

Universidad de Los Andes
Facultad de Ingeniería
Postgrado de Computación

Propuesta e implantación de un modelo de integración
bibliotecaria con el protocolo Z39.50

www.bdigital.ula.ve

Valeria Karolina León Bravomalo
Tutor: Dr. Leandro León

Mérida – Venezuela

Noviembre/2003

C.C.Reconocimiento

Resumen

Este trabajo aborda la problemática de interactuar con diversas interfaces para poder acceder a catálogos electrónicos mantenidos por sistemas bibliotecarios diferentes. Propone un modelo de integración bibliotecaria, llamado MIZ, que plantea la disminución de las restricciones derivadas de la heterogeneidad de sistemas distribuidos, mediante la preservación de las interfaces: la usada por un usuario extremo para consultar información y la que usa un sistema bibliotecario para publicar sus contenidos.

El MIZ está basado en el protocolo *Z39.50* como lenguaje común para el intercambio de información. La técnica para no afectar las interfaces es utilizar traductores que hacen la intermediación entre sistemas. Adicionalmente, el MIZ provee dos gramáticas que facilitan la interrogación a múltiples fuentes de información y la interpretación de sus respuestas.

Esta tesis muestra la implementación del MIZ en el contexto del sistema *Alejandro*. Aspectos fundamentales de la implementación son el manejo de traductores y de sesiones concurrentes entre los extremos *interrogador* y *proveedor*.

Palabras clave: *Z39.50*, MARC, integración bibliotecaria, preservación de interfaces, traducción, sesiones concurrentes.

www.bdigital.ula.ve

Para Germán, Camila y Valeria Sofía

www.bdigital.ula.ve

Reconocimientos

Este trabajo fue posible por la contribución de varias personas e instituciones a las que quiero agradecer.

La valiosa colaboración de mi amigo y profesor, José Gregorio Silva, fue vital en la implementación y pruebas del software, pues largamente discutimos y programamos en conjunto con nuestros compañeros de CITEC - ULA y Hacer Sistemas, empresas que me prestaron sus recursos, humanos y materiales, y proporcionaron el contexto en el cual fue desarrollada esta tesis que ya es el corazón de algunos productos comerciales.

Mi tutor, el profesor Leandro León, fue un guía certero en el desarrollo de los conceptos expuestos en este trabajo y de quien he aprendido muchísimo.

Mi esposo, Germán, me animó a seguir adelante y culminar este trabajo.

A todos ellos, muchas gracias.

www.bdigital.ula.ve

Índice general

1	INTRODUCCIÓN	9
2	PLANTEAMIENTO FORMAL DEL PROBLEMA Y POSIBLES SOLUCIONES	11
2.1	Entorno bibliotecario.....	11
2.2	Antecedentes de la interacción bibliotecaria.....	13
2.2.1	<i>Formato MARC.....</i>	<i>13</i>
2.2.2	<i>Telnet</i>	<i>13</i>
2.2.3	<i>Html/Http.....</i>	<i>14</i>
2.3	Heterogeneidad en automatización de bibliotecas	14
2.3.1	<i>Las interfaces de un sistema bibliotecario</i>	<i>14</i>
2.3.2	<i>El problema de las interfaces</i>	<i>16</i>
2.4	El objetivo de este trabajo	16
2.4.1	<i>Los mecanismos de interoperabilidad entre sistemas bibliotecarios</i>	<i>17</i>
2.4.2	<i>Modelo 1: Traducción Local.....</i>	<i>18</i>
2.4.3	<i>Modelo 2: Traducción Remota</i>	<i>18</i>
2.4.4	<i>Mecanismos de implantación de interoperabilidad</i>	<i>20</i>
2.4.5	<i>Modelo 3: Uso de un lenguaje común</i>	<i>20</i>
2.5	Conclusión.....	21
3	BREVE REVISIÓN DEL PROTOCOLO Z39.50	23
3.1	El estándar Z39.50	24
3.1.1	<i>Reseña histórica y actualidad</i>	<i>25</i>
3.1.2	<i>Arquitectura del protocolo Z39.50</i>	<i>27</i>
3.1.3	<i>Servicios básicos del protocolo Z39.50.....</i>	<i>28</i>
3.1.4	<i>Servicios secundarios del protocolo Z39.50</i>	<i>30</i>
3.1.5	<i>Gramática de servicios y registros</i>	<i>31</i>
3.2	El formato MARC	31
3.2.1	<i>Estructura del registro MARC.....</i>	<i>32</i>
3.3	Conclusión.....	37

