que procesa el registro.

Desplazamiento	Longitud	Descripción
-----	-----	-----
0	5	Longitud del registro
5	1	Estado del registro
6	4	Códigos de implementación
10	1	Longitud del indicador
11	1	Longitud del identificador de subcampo
12	5	Dirección base de los datos
17	3	Para sistemas del usuario
20	1	Longitud del campo "longitud del dato"
21	1	Longitud de la posición del carácter inicial
22	2	Para uso futuro

Longitud del registro: Este es un número decimal que indica la longitud total del registro (incluye el rótulo del registro, el directorio y el separador de registros).

Estado del registro: Siempre '0' para archivos de salida, ignorado en archivos de entrada.

Códigos de Implementación: Estos códigos se definen para cada implementación específica de la norma ISO 2709. ISISXCH los establece como '0000' cuando genera archivos de salida.

Longitud del Indicador: Este campo define la longitud de los indicadores que pueden encontrarse al inicio de cada campo. Los indicadores pueden utilizarse en algunos casos para proveer información descriptiva acerca del campo.

Longitud del Identificador de Subcampo: Este campo define la longitud de las marcas que delimitan e identifican los subcampos existentes en algunos campos de datos (normalmente 2 caracteres).

Dirección base de los datos: Es un número decimal que indica el número de caracteres de control (longitud del rótulo del registro más el directorio y el separador de campo) que preceden a los campos de datos variables. Este número, cuando se suma a la dirección donde inicia el registro, proporciona la dirección del primer carácter de los campos variables.

Para sistemas de los usuarios: Estas posiciones pueden contener datos para uso del sistema en algunas aplicaciones de la norma. Se les asigna '000' en archivos de salida.

Longitud del campo “Longitud del Dato”: Este es un número decimal que indica la longitud en caracteres del elemento 'longitud del campo' en cada entrada del directorio.

Longitud de la “Posición del Carácter Inicial”: Este es un número decimal que indica la longitud en caracteres del elemento 'posición del carácter inicial' en cada entrada del directorio.

Para uso futuro: Contiene ceros.

Segmento del Directorio

Cada entrada del directorio es un campo de longitud fija que contiene tres elementos:

- Un rótulo del campo, o código que identifica el campo
- La longitud del campo variable
- La ubicación del campo, relativa al comienzo del área de campos variables.

La longitud de una entrada del directorio está definida en los campos descriptivos del mapa del directorio, en el segmento del rótulo del registro.

Rotulo del Campo

Este elemento es un símbolo que identifica el campo. Los valores de los rótulos de los campos son números decimales. La interpretación específica del significado de los rótulos de campo está determinada por el programa de proceso.

Longitud del Campo

Es un valor decimal igual a la longitud (en caracteres) del campo variable al cual apunta esta entrada del directorio (incluyendo el separador de campo al final), esto es, la longitud del campo en la base de datos más uno).

Posición del Campo

Es un valor decimal que indica la posición del primer carácter del campo, relativa al comienzo del segmento de campos variables. Este número sumado a la dirección real del inicio de los campos variables, dará la dirección real del comienzo de este campo.

Segmento de los campos de datos variables

El registro contiene campos de datos de longitud variable después del rótulo del registro y del directorio. Para localizar el punto inicial de un campo es necesario explorar el directorio hasta encontrar el rótulo del campo deseado. Hay una entrada en el directorio para cada uno de los campos presentes en el registro. Los campos variables

se encuentran en el registro en la misma secuencia que las entradas correspondientes en el directorio. Después del directorio y de cada campo variable hay un carácter separador de campos. El registro se termina con un carácter separador de registros.

Formato De Archivos ISO Producidos y Aceptados Por CDS/ISIS

Los archivos ISO producidos por CDS/ISIS son archivos de texto estándar descritos en el formato anterior. Debido a que no es práctico manejar archivos de texto con líneas relativamente largas, especialmente cuando tienen que ser inspeccionados con editores y/o transmitidos por líneas de telecomunicación, CDS/ISIS dividirá cada registro en bloques de 80 caracteres, cada uno de ellos seguido por el carácter estándar de fin de línea ^M^J (Carriage Return/Line Feed - Retorno del carro/salto de línea). Todos los bloques excepto el último contendrán exactamente 80 caracteres. De esta forma un registro ISO de 835 caracteres constará de 11 líneas, las primeras 10 de 80 caracteres y la última de 35.

Por ejemplo, supongamos el siguiente registro de una base de datos CDS, el cual contiene los siguientes campos:

- 44 Methodology of plant eco-physiology: proceedings of the Montpellier Symposium
- 50 Incl. bibl.
- 69 Paper on: <plant physiology><plant transpiration><measurement and instruments>
- 24 Techniques for the measurement of transpiration of individual plants
- 26 ^aParis^bUnesco^c 1965
- 30 ^ap. 211-224^billus.
- 70 Magalhaes, A.C.
- 70 Franco, C.M.

Si este registro fue exportado, el registro ISO correspondiente generado por CDS/ISIS consistiría de las 6 líneas que se muestran:

```
004320000000012100045000440078000000500012000780690079000900240069001
69026002200238030002100260070001600281070001300297#Methodology of plant
eco-physiology: proceedings of the Montpellier Symposium#Incl. bibl.#Paper on:
```

<plant physiology><plant transpiration><measurement and instruments>#Techniques for the measurement of transpiration of individual plants#^aParis^bUnesco^c1965#^ap. 211-224^billus .#Magalhaes, A.C.#Franco, C.M.##

MARC

Que es MARC?

Los formatos de intercambio son sistemas de códigos y contenidos que permiten la codificación de registros legibles por computadoras. Dentro de estos formatos se ubica el formato MARC.

El formato MARC (*Machine Readable Cataloguing Record*) es un conjunto de normas que permite almacenar información en registros de cualquier tipo, para posteriormente, poder tratarla, localizarla, intercambiarla o ponerla a disposición del usuario.

El formato MARC fue desarrollado para ayudar a las bibliotecas en el uso, desarrollo y mantenimiento de sus bases de datos, y precisamente el desarrollo de este formato ha hecho realidad la catalogación compartida y la automatización de bibliotecas.

Evolución de MARC hacia MARC 21

MARC fue desarrollado en la década de 1960, a partir de las fichas catalográficas que *Library of Congress* vendía a otras bibliotecas. Por esos años, cuando se difundió el uso de las computadoras, no tenía sentido seguir duplicando el trabajo de varios catalogadores para redactar la misma ficha catalográfica si se podía confeccionar un solo registro y compartirlo.

El proyecto original iniciado en 1964 se denominaba *MARC Pilot Project*, estaba liderado por Henriette Avram y su objetivo era comprobar la viabilidad de producir registros catalográficos legibles por computadora. La idea era crear un estándar de comunicación entre distintas bibliotecas, estructurar esos datos de forma tal que se pudieran realizar adaptaciones y modificaciones en cada una de ellas y que el formato pudiera ser utilizado en diferentes equipos informáticos.

La primera etapa del proyecto se denominó MARC I y el formato se restringía exclusivamente a libros. La segunda etapa se conoció como MARC II y su objetivo era proveer una estructura que permitiera la descripción de todos los tipos de materiales.

Entre las tareas que hubo que desarrollar para lograr el intercambio de registros redactados en alfabetos diferentes, fue el establecimiento de un conjunto de caracteres. La estructura de los registros USMARC (así se conoció al formato desarrollado por LC) es el resultado de la implementación de la *American National Standard for Information Interchange on Magnetic Tape (ANSI Z39.2-1979)* y del *Documentation-Format for Bibliographic Information Interchange on Magnetic Tape (ISO 2709-1981)*.

Casi al mismo tiempo en que desarrollaba el proyecto en los Estados Unidos, la *British National Bibliography* comenzó a trabajar en la producción de su bibliografía nacional por medios automáticos, de modo que ambas instituciones estuvieron de acuerdo en considerar el intercambio de registros. El formato británico fue conocido como UKMARC

Luego se fueron sumando otros países como Francia, Alemania, Italia, Canadá y Australia, donde se desarrollaron formatos nacionales compatibles con MARC.

En 1980 la IFLA (International Federation of Library Associations and Institutions) compiló la *International Guide to MARC Databases and Services* donde se incluían 21 bibliotecas nacionales que producían sus registros en 14 MARC nacionales diferentes. Ya para mediados de los '90 las bibliotecas nacionales de 60 países del mundo habían redactado sus registros en ese formato. Aproximadamente la mitad de los formatos nacionales se basaban en US MARC, un cuarto de ellos se basaban en UNIMARC y el resto en otros formatos nacionales como UK MARC o combinaciones de otros formatos.

Integración de formatos

La proliferación de diferentes tipos de materiales y la necesidad de crear registros para cada uno de ellos ha sido una complicación, acentuada por el criterio de utilizar formatos separados adoptado por Library of Congress. Desde 1984 se ha trabajado para integrarlos y lograr que la misma información pueda ser tratada de la misma forma para todos los tipos de materiales, es decir que la designación de contenidos sea consistente en todos los casos.

En el caso de UNIMARC la integración no ha sido un tema significativo pues desde el principio fue concebido como un formato simple donde se aplicaba la misma información para los diferentes tipos de materiales, al estilo de las ISBD(G). Por ejemplo, los campos de datos codificados funcionan independientemente del tipo de registro de que se trate. Sin embargo, hay inconsistencias en la forma en que han sido definidos estos campos debido a su desarrollo independiente para cada material a lo largo del tiempo.

En diferentes países se ha trabajado para desarrollar formatos que se adaptasen a las necesidades propias. Al existir distintos códigos de catalogación, diferentes lenguas y al variar los fines para los cuales han sido desarrollados (para la confección de bibliografías nacionales o la impresión de sus fichas catalográficas) los formatos difieren entre sí. En todos los casos han tomado como modelo a los ya desarrollados por LC y BNB y a UNIMARC.

Formatos basados en USMARC	Formatos basados en UKMARC	Formatos basados en UNIMARC
CANMARC (Canadá)	AUSMARC* (Australia)	SAMARC (South Africa)
INTERMARC (Francia)	THAIMARC (Tailandia)	Taiwan MARC
IBERMARC (España)	ANNAMARC (Italia)	JAPAN/MARC (Japón)
INDOMARC	SINGMARC	YU-UNIMARC**

(Indonesia)	(Singapur)	(Croacia)
-------------	------------	-----------

* sólo la primera edición

** denominado así en sus inicios

Entre 1994 y 1997 Library of Congress y la Biblioteca Nacional de Canadá trabajaron en conjunto, a través de sus respectivos comités MARC, para integrar los formatos utilizados en ambos países.

MARC 21 surgió como resultado de la armonización de CANMARC y USMARC en 1999.

UNIMARC

En 1972 la IFLA creó el *Working Group on Content Designators*, quien tuvo a su cargo el análisis de los designadores de contenido para los distintos formatos nacionales que habían sido desarrollados, con el objeto de uniformarlos y que pudieran servir como un estándar para el intercambio internacional.

En sus inicios este formato se denominó **SuperMARC** y más tarde **UNIMARC**, y fue concebido más que como un estándar impuesto a las agencias bibliográficas nacionales, como una herramienta para transferir los datos entre formatos diferentes. La idea era que funcionara como un *vocabulario común* que posibilitara a las distintas agencias realizar sólo una conversión hacia y desde UNIMARC.

Después de la publicación de su segunda edición en 1980, la Biblioteca Nacional de Alemania condujo en el marco del Programa de la IFLA, una evaluación del formato en la que participaron además Library of Congress, British Library, la Royal Library de Estocolmo, la Biblioteca Nacional de Canadá, entre otras. El resultado de la evaluación demostró una serie de inconsistencias en la interpretación de las especificaciones de UNIMARC, deficiencias de precisión en las pruebas de conversión entre formatos y una variedad de técnicas utilizadas, lo que motivó una revisión del formato y la publicación del Manual de UNIMARC en 1983.

El Registro MARC

El registro MARC puede definirse como una agrupación lógica de elementos, dispuestos en un orden previsto y que constituyen una sola unidad de información que podría ordenarse en cualquier lista, catálogo, bibliografía, etc.

Existen tres normas ISO que regulan la estructura de estos registros:

- ISO 2709: establece las tres partes principales del registro: cabecera, directorio y área de datos
- ISO 1001: especifica la estructura física del archivo para el intercambio de información
- ISO 6630: regula el juego de caracteres en el registro bibliográfico

MARC para diferentes tipos de materiales

El formato desarrollado en 1966 por el MARC Pilot Project fue diseñado para cubrir los requerimientos de los registros que describieran libros impresos. En los siguientes años LC fue desarrollando diferentes versiones para otros tipos de materiales:

- 1970 - publicaciones seriadas, mapas
- 1971 - películas
- 1973 - manuscritos
- 1976 - música, registros sonoros

UNIMARC también se encaminó en este sentido. La primera edición del formato publicada en 1977 refleja la estructura de la *General International Standard Bibliographic Description*. Como las diferentes versiones de las ISBDs fueron siendo aprobadas una por una, UNIMARC optó por adoptar las especificaciones para materiales especiales como un apéndice del formato que incluía campos y subcampos provisionarios, para recién incluirlos de manera definitiva en su edición de 1987. En la edición de 1994 sólo se mantienen provisionales los campos que describen archivos de computadora.

Normas MARC 21

Formato MARC 21 para Autoridades

El Formato **MARC 21 para Registros de Autoridades** ha sido diseñado para servir como portador de la información relativa a las formas autorizadas tanto de los nombres como de las materias, usados como puntos de acceso en los registros MARC; las formas de esos nombres, materias y subdivisiones que se usarán como referencias hacia la forma autorizada; y las interrelaciones existentes entre dichas formas.

Los nombres puede ser utilizados en registros bibliográficos como: asientos principales, asientos secundarios, o como asientos secundarios de materia o de serie. El término

nombre puede referirse a:

- Nombres de personas (X00)
- Nombres de entidades corporativas (X10)
- Nombres de reuniones (X11)
- Nombres de jurisdicciones (X51)
- Títulos uniformes (X30)
- Combinaciones de Nombre/Título

Las materias pueden utilizarse únicamente como asientos secundarios temáticos.

El término **materia** puede referirse a:

- Términos cronológicos (X48)
- Términos temáticos (X50)
- Nombres geográficos (X51)
- Términos de género/forma (X55)
- Nombres con subdivisiones temáticas
- Términos temáticos y de género/forma con subdivisiones temáticas

El *Formato MARC 21 para Registros de Autoridades* también provee información referente a los términos utilizados como etiquetas de nodo, las cuales indican la base lógica de subdivisión de cada categoría que se utiliza dentro de la sección sistemática de un tesoro. Las etiquetas de nodo no se deben asignar a documentos como términos indizables.

Formato MARC 21 para Registros Bibliográficos

El *Formato MARC 21 para Registros Bibliográficos* ha sido diseñado para servir como portador de la información bibliográfica relativa a: materiales textuales impresos y manuscritos, archivos de computadora, mapas, música, recursos continuos, materiales visuales y materiales mixtos. La información bibliográfica comúnmente incluye: títulos, nombres, tópicos, notas, datos de publicación e información sobre la descripción física de un ítem. El formato bibliográfico contiene elementos de información para los siguientes tipos de materiales:

- **Libros (BK)** - Se utiliza para materiales textuales impresos, electrónicos, manuscritos y en microformatos, cuya naturaleza es monográfica.
- **Recursos continuos (CR)** - Se utiliza para materiales textuales impresos, electrónicos y en microformatos, que se emiten en partes con un patrón de publicación recurrente, p. ej., publicaciones periódicas, diarios, anuarios. (**Nota:** Anteriormente a 2002 se hacía referencia a los recursos continuos como publicaciones seriadas.)
- **Archivos de computadora (CF)**- Se utiliza para programas lógicos de computadora, datos numéricos, multimedia, sistemas o servicios en línea. Otras clases de recursos electrónicos se codifican de acuerdo a su aspecto más significativo. Los materiales pueden ser de naturaleza monográfica o seriada.
- **Mapas (MP)** - Se utiliza para todo tipo de materiales cartográficos impresos, manuscritos, electrónicos y en microformatos, incluyendo atlas, mapas planos y globos. Los materiales pueden ser de naturaleza monográfica o seriada.
- **Música (MU)** - Se utiliza para música impresa, electrónica y en microformato; así como para grabaciones sonoras musicales y grabaciones sonoras no musicales. Los materiales pueden ser de naturaleza monográfica o seriada.
- **Materiales visuales (VM)** - Se utiliza para medios proyectables, medios no proyectables, gráficos bidimensionales, artefactos tridimensionales u objetos naturales; y, conjuntos. Los materiales pueden ser de naturaleza monográfica o seriada.
- **Materiales mixtos (MX)** - Se utiliza principalmente para colecciones de archivo y manuscritos con una mezcla de formas de materiales. Los materiales pueden ser de

naturaleza monográfica o seriada. (NOTA: Anteriormente a 1994, se hacía referencia a los Materiales mixtos (MX) como Materiales de archivo y manuscritos (AM)).

Tipos de Registros Bibliográficos

Los registros bibliográficos MARC se distinguen de todos los otros tipos de registros MARC mediante códigos específicos en la Cabecera/06 (Tipo de registro) que identifican a los siguientes tipos de registros bibliográficos:

Material impreso textual	Grabación sonora no musical
Material manuscrito textual	Grabación sonora musical
Archivos	Medio proyectable
Material cartográfico	Gráfico bidimensional no proyectable
Material cartográfico manuscrito	Artefacto tridimensional y objetos naturales
Música impresa con notación	Conjunto
Música manuscrita	Materiales mixtos

Los **Microformatos**, ya sea originales o reproducciones, no se identifican como un tipo especial de registro. El aspecto de la microforma es secundario al tipo de material al cual pertenece el ítem original (p. ej., libro). Lo mismo es aplicable para los **Archivos**, ya que el aspecto de archivo es secundario; sin embargo, ciertas categorías de recursos electrónicos se codifican archivos

Descripción de las partes del registro bibliográfico

Un registro MARC bibliográfico consiste en tres componentes principales:

- la **Cabecera**,
- el **Directorio**
- y los **campos variables**.

La siguiente información resume la estructura de un registro individual MARC.

Cabecera → elementos de información que esencialmente proveen datos para el procesamiento del registro. Los elementos de información contienen números o valores codificados; y se identifican por la posición relativa del carácter. La Cabecera posee una longitud fija de 24 caracteres y constituye el primer campo de un registro MARC.

Directorio → serie de entradas que corresponden a la etiqueta, la longitud y el punto de inicio de cada campo variable dentro de un registro. Cada entrada posee una longitud de 12 caracteres de posición. Las entradas del Directorio correspondientes a los campos variables de control aparecen primero, en una secuencia que sigue el orden numérico de las etiquetas de campo; le siguen las entradas de los campos de datos variables en orden ascendente de acuerdo al primer carácter de la etiqueta. La secuencia almacenada de los campos de datos variables del registro no corresponde necesariamente al orden de las entradas correspondientes del Directorio, Las etiquetas duplicadas se distinguen únicamente por la localización de los campos respectivos dentro del registro. El Directorio finaliza con un carácter terminador de campo (ASCII 1E hex).

Campos variables → dentro de un registro bibliográfico MARC, los datos se organizan como campos variables, y cada uno de estos se identifica mediante una etiqueta numérica de tres caracteres, que se almacena en la entrada correspondiente en el Directorio. Cada campo termina con un carácter terminador de campo. El último campo de un registro finaliza tanto con un terminador de campo como con un terminador de registro (ASCII 1D hex). Existen dos tipos de campos variables.

Campos variables de control → los campos 00X. Estos campos se identifican mediante una etiqueta de campo dentro del Directorio, pero no contienen posiciones de indicadores ni códigos de subcampo. Los campos variables de control son estructuralmente diferentes de los campos variables de datos, ya que pueden contener ya sea un elemento de información individual o una serie de elementos de longitud fija que se identifican por su posición relativa de carácter.

Campos variable de datos → los campos variables restantes que se definen dentro del formato. Además de estar identificados en el Directorio mediante una etiqueta de campo, los campos variables de datos contienen dos posiciones de indicadores almacenados al inicio de cada campo; y un **código de subcampo** de dos caracteres precede a cada elemento de información dentro del campo.

Los campos variables de datos se agrupan en bloques de acuerdo con el primer carácter de la etiqueta, la cual identifica, con algunas excepciones, la función de los datos dentro del registro. El tipo de información del campo se identifica por el resto de la etiqueta.

- 0XX Información de control, números de identificación y clasificación, etc.
- 1XX Asientos principales
- 2XX Títulos y párrafo del título (título, edición, pie de imprenta)
- 3XX Descripción física, etc.
- 4XX Menciones de serie
- 5XX Notas
- 6XX Campos de acceso temático
- 7XX Asientos secundarios diferentes a los de materias y series; campos ligados
- 8XX Asientos secundarios de series, existencias, etc.
- 9XX Reservados para implementación local

Dentro de los bloques 1XX, 4XX, 6XX, 7XX y 8XX, se preservan ciertos paralelos de la designación de contenidos. Los siguientes significados se dan, con algunas excepciones, a partir de los dos últimos caracteres de las etiquetas de los campos:

- X00 Nombres personales
- X40 Títulos bibliográficos
- X10 Nombres corporativos
- X50 Términos temáticos
- X11 Nombres de reuniones
- X51 Nombres geográficos
- X30 Títulos uniformes

Dentro de los campos variables de datos se utilizan los dos siguientes tipos de designación de contenido:

Posiciones de indicadores: las dos primeras posiciones de carácter de un campo variable de datos; contienen valores que interpretan o complementan los datos que se encuentran en el campo. Los indicadores se interpretan de manera independiente, es decir, su significado no está adscrito a los indicadores tomados en conjunto. Los valores de los indicadores pueden corresponder a caracteres alfabéticos en minúscula o a un carácter numérico. Se utiliza un espacio en blanco (ASCII SPACE), representado en

este documento como un “#”, cuando una posición de carácter no ha sido definida. En una posición definida de indicador, un espacio en blanco podría tener un significado asignado, o podría significar que *no se provee información*.

Códigos de subcampo: un código de subcampo consiste de un delimitador (ASCII 1F hex), representado en este documento mediante un “\$”; seguido por un elemento identificador. Los elementos identificadores pueden corresponder a un carácter alfabético en minúscula o a un carácter numérico. Los códigos de subcampo se han definido en forma independiente para cada campo; sin embargo, se preservaron algunos significados paralelos siempre que fue posible (p. ej., en los campos 100, 400 y 600 correspondientes a campos de Nombres Personales). Los códigos de subcampos se han definido con propósitos de identificación, no de arreglo interno. El orden de los subcampos se especifica generalmente mediante pautas para el contenido de los datos, tales como las reglas de catalogación.

Formato MARC 21 para Datos de Existencia

El fin de este tipo de formato es proporcionar información sobre cada ítem (unidad circulante), identificar cada uno de los documentos que constituyen el fondo, con datos como la signatura, colección, número de copia, etc. Identifica tres tipos de ítems: simple (por ejemplo, monografía en un volumen), multiparte (por ejemplo, obra en varios volúmenes), seriado (por ejemplo, publicación periódica o seriada).

Formato MARC 21 para Datos de Clasificación

Este formato permite codificar los datos relacionados con los números de clasificación y las notas asociadas con ellos. Permite, por tanto, relacionar las notaciones con sus encabezamientos de materia o descriptores correspondientes.

Formato MARC 21 para Información de la Comunidad

Es un formato diseñado para la descripción de recursos no bibliográficos. Identifica 5 tipos de información no bibliográfica:

- individual: datos sobre un individuo particular
- organización: datos sobre un grupo o entidad

- programas o servicios: servicios o actividades que no constituyen entidad
- eventos: entidades temporales
- otros: por ejemplo, tipos de transportes, entretenimientos...

Z3950

Introducción

Su nombre oficial "Information Retrieval (Z39.50); Application Service Definition and Protocol Specification. ANSI/NISO Z39.50-1995", aunque es más conocido por Z39.50. Su nombre deriva de haber sido desarrollado por el comité número 39 de la ANSI y por ser el estándar 50 de la NISO.

El Z39.50 es un protocolo para la recuperación de información basado en la estructura cliente/servidor que facilita la interconexión de sistemas informáticos.

El objetivo principal del CLIENTE Z39.50 consiste en permitir al usuario realizar búsquedas en bases de datos que cuenten con un servidor Z39.50, sin tener que conocer para ello las sintaxis de búsqueda que utilicen dichos sistemas.

Formalmente, facilita la interconexión entre los usuarios y las bases de datos donde se encuentra la información que necesitan a partir de una interfaz común y da fácil manejo, independientemente del lugar en que se encuentren las bases de datos así como la estructura y la forma de acceso de estas.

Historia

En los últimos años, especialmente a partir de 1995, es bastante habitual encontrar menciones y referencias al protocolo z39.50 en publicaciones del sector bibliotecario, y en catálogos de software, ya que los proveedores de programas de gestión bibliotecaria van paulatinamente incluyendo productos, basados en el protocolo, integrados en sus sistemas. Uno de los beneficios básicos que ha conseguido este protocolo, es en el ámbito de las bibliotecas y de los centros de documentación, donde hace posible la comunicación entre sistemas que utilizan diferente hardware y software.

Hasta el momento ha habido tres versiones del protocolo:

Versión 1: la primera versión del estándar Z39.50 se aprobó en 1988, y ha quedado obsoleta. En 1990 se establecieron dos importantes grupos que garantizan el desarrollo controlado y la continua evolución del estándar: un grupo de implementadores ZIG (Z39.50 Implementors Group) y una agencia para el soporte del estándar (Z39.50 Maintenance Agency).

Versión 2: la versión 2 en 1992 evita las incompatibilidades con el protocolo de ISO "Search and Retrieve" SR (ISO 10162 y 10163). Incluye dos nuevas operaciones: control de acceso de los clientes y control de recursos.

Versión 3: en los años siguientes se sigue trabajando en el desarrollo del estándar y se cuenta con participación de otros países como Canadá, Australia y países europeos; lo que lleva a una versión 3, aprobada en 1995 y aceptada como estándar ISO (ISO 23950) en Marzo de 1997, garantizando así un estándar de aplicación realmente internacional, que incorpora numerosas mejoras, se mantiene compatible con la versión 2 y, sobre todo, reconoce como medio de aplicación TCP/IP e Internet.

En la actualidad, Z39.50 es un estándar maduro, con una amplia presencia en la comunidad bibliotecaria, al menos de algunos países. Para algunos es la norma más importante para el mundo de las bibliotecas y la documentación desde la aparición del formato MARC. Pero, el desarrollo e implementación de Z39.50 convive con la popularización de páginas Web, cada vez mejor diseñadas y potentes, que se han convertido en una manera barata y muy extendida de ofrecer un acceso amigable a la información bibliográfica y cuya funcionalidad crece cada día.

Servicios Z

Los servicios o facilidades principales del estándar son:

1. La inicialización, en la que se establecen los parámetros básicos de la sesión que se va a iniciar entre el cliente y el servidor. Esta negociación incluye la versión del protocolo, las operaciones que podrán efectuarse, juegos de caracteres, lenguas, segmentación y tamaño de la información, etc. Permite asimismo la autenticación del usuario.
2. La búsqueda, funcionalidad más importante del estándar, que permite realizar búsquedas simples o complejas con la misma herramienta a múltiples bases de

datos, agilizando la recuperación de información. Los parámetros de búsqueda, en el caso de los registros bibliográficos, están definidos en el set de atributos Bib-1. Las estrategias de búsqueda pueden utilizar operadores booleanos, de proximidad, etc.

3. La recuperación de la información: una vez realizada la búsqueda, el cliente solicita al servidor los registros que quiere visualizar, que dependiendo del número solicitado, podrán aparecer segmentados en conjuntos de registros.

El estándar ofrece otras muchas facilidades y características adicionales, aunque no es necesario que se implementen todas. Algunas de ellas son: controlar el acceso, realizar búsquedas utilizando índices, ordenar la información recuperada, y poder acceder a información sobre el servidor y los servicios que ofrece. En el ámbito bibliotecario son muy útiles los denominados servicios extendidos que permiten archivar las estrategias y resultados de las búsquedas, actualizar bases de datos, pedir documentos, y crear especificaciones de exportación, entre ellas el formato MARC, El protocolo constituye un gran avance en la interconexión entre sistemas bibliotecarios al permitir superar las enormes barreras que conlleva operar con diferentes sistemas informáticos. En otros ámbitos, como Museos y colecciones digitales, se está trabajando también en la adaptación y ampliación del protocolo, especialmente el CIMI (Consortium for the Computer Interchange of Museum Information).

Atributos Bib-1

El conjunto de atributos Bib-1 se utiliza en la formulación de estrategias de búsqueda por parte del usuario para delimitar la búsqueda. Estos serán mapeados por el servidor a la base de datos. Este proceso es la clave para decidir lo que se recupera en un petición concreta. Tenemos atributos de uso (campos de la búsqueda), de relación, de posición, de estructura, de truncamiento y de completitud.

Bib-1 contiene los siguientes tipos de atributos:

- Atributos de uso
- Atributos de relación
- Atributos de posición

- Atributos de truncación
- Atributos de exhaustividad

Cada tipo de atributos tiene un rango de atributos:

- Atributos de uso
 - Nombre personal
 - Nombre corporativo
 - Nombre de conferencia
 - Título
 - Título de serie
 - Título uniforme
 - ISBN
 - ISSN
 - No. de tarjeta de la LC
- Atributos de relación
 - Menos que
 - Menos que o igual
 - Igual
 - Mayor que o igual
 - Mayor que
 - Diferente
 - Fonético
 - Raíz
 - Relevancia
 - AlwaysMatches
- Atributos de truncación
 - Izquierda
 - Derecha
 - Izquierda y derecha
 - No truncar
 - Procesar

¿Qué hace la Z39.50?

La Z39.50 reconoce que la recuperación de información posee dos componentes principales: la selección de la información basada en algunos criterios, y la recuperación de la información. Y ella proporciona un lenguaje común para ambas actividades. La Z39.50 normaliza la forma en que el cliente y el servidor se comunican e interoperan, aún cuando existan diferencias entre los sistemas computacionales, motores de búsqueda, y las bases de datos.

La arquitectura distribuida cliente/servidor proporciona las bases para la comunicación Z39.50 y las especificaciones de recuperación de información. El software para la interacción del usuario final se separa del software que administra la información, realiza las búsquedas, y devuelve los resultados. La Z39.50 no "dirige" la interface al usuario, pero hay especificaciones y procedimientos del protocolo que pertenecen al cliente Z39.50, tal como es el inicio de la consulta para recuperar información y cómo este requiere de operaciones específicas del servidor. Hay también especificaciones y procedimientos del protocolo que pertenecen al servidor Z39.50, tal como es la manipulación del set de resultados y los formatos en los cuales se devuelven los registros al cliente.

La Z39.50 considera una serie de mensajes iniciales entre el cliente y el servidor, tal como establecer una conexión, iniciar una sesión Z39.50 y negociar las expectativas y las limitaciones de las actividades que ocurrirán (por ejemplo, el tamaño máximo de los registros que serán transferidos desde el servidor al cliente, la versión del protocolo, etc.). Después que se negocia el acuerdo, el cliente puede enviar la consulta. El cliente Z39.50 traduce la consulta a una representación normalizada y la pasa al servidor Z39.50. El servidor ejecuta la búsqueda en la(s) base(s) de datos, y crea un set de resultados. Entonces, el cliente puede solicitar los registros del set de resultados o solicitar al servidor un procesamiento adicional dentro del set de resultados. Al recibir los registros, el cliente puede procesarlos y desplegarlos al usuario. El límite que tiene el cliente para realizar un procesamiento adicional sobre los registros recuperados (por ejemplo, combinar registros de varias búsquedas separadas) dependerá del software de interface al usuario, ya que esto está separado del software cliente Z39.50.

Cada base de datos que reside en servidores de información puede tener características únicas. Por ejemplo, las bases de datos pueden diferir en la forma en que ellas almacenan los datos y en los puntos de acceso disponibles para la búsqueda. Los registros en cada una de las bases de datos también pueden tener diferentes estructuras y consistir de diferentes elementos de datos. El objetivo de la Z39.50 es permitir la comunicación computador a computador en términos normalizados y mutuamente comprensibles, y permitir la transferencia de datos entre los sistemas, independientemente de la estructura, contenido o formato de los datos del sistema en particular

Como trabaja Z39.50

El estándar Z39.50 especifica formatos y procedimientos para gestionar el intercambio de mensajes entre un cliente y un servidor, permitiendo así que el usuario busque en bases de datos remotas, identifique los registros que se ajusten a determinados criterios, y recupere algunos o todos los registros identificados, así como otras informaciones asociadas en función de la base de datos.

La ventaja de cumplir dicho estándar es conseguir un acceso uniforme a un gran número de fuentes de información diversas y heterogéneas, incluso de manera simultánea, superando las diferencias entre los sistemas informáticos, los motores de búsqueda y las distintas bases de datos.

Para conseguir esta interoperabilidad entre distintos sistemas, Z39.50 facilita un lenguaje común para realizar las dos operaciones básicas que garantizan la recuperación de información: selección de información y obtención de la misma. Por ello, Z39.50 contempla la estandarización tanto de los mecanismos de codificación (cómo deben codificarse los datos para ser transferidos), como de la semántica del contenido (modelización de los datos con una semántica común para cada comunidad específica). Desde el punto de vista de los mecanismos de codificación, es importante recordar que Z39.50 simplemente es un estándar de comunicación, es decir, sólo especifica qué debe transferirse, pero adicionalmente necesita un servicio de transporte fiable como TCP. Z39.50 especifica la estructura (Abstract Syntax Notation One, ASN.1) de los mensajes

(Protocol Data Units, PDUs) que se intercambian entre cliente y servidor. Pero, Z39.50 es un estándar muy amplio que ofrece una gran funcionalidad y atiende muy diversos entornos, no sólo el bibliotecario. Ningún desarrollo comercial, ni particular, soporta el estándar completo definido para todos los entornos, aunque el propio estándar describe los mínimos que deben cumplir todos los desarrollos para garantizar la interoperabilidad. Estas diferencias de un desarrollo a otro conllevan ciertos problemas. Los servicios mínimos que deben contemplarse permiten negociar el inicio de una sesión, realizar una búsqueda en una base de datos, crear un conjunto de resultados en dicha base que se ajusten a la búsqueda y obtener uno o varios de los registros del conjunto.

Para garantizar el otro aspecto de la interoperabilidad, conjuntamente a esta codificación estándar de mensajes, Z39.50 se apoya en el concepto de conocimiento de una semántica compartida. Los distintos ámbitos proveedores o consumidores de contenidos de información han ido acordando estructuras y atributos comunes en cada uno de ellos, lo que permite un acceso uniforme a información heterogénea dentro de cada ámbito o comunidad. El modelo de arquitectura básico del estándar Z39.50 se apoya en este concepto de semántica en función del contenido. Es decir, cada servidor ofrece una visión de sus bases de datos en función del dominio o el ámbito en que nos encontremos, una representación virtual de los registros que contiene, donde la estructura lógica real de la base de datos permanece oculta, y sólo se refleja la que corresponde a la semántica de ese dominio.

Adicionalmente a la representación virtual de la base de datos, se definen y registran para cada ámbito los puntos de acceso que se pueden emplear en las consultas (Attribute Sets) y las maneras de estructurar los datos al facilitárselos al cliente (Schemas). En el caso de la comunidad bibliográfica, se ha registrado el Bib-1 Attribute set y distintos formatos para presentar las respuestas, como los distintos formatos MARC, el formato propio del OPAC a que accedemos, etc. Además, se sigue trabajando para ampliar las semánticas registradas en función de las carencias detectadas en la práctica.

En los diferentes ámbitos en que se ido adoptando Z39.50 como estándar para acceder a distintos tipos de información, se han ido definiendo y registrando sus correspondientes Attribute Sets y Schemas. Aunque los puntos de acceso a la información suelen variar mucho de una semántica a otra, los formatos de presentación de los resultados.

Simplificadamente, si efectuamos una búsqueda en una base de datos tradicional nos encontramos con una interfaz de usuario, un motor de búsqueda y los datos.

De esta manera vamos a describir en forma práctica dos características fundamentales de este protocolo como ser:

Arquitectura Cliente y servidor

Un sistema z39.50 sustituye la interfaz de usuario por un servidor-Z, que accede al motor de búsqueda, y varios clientes-Z, que dialogan con el servidor-Z e interactúan directamente con el usuario:

- El cliente-z: la aplicación cliente es un programa utilizado por el usuario final. Mediante este programa se pueden hacer peticiones a los sistemas remotos y se obtienen respuestas de dichos sistemas.
- El servidor: la aplicación servidor es un programa que acepta las conexiones en redes y recibe las preguntas del protocolo z39.50 en dichas conexiones. A partir de estas preguntas elabora una respuesta y la devuelve al cliente. Generalmente, las preguntas que recibe las traduce al formato de consulta de una base de datos y la respuesta contiene los datos recuperados de la base de datos.

La ventaja de este sistema radica en la separación de la interfaz de usuario del motor de búsqueda, lo que significa que una misma interfaz puede ser utilizada con diferentes motores de búsqueda y bases de datos. El servidor se encarga de traducir el lenguaje, de forma que pueda ser comprendido por ambas partes de la cadena.

Una de las funciones básicas de z39.50 define cómo interroga el cliente al servidor. Este proceso se puede dividir en cinco fases:

- El protocolo especifica formatos y procedimientos que deben regir el intercambio de mensajes entre el cliente y el servidor.
- El cliente puede iniciar peticiones; el protocolo dirige la comunicación entre las aplicaciones correspondientes de recuperación de información y entre cliente y servidor.

- El cliente puede proponer una búsqueda, especificando una o más bases de datos, e incluir una pregunta y los parámetros que determinan la forma de presentación del resultado de la búsqueda.
- El servidor responde con una cuenta de registros identificados.
- El cliente asume que los registros seleccionados por la búsqueda componen un conjunto ordenado (según el orden determinado por el servidor).

Búsqueda Z

Un ejemplo de búsqueda desde un OPAC con funciones de cliente-Z, se puede dividir en siete partes:

- El usuario del OPAC selecciona la biblioteca objeto de la búsqueda a través de un menú de OPACs accesible en el cliente.
- El usuario introduce los términos de búsqueda.
- El cliente-Z traduce los términos de búsqueda a un lenguaje común (*Z-Speak*) y establece contacto con la biblioteca.
- Existe una negociación preliminar entre el cliente-Z y el servidor-Z mediante la cual se establecen las reglas de asociación entre los dos sistemas.
- El servidor-Z traduce el lenguaje común (*ZSpeak*) en una petición de información para que pueda ser comprendida por la biblioteca (base de datos) objeto, y recibe la respuesta sobre el número de registros encontrados.
- El cliente-Z recibe los registros
- El OPAC presenta los registros al usuario.

Durante la negociación, el cliente- Z (origen) y el servidor-Z (destino) intercambian una serie de mensajes. Cada uno de esos mensajes es identificado como un “servicio técnico”. Los servicios se pueden agrupar en lo que la norma llama “*facility*”. Existen 11 “*facilities*”:

1. – *Initialization*: Configuración de las reglas de comunicación, negociación de los niveles del servicio
2. – *Search*: Envío de la ecuación de búsqueda a la base de datos y recogida de los primeros resultados
3. – *Retrieval*: Recuperación de registros tal y como se especificó en el cliente-Z.

4. - *Result-set-delete*: Eliminación del conjunto de resultados almacenados en el servidor-Z.
5. - *Access control*: Proceso de verificación por parte del servidor-Z.
6. - *Accounting-resource control*: Gestión de cuentas
7. - *Sort*: Ordenación del conjunto de resultados.
8. - *Browse*: Escaneo e índice en el servidor-Z.
9. - *Extended services*: El cliente-Z puede enviar peticiones de paquetes (préstamo interbibliotecario,...).
10. - *Explain*: El cliente-Z interroga la base de datos en busca de información sobre la implementación del servidor-Z.
11. - *Termination*: Finaliza la negociación y la conexión.

La versión 3 define también el uso de la norma para implementar lo que denomina como “servicios extendidos”. A pesar de no estar incluido dentro de la norma, z39.50 incluye los siguientes servicios para facilitar el control:

- Almacenar resultados.
- Almacenar una *query*.
- Definir un esquema de búsqueda.
- Solicitar un ejemplar.
- Actualizar la base de datos.
- Crear un fichero de exportación.

Otros acuerdos

El estándar también especifica el manejo y ordenación de los juegos de resultados, presentación de índices, apertura y cierre de sesiones y extensiones a otros servicios no definidos en el estándar mismo.

Además, z39.50 define otros aspectos, como:

- Lenguaje de búsquedas basado en juegos de atributos.
- Sintaxis de resultados aceptables (MARC, GRS-1).
- Lenguaje de construcción de resultados para su transferencia.

- Facultad para que el servidor pueda explicar su estructura interna y sus capacidades.

La versión 3 (1995) de la norma z39.50 permite establecer un sistema de búsqueda muy potente, que puede incluir:

- Todos los operadores booleanos (que no implementan en la actualidad la mayoría de los clientes).
- Operadores de comparación de fechas (*greater than, equal to...*).
- Operadores de proximidad.
- Diversas opciones para realizar el truncamiento.
- Búsquedas completas (*part of field, complete field...*)

Además, existen otros rasgos adicionales que ofrecen múltiples posibilidades:

- Autenticación: esto permite que el servidor-Z pueda controlar quién accede a las bases de datos.
- Control de los recursos y de los accesos (cuentas).
- Opción “*explain*”, que permite obtener información sobre bases de datos remotas, servicios disponibles...
- “*Browsing*” del índice.
- Definir el formato de los registros.

Aplicaciones Z

Hasta ahora Z39.50 ha sido utilizado casi exclusivamente por bibliotecarios y utilizando estándares como el formato MARC, pero las sucesivas versiones de la norma están dando cabida a otro tipo de documentos y servicios complementarios que, junto a la universalización del acceso a Internet en curso, hacen que las posibilidades de crecimiento y evolución futuras sean impresionantes.

Las implicaciones para los servicios bibliotecarios y de información son profundas. Las aplicaciones más destacables a efectos de tareas bibliotecarias son:

- OPACs -- Este es el beneficio básico para los usuarios finales, el acceso a las bases de datos más importantes del mundo, o simplemente a fuentes locales con una sola búsqueda.

- Catalogación -- Búsqueda y captura de registros bibliográficos, lo que supone un ahorro de tiempo y trabajo para las bibliotecas. También destaca la posibilidad de construir un catálogo colectivo virtual sin interferir en los métodos y procesos de la organización individual.
- Préstamo interbibliotecario -- Es la consecuencia inmediata de un catálogo colectivo virtual.
- Acceso a CD-ROMs -- Se puede acceder a la información de diferentes CD-ROMs con interfaces únicos, incluso utilizando diferentes clientes.
- Difusión Selectiva de la Información (DSI) -- El usuario puede especificar y grabar estamentos de búsqueda para ser ejecutados posteriormente, pudiéndose ejecutar las búsquedas cuando se quiera.
- Bases de datos comerciales -- Los catálogos no son la única información consultable, ya que existen cientos de proveedores de servicios de información comercial disponibles (Dialog, Lexis Nexis, EBSCO, Chemical Abstracts...). El Z39.50 reduce además la complejidad de las búsquedas en bases de datos muy diferentes.
- Búsqueda web y filtrado -- La búsqueda WEB depende de los diferentes motores de búsqueda e interfaces de usuario, lo que muchas veces la convierte en frustrante. El Z39.50 hace que disminuya la frustración y la pérdida de tiempo.
- Actualización de bases de datos -- No sólo es una herramienta de búsqueda y recuperación, sino que también puede ser utilizado para la actualización de bases de datos.

Ventajas

Las ventajas del Z39.50 aplicado al entorno de las bibliotecas pueden ser muchas, y su importancia dependerá de cada usuario potencial y de sus necesidades. Presenta numerosas ventajas tanto para el usuario final como para el bibliotecario. A grandes rasgos podríamos destacar:

- Relacionar bases de datos diferentes.

- Realizar peticiones simultáneamente a diferentes bibliotecas, propiciando un ahorro de tiempo al realizar búsquedas de ítems poco comunes o que contengan muchos registros.
- Sencillez en la localización de la información sin que el usuario tenga la necesidad de aprender el manejo de los motores de búsquedas de diferentes sistemas y bases de datos.
- Compartir fuentes de información. Permite la localización de información en forma rápida y precisa evitando la compra de fuentes de información disponibles en otros centros.
- Catálogos colectivos virtuales. Permite realizar búsquedas en varias bases de datos de forma sencilla facilitando a los catalogadores intercambiar registros catalográficos ahorrando así recursos en la catalogación y clasificación de los materiales.
- El formato básico de intercambio es el formato MARC.
- Permite facilitar la interconexión entre usuarios de información y las bases de datos donde se encuentra la información que necesitan a partir de una interfaz común y de fácil manejo, independientemente del lugar en que las bases de datos se encuentren, cual sea la estructura de la base de datos y la forma de acceso.

ISIS DLL

Introduccion

La librería de enlace dinámico ISIS_DLL es una implementación de la API ISIS para los sistemas operativos Linux y MS-Windows, desarrollada por BIREME/PAHO/WHO y la UNESCO.

Esta herramienta permite el desarrollo de aplicaciones en entornos graficos y puede ser llamada desde cualquier aplicación escrito en Visual Basic, Delphi, C++, C, Java, Power Builder y cualquier otro lenguaje que soporte llamadas a DLL.

Esta librería es completamente compatible con los estándares de sistemas ISIS, es por ello que pueden coexistir aplicaciones que usan funciones de esta librería, con aplicaciones usando el software ISIS estándar.

Generalidades

Una librería de enlace dinámico es un tipo especial de librerías usada en sistemas operativos MS Windows. Esta se comporta como un modulo ejecutable, cuando algún programa llama a alguna de sus funciones, esta se carga en memoria y puede ser, en adelante, llamada simultáneamente por varias aplicaciones. Esto significa que una copia de la librería puede ser compartida por varias aplicaciones, y que las funciones de la misma son externas al modulo o programa que las usa.

La librería ISIS_DLL es una ISIS API para MS Windows y Linux que contiene funciones específicamente diseñadas para manipular entidades ISIS. Las entidades ISIS principales son: el archivo de registros maestro (Master File Records), el archivo de términos invertidos (Inverted file terms) y las especificaciones del lenguaje de búsqueda y formateo.

Esta librería está diseñada de manera tal que las limitaciones relacionadas con el número de archivos maestros e invertidos abiertos, así como el número de registros y términos cargados en memoria en un momento dado, dependen exclusivamente de los recursos disponibles de la computadora.

Dado que la librería permite no solo atender problemas específicos sino que también interfaces de aplicación no implementadas en sistemas ISIS estándares, esta ofrece a los programadores y desarrolladores de sistemas una gran flexibilidad para el desarrollo de aplicaciones simples o complejas para el procesamiento de bases de datos ISIS.

La librería ISIS_DLL está diseñada por y para programadores, y es altamente recomendado tener un amplio conocimiento de ISIS en orden de usar la librería y explotar todas sus capacidades.

Java Native Interface (JNI)

Java Native Interface (JNI) es un framework de programación que permite que un programa escrito en Java ejecutado en la máquina virtual java (JVM) pueda interactuar con programas escritos en otros lenguajes como C, C++ y ensamblador.

La JNI se usa para escribir métodos nativos que permitan solventar situaciones en las que una aplicación no puede ser enteramente escrita en Java, como por ejemplo en el caso de que la biblioteca estándar de clases no proporcione soporte para funcionalidades dependientes de la plataforma

También se usa para modificar programas existentes escritos en algún otro lenguaje, permitiéndoles ser accesibles desde aplicaciones Java. Muchas de las clases de la API estándar de Java dependen del JNI para proporcionar funcionalidad al desarrollador y al usuario, por ejemplo las funcionalidades de sonido o lectura/escritura de ficheros. El desarrollador debe asegurarse que la API estándar de Java no proporciona una determinada funcionalidad antes de recurrir al JNI, ya que la primera ofrece una implementación segura e independiente de la plataforma.

JNI permite a un método nativo utilizar los objetos Java de la misma forma en que el propio código de Java lo hace. Un método nativo puede crear objetos Java; y examinarlos y utilizarlos para que lleven a cabo su función. Un método nativo puede asimismo examinar y utilizar objetos que han sido creados por código de aplicación escrito en Java.

A menudo se denomina a JNI como la "válvula de escape" para desarrolladores dado que les permite añadir funcionalidades a sus aplicaciones que la API de Java no puede proporcionar.

Dado que -como se ha dicho antes- puede ser usado para interactuar con código escrito en otros lenguajes como C++, también se usa para operaciones y cálculos de alta complejidad temporal, porque el código nativo es por lo general más rápido que el que se ejecuta en una máquina virtual.

JZKIT

JZKIT, es un conjunto de herramientas de la empresa Knowledge Integration, la cual se especializa en el desarrollo de productos y componentes de alta calidad, basados en especificaciones y estándares de código abierto.

JZKIT, desarrollado puramente en Java, está destinado a aplicaciones de búsqueda y recuperación. Al momento de escribir este documento existen tres versiones de esta herramienta (JZKIT, JZKIT 2, JZKIT 3), aunque las últimas dos versiones fueron evoluciones del producto JZKIT original. Estas proveen una estructura modular en la cual se pueden instalar proveedores de búsqueda basada en componentes.

Esta estructura está diseñada para ser un mecanismo verdaderamente independiente del protocolo de búsqueda y el mecanismo de acceso. Este conjunto de herramientas incluye una serie de conectores que abarcan un ancho rango de protocolos de búsqueda incluyendo el Z39.50, SRW y bases de datos relacionales. La configuración por defecto provee una cantidad de facilidades de búsquedas cruzadas y una librería que permite a los usuarios desarrollar sus propios conectores.

ROBLE

ROBLE es un proyecto cooperativo por medio del cual todas las bibliotecas de la UNLP coordinan su esfuerzo para mantener un catalogo colaborativo.

El objetivo del proyecto es el desarrollo de herramientas que a partir de los desarrollos individuales de cada biblioteca, permitan la generación de la Biblioteca Digital Integrada.

El proyecto ROBLE tiene los siguientes objetivos:

- Desarrollar soportes informáticos que permitan integrar fondos bibliográficos existentes en las bibliotecas cooperantes
- Incorporar nuevas facilidades de acceso a la información (referenciales y a texto completo)
- Obtener productos cooperativos (Catálogos cooperativos de libros, revistas y tesis)
- Actualizar y mantener un sitio virtual para ofrecer productos y servicios a los usuarios (Portal Roble)

- Conocer el estado de las bibliotecas y desarrollar indicadores de gestión para el proceso de mejora continua.

Portal ROBLE un sitio que integra el acervo bibliográfico de todas las Bibliotecas de la UNLP a través de "Catálogos Colectivos de Libros, Revistas y Tesis" (OPACs) que facilitan la localización de los documentos por parte de los usuarios. Su objetivo primario responder a las preguntas básicas: Qué hay? y Quién lo tiene?

Es un lugar desde donde todas las Bibliotecas tienen acceso a un conjunto de herramientas que ayudan a realizar las tareas en los diferentes puestos de trabajo. Sirve de enlace permanente para el planteo y discusión de temas comunes y constituye un espacio informativo de apoyo a la gestión y toma de decisiones.

En definitiva, es un espacio virtual con un firme compromiso de fortalecer la imagen institucional de las Bibliotecas de la Universidad Nacional de La Plata, mostrándolas en un todo armónico, desarrollado a partir del esfuerzo de muchos profesionales que hace tiempo están trabajando aisladamente y hoy quieren ser parte del todo.

El catalogo que almacena la información del proyecto es un catalogo ISIS, el cual es accedido desde el portal Roble vía wwwisis.

Desarrollo de la Tesina

Objetivos de la tesina

El objetivo que se persiguió en el siguiente trabajo de grado surge del interés que existe por parte de la biblioteca de la UNLP de integrar su catálogo denominado Roble a la comunidad de servidores Z39.50.

Esto permitirá ampliar la comunidad de usuarios de información catalogada que tiene Roble en este momento, la cual está limitada al conjunto de bibliotecas de la UNLP, es decir, cualquier cliente Z39.50 conectado a internet, podrá utilizar la información que se encuentra en Roble.

Primeros Pasos

Al inicio de esta investigación decidimos abrir la misma en tres grandes temas que nos permitieran avanzar en el conocimiento de la problemática:

- Sistema Roble
- Protocolo z39.50
- Integración de ambos

Sistema Roble

La experiencia que adquirimos con el sistema Roble, se inicia con el material que nos facilitó la Directora de la Biblioteca de la UNLP, la Dra. Norma Mangiaterra, quien nos entregó un CD con la base de datos Roble, que contenía un archivo de intercambio (roble.iso) junto con la definición de campos de la base de datos.

A partir de allí tuvimos la necesidad de estudiar diversas normas y protocolos de catalogación, como así también la forma en que la Biblioteca de la UNLP mantiene el catálogo Roble.

Este catálogo está compuesto, al momento de la investigación, por aproximadamente 250000 registros, y es consultado, de acuerdo a lo investigado, por toda la comunidad académica regional.

Las normas de catalogación investigadas en esta primera etapa fueron ISO 2709, las especificaciones MARC y MARC21 y el entorno de catalogación CDS/ISIS (U.N.E.S.C.O.).

Para poder determinar si era factible el desarrollo de un aplicativo que permitiera lo que nos proponíamos, decidimos hacer unos experimentos prácticos.

En primer lugar, intentamos importar el archivo de intercambio que se nos había facilitado (roble.iso) utilizando alguna herramienta disponible en el mercado.

Para ello decidimos descargar y utilizar la aplicación libre de código abierto JavaIris, y los intentos de importación realizados no fueron exitosos. Al no tener demasiada experiencia con la herramienta, nos surgió el siguiente interrogante: ¿La falla de importación es por error de la aplicación o por error del archivo de intercambio?

Decidimos en principio tomar la hipótesis que la falla estaba en la aplicación JavaIris, por lo que intentamos importar el archivo de intercambio con una herramienta más popular en el mercado.

Este paso nos llevo a la descarga de la aplicación WINISIS, de la UNESCO. Para poder realizar la descarga de esta aplicación fue necesaria una registración previa en el sitio de Internet del Centro Nacional de Energía Atómica, ya que la distribución de la aplicación en Argentina es mediante ese ente.

Utilizando WINISIS, definimos la estructura del catalogo de acuerdo a la información que nos había otorgada previamente la Dra. Mangiaterra y luego intentamos importar el archivo de intercambio. Este procedimiento lo realizamos varias veces obteniendo siempre el mismo resultado no exitoso.

El proceso de importación comenzaba y luego de la incorporación automática de algunos registros, el mismo abortaba y cerraba la aplicación WINISIS automáticamente.

Dada esta circunstancia, nos encontramos ante un nuevo interrogante, porque a diferencia del primer aplicativo, JavaIsis, la aplicación WINISIS sí corría, al menos un instante, y daba la impresión de cargar algunos registros en memoria, pero al abortar la operación no terminaba de concretar el almacenamiento de datos en el catalogo. Además WINISIS es una herramienta respaldada por la UNESCO y distribuida por el CNEA por lo que dudar de la estabilidad de la misma ya no era una opción tan viable.

Por lo tanto en este punto comenzamos a elaborar la hipótesis que posiblemente el archivo de intercambio tuviera alguna falla generada en el momento de creación del mismo, o tal vez, en el momento de grabación al CD que se nos entrego.

Para poder respondernos este interrogante tuvimos una reunión con la Licenciada Mariana Pichinini, responsable del área operativa de la biblioteca FaHCE-UNLP, y le expusimos los problemas que tuvimos para importar la base de Roble. En esta reunión la licenciada nos explico ampliamente acerca del conjunto de archivos que conforman el catalogo ISIS y de algunas características del WINISIS y nos recomendó probar con una herramienta aun más simple que suelen usar en su ámbito.

Luego de la reunión, decidimos instalar una versión de ISIS ejecutable en el sistema operativo DOS, el CDS/ISIS propiamente dicho, basados en la suposición que lo problemas que encontrábamos habían podido originarse en la incompatibilidad de sistemas operativos, o que tal vez, alguna de las librerías que utiliza WINISIS la hubiéramos descargado dañada.

Utilizando entonces el CDS/ISIS volvimos a estructurar el catalogo Roble de acuerdo al formato especificado para poder importar el archivo de intercambio y comenzamos el proceso de importación.

El proceso, repetido varias veces, arrojaba siempre el mismo resultado. Cargaba aproximadamente 100.000 registros, luego automáticamente cancelaba la importación. Al consultar cuantos registros tenía el catalogo, el mismo reportaba que no había ningún registro cargado (MFN=0).

Ya que en algún u otro punto, todas las aplicaciones probadas fallaban en la carga del archivo de intercambio roble.iso, el paso que decidimos seguir fue el de inspeccionar el archivo, con el fin de determinar algún patrón de falla en el mismo.

Intentamos editar el archivo con los editores incorporados en el sistema operativo Windows pero estos no permitían visualizarlo de manera clara, por lo que decidimos descargar el editor gratuito WinVi.

Luego de examinar durante un periodo el archivo de intercambio, tomamos la hipótesis que el problema podría estar en los delimitadores de registro o de campo.

Acto seguido, decidimos, utilizando el WinVil, ir fraccionando el archivo de intercambio para tratar de acotar el error dentro del mismo por lo que tomamos los primeros 4000 registros del archivo y tratamos de importarlos al catalogo usando el CDS/ISIS previamente instalado en DOS.

El resultado de la importación fue exitosa, lo que nos llevo a pensar que posiblemente ahora podríamos tener éxito también con la aplicación WINISIS, que preferíamos por su interface, mucho más amigable.

Con este archivo recortado intentamos la importación utilizando WINISIS, y nuevamente volvimos a obtener un resultado exitoso.

Como para el objetivo de nuestras pruebas tener 4000 registros era lo mismo que 200000, decidimos descartar el resto del archivo, considerándolo dañado, y seguimos adelante con las pruebas de las otras partes del proyecto que nos llevarían al resultado obtenido.

Z39.50

En forma paralela al estudio de Roble hemos analizado protocolo cliente servidor del estándar ISO23950, para lo cual tomamos como base de conocimiento la revisión de la ANSI/NISO z39.50-2003. De la misma ahondamos sobre el modelo y las características del servicio de obtención de información y, además, sobre los servicios adicionales que debíamos proveer para cumplir con la norma

Luego de entender lo básico necesario del protocolo para poder diseñar una solución estratégica comenzamos a evaluar una gran cantidad de herramientas que implementan el mismo.

Finalmente decidimos utilizar una herramienta de la comunidad de software libre, disponible en internet, desarrollado por la empresa developer.k-int.com, el framework JZKit, desarrollado en Java bajo licencia “GNU Public License”

Este paquete de software esta desarrollado en forma de arquitectura modular y permite el agregado de componentes que realizan búsquedas a medida en cualquier recurso electrónico de información.

Esta característica fue la que nos motivo a elegir esta herramienta, dado que a primeras luces nuestra intención era la de escribir un modulo de búsqueda de información en un catalogo Isis a partir de una solicitud hecha vía un cliente Z39.50, y luego, poder retornar el resultado de la búsqueda al mismo cliente.

Al encontrar una herramienta que nos permitiera efectuar nuestra búsqueda particular y encapsularla para ser recibida y enviada utilizando el protocolo Z39.50 decidimos utilizarla como base para el desarrollo pretendido.

Definición y puesta a punto.

Elección de la versión de JZKit.

Un tema que nos ocupo bastante tiempo fue decidir con que versión del framework JZKit nos quedábamos como elegido. La tarea que suponíamos fácil en principio se torno más complicada de lo pensado porque nos encontramos ante la disyuntiva siguiente.

La primera versión del framework JZKit, llamado de esa manera, al momento del desarrollo de la tesina, ya estaba discontinuada, sin desarrollo y con algunas tareas pendientes de acuerdo a lo que se leía en el código fuente.

Por el otro lado teníamos la versión 2 de framework, el JZKit 2, a todas vistas mucho más completo, modular, pensado para búsquedas paralelas en variadas plataformas de almacenamiento de datos y, además, con desarrollo permanente y en estado vigente.

Por lo que luego de evaluar las dos alternativas decidimos, en principio, instalar y probar el JZKit 2. Para ello tuvimos que bajar el proyecto en formato Maven para obtener el código fuente de un repositorio SVN

Desde un comienzo, este framework fue bastante duro de seguir, aun utilizando herramientas de depuración. Cuando finalmente obtuvimos un entendimiento acerca de la manera de modificar el framework para nuestra necesidad, nos dimos cuenta que la tarea se iba a enfocar más en el tratamiento del framework que en el verdadero objetivo, el realizar un módulo de traducción de búsqueda en catálogo CDS/ISIS acoplable al framework.

Aquí decidimos parar para evaluar más en profundidad la primera versión del framework JZKit. Como mencionamos anteriormente este software es más simple y a primera impresión se puede suponer que lo es porque los esfuerzos se dirigieron a la concepción de la segunda versión.

Esta versión del JZkit, aunque con tareas pendientes, nos permitía llevar a cabo lo que queríamos, dado que era posible modificar el código del mismo para acoplarle el módulo que nosotros íbamos a desarrollar por lo que luego de una minuciosa inspección decidimos armar un proyecto utilizando el código fuente de este framework y pusimos a funcionar el servidor Z3950.

Comunicando el servidor con un cliente estándar.

En nuestro primer paso hacia la viabilidad de nuestra propuesta nos dirigimos al sitio web de la librería del congreso y buscamos un cliente Z3950 estándar para de esta manera probar la comunicación con el servidor del framework JZKit.

Descargamos un cliente Z3950 llamado Mercury versión 1.3 y lo instalamos en otra computadora. Configuramos el cliente para acceder a la dirección IP y puerto de la computadora en la cual se encontraba instalado el servidor Z3950 del framework JZKit.

Ejecutamos una búsqueda en el cliente y pudimos comprobar que había comunicación entre el cliente y el servidor pero sabíamos de antemano que la búsqueda no se iba a ejecutar dado que aun no habíamos incluido el modulo de búsqueda determinado en cuestión.

Habíamos dado un pequeño y alentador paso, el software era capaz de atender el pedido de un cliente comercial cualquiera y eso nos llevo dar el siguiente paso, la generación y acoplamiento del modulo de búsqueda en catalogo CDS/ISIS desarrollado por nosotros.

Aprendiendo a prueba y error.

Con el servidor Z3950 funcionando comenzamos a analizar el código fuente del mismo más en detalle para determinar de esta manera, en qué lugar del código era posible insertar nuestro modulo de búsqueda en catalogo CDS/ISIS.

Inherentemente esto no involucro en la tarea de aprender sobre estructuras, tipos de codificación y protocolos para la recuperación y transferencia de datos de información catalogada.

Durante un tiempo el trabajo realizado fue el de lanzar una búsqueda desde el cliente Mercury y seguir el código en el servidor utilizando un depurador observando de esta forma en que parte del servidor la ejecución levantaba una excepción o terminaba abruptamente por algún error.

El primer error encontrado fue en la generación por parte del servidor de la respuesta hacia el cliente en formato usmarc, por lo tanto tuvimos que detenernos en el código del servidor para analizar de qué manera estaba armando esta respuesta.

Esta tarea nos sumergió en el estudio de varias clases de la librería del framework JZKit. El proceso de construcción de respuesta involucra la transformación de los datos de una instancia de clase Java a un documento XML, a través del uso de plantillas XLS, y finalmente el envío de la respuesta en formato usmarc el cual es un formato estándar de transferencia y recepción de datos.

Siguiendo el depurador encontramos que el error era provocado por una clase empaquetada en una librería adjunta al proyecto JZKit de la cual no teníamos código fuente, por lo cual, decidimos buscar un de-compiler de bytecodes Java y pasándole

varios archivos de clase compilados de la librería adjunta obtuvimos los códigos fuente respectivos.

Nuevamente, ahora con todos los fuentes necesarios, volvimos a ejecutar varias veces las búsquedas hasta encontrar que existía un error de programación en el proyecto JZKit relacionado con la conversión de objetos a XML que luego eran utilizados para transferir al cliente en formato usmarc. Específicamente esta tarea nos derivó en la re-codificación de varias clases, plantillas xls y documentos de propiedades necesarios para la configuración del framework.

Una vez finalizada esta etapa concretamos otro importante paso para el proyecto que fue el de devolver la respuesta al cliente Mercury. Si bien en principio la respuesta era de prueba, dado que armábamos de manera constante y fija un conjunto de registros, vimos realizado nuestro objetivo en ese punto del desarrollo, el de poder recibir un pedido de búsqueda y devolverlo en el formato convenido.

Implementando la búsqueda en el catalogo CDS/ISIS.

A partir de aquí, nuestro siguiente paso fue sumergirnos en el mundo ISIS para entender como se componía el catalogo, como se realizaban las búsquedas, como se recuperaba la información del mismo y como se actualizaba el catalogo.

Estudiamos a fondo la librería que utilizamos finalmente de apoyo para la ejecución de las búsquedas, la isis.dll, sus funciones y características.

Esta librería nos permitió embeberla en nuestro código Java y utilizando las funciones que la misma provee fue que comenzamos a definir nuestro modulo de búsqueda.

Fijamos como objetivo y para poder establecer una meta clara, que solo permitiríamos en esta instancia búsqueda por Autor y/o por Título. De esta manera dejamos a un lado la dificultad de tener que determinar la búsqueda por una serie de campos interminables. Concluimos que si la búsqueda era bien efectuada utilizando estos campos como filtro, no habría inconveniente alguno en realizar la búsqueda por los demás campos.

Aquí pudimos determinar que la búsqueda en el catalogo era muy dependiente de cómo estuvieran armados los archivos indexados que el mismo utiliza cuando se le envía una solicitud de consulta.

Tuvimos que lidiar con bastantes errores inmersos en el código mismo del framework JZKit, pero específicamente un error que nos complicó durante un tiempo estaba relacionado con la cantidad de registros de respuesta que el servidor le retornaba al cliente. A raíz de este error el cliente Mercury siempre obtenía una respuesta errónea.

Solucionado este inconveniente mediante la modificación del código fuente del framework, estudiamos las estructuras necesarias para la conversión de las consultas recibidas por el servidor y las estructuras para la conversión de los datos que se eran extraídos del catálogo.

Las estructuras involucradas en la conformación de una consulta tenían forma de árbol, mientras que la información recuperada del catálogo tenía estructura de lista de registros.

Nuestro siguiente paso fue convertir la estructura de árbol de la consulta a un formato simple de interpretación para poder realizar la búsqueda en el catálogo. Una vez realizada esta conversión el siguiente paso era realizar la búsqueda utilizando la librería isis.dll. Entonces construimos una clase de testeo que realizara esta conversión y obtuviera solo algunos campos del catálogo.

Cuando logramos tener éxito en este paso, ya nos encontrábamos muy cerca del objetivo propuesto en la tesis.

Refinando el módulo para su integración con proyecto KOHA-UNLP.

Ya con la consulta al catálogo CDS/ISIS funcionando a través de la librería isis.dll, surgió la posibilidad de utilizar este proyecto como un servicio más del producto KOHA-UNLP por lo cual decidimos comenzar a expandir la cantidad de campos de resultados de la búsqueda.

Para ello nos contactamos con los responsables del proyecto KOHA-UNLP y le solicitamos que nos enumeraran los campos con los que ellos suelen tratar en el intercambio de información catalogada.

Una vez contamos con esa lista modificamos el módulo para extraer de los registros del catálogo todos los campos y subcampos necesarios para el proyecto KOHA-UNLP.

Cuando tuvimos el modulo terminado volvimos a hacer todas las pruebas con el cliente Mercury y obtuvimos resultados satisfactorios, por lo cual decidimos que el producto funcionaba y organizamos entonces una prueba de integración con el proyecto KOHA-UNLP.

Prueba de integración con proyecto KOHA-UNLP.

Organizada la reunión de prueba, se instalo nuestro proyecto en un servidor externo a la UNLP que se nos fue facilitado para que pudiéramos efectuar la misma. Nuestro servidor quedo funcionando a la espera de pedidos de consultas.

Posteriormente nos dirigimos al laboratorio LINTI de la facultad de informática de la UNLP y mantuvimos una reunión con los responsables del proyecto KOHA-UNLP. Durante la reunión se hicieron varias pruebas para que el cliente Z3950 que ofrece KOHA-UNLP como servicio se conectara contra nuestro servidor y le efectuara consultas.

Luego de hacer los cambios necesarios en el proyecto para que pudiera acceder a nuestro servidor, las pruebas realizadas durante nuestra reunión fueron exitosas.

Conclusión.

Finalizadas las pruebas en las que integramos nuestro proyecto de servidor con un producto generado por un equipo especializado de la facultad de informática de la UNLP, pudimos concluir que llegamos a encontrar una solución exitosa posible.

La propuesta original fue la de conectar el catalogo Roble con el mundo Z3950, y eso, finalmente, fue lo que logramos.

Configuramos un servidor Z3950 a través del cambio de su código fuente para que atendiera pedidos y los resolviera consultando el catalogo ROBLE en formato CDS/ISIS.

Futuros Trabajos.

A partir de aquí, las posibilidades son numerosas dado que el código fuente permite las modificaciones necesarias para implementar las búsquedas más variadas y los procesamientos de las mismas de acuerdo a las necesidades o servicios que se quieran brindar.

Aporte.

Con este proyecto quisimos contribuir a que la UNLP se convierta en el primer catalogo argentino Z3950 registrado en la Librería del Congreso de los EE.UU y también pueda utilizarse como server de testeo registrado en dicha Librería.

Referencias

Referencias a sitios

<http://www.k-int.com>.

<http://www.cindoc.cis.es>

<http://www.roble.unlp.edu.ar>

<http://www.unesco.org>

<http://www.cnea.gov.ar>

<http://regional.bvsalud.org/php/index.php>

<http://java.sun.com/j2se/1.4.2/docs/guide/jni/>

<http://web.tiscali.it/javaisis/>

<http://java.sun.com>

www.loc.gov/z3950/gateway.html

<http://www.loc.gov/marc/marcspace.html>

<http://www.winvi.de/es/>

http://es.wikipedia.org/wiki/Java_Native_Interface

<http://maven.apache.org/>

<http://www.loc.gov/z3950/agency/zip/meetings/oclc2002/esns.html>

<http://subversion.tigris.org>

http://www.sirsidynix.com/Resources/Pdfs/Company/Z3950_Multilis.pdf

<http://www.cindoc.csic.es/isis/01-3.htm>

http://productos.bvsalud.org/reference/pt/cisis_es/Estructura.htm

http://catalis.uns.edu.ar/wiki/index.php/International_Standard_Bibliographic_Descriptor

<http://www.cindoc.csic.es/isis/ap-b.htm>

<http://catalis.uns.edu.ar/wiki/index.php/IFLA>

http://catalis.uns.edu.ar/wiki/index.php/MARC_21

<http://www.loc.gov/marc/authority/spanish/ecadintr.pdf>

http://www.ifigenia.es/zeta/z39.50_bib-1.html

<http://www.bl.uk>

Referencias Bibliograficas

ANSI/NISO Z39.50-2003 ISSN: 1041-5653 (revision of Z39.50-1995)

Information Retrieval (Z39.50): Application Service Definition and Protocol Specification

ISIS Application Program Interface - ISIS_DLL User's Manual