4	UN MODELO DE INTEGRACIÓN BIBLIOTECARIA CON Z39.50.....	38
4.1	Consideraciones de diseño del MIZ	39
4.2	Arquitectura final del MIZ.....	42
4.2.1	<i>Modelado de objetos del MIZ en UML</i>	47
4.2.2	<i>Lenguajes del MIZ</i>	50
4.3	Conclusión.....	53
5	UNA IMPLANTACIÓN DEL MIZ.....	54
5.1	Traducción.....	55
5.1.1	<i>Requerimientos de traducción</i>	57
5.1.2	<i>Diseño de los traductores</i>	59
5.2	Sesión	64
5.2.1	<i>Diseño de subsesión interrogadora</i>	66
5.2.2	<i>Diseño de subsesión Z39.50</i>	67
5.2.3	<i>Diseño de subsesión proveedora</i>	68
5.2.4	<i>Manejo de sesiones concurrentes</i>	69
5.2.5	<i>Integración</i>	72
5.3	Implantación y pruebas	73
5.3.1	<i>Bibliotecas utilizadas</i>	74
5.3.2	<i>Compilador y programas</i>	75
5.3.3	<i>Pruebas</i>	75
5.4	Conclusión.....	77
6	CONCLUSIÓN.....	78
7	ANEXO I	80
7.1	Sección A- Lenguaje Intermedio de Interrogación	80
7.2	Sección B –Lenguaje Intermedio de Respuesta	81
8	ANEXO II.....	82
9	BIBLIOGRAFÍA Y REFERENCIAS.....	84

Índice de figuras

Figura 2.1: Interfaces de un sistema bibliotecario	15
Figura 2.2: Interacción de las interfaces con sus sistemas bibliotecarios, sus usuarios y bases de datos.....	15
Figura 2.3 : Sistemas bibliotecarios heterogéneos	16
Figura 2.4: Traducción hecha por el sistema local.....	18
Figura 2.5: Traducción hecha por el sistema remoto	19
Figura 2.6: Traducción hecha por elementos traductores en cada sistema	21
Figura 3.1: Arquitectura Cliente/Servidor del protocolo Z39.50.....	27
Figura 3.2: Ejemplo de uso del protocolo Z39.50 en bibliotecas	29
Figura 3.3: Estructura de un registro MARC.....	33
Figura 3.4: Transmisión de registros MARC.....	36
Figura 4.1: Interacción de las aplicaciones bibliotecarias A y B a través del MIZ.....	39
Figura 4.2: Uso del protocolo Z39.50 y de las interfaces humano computadora y de base de datos en el MIZ.	40
Figura 4.3: Cliente Z39.50 y Servidor Z39.50 dentro del MIZ.	41
Figura 4.4: Traductores en el MIZ.....	42
Figura 4.5: MIZ Interrogador y MIZ Proveedor con traductores.....	42
Figura 4.6: Arquitectura final del MIZ.....	43
Figura 4.7: Vista de cada componente del MIZ.....	45
Figura 4.8: Lenguajes de interacción entre los componentes del MIZ.....	45
Figura 4.9: Clases componentes del MIZ.....	47
Figura 4.10: Clases traductoras del MIZ	48
Figura 4.11: Clases cliente y servidor Z39.50.....	49
Figura 4.12: Transformación de una instrucción en LB a LII	50
Figura 4.13: Transformación de registros en LZ39.50 a LIR.....	51
Figura 5.1: Dinámica del Traductor Interrogador.....	55
Figura 5.2: Petición en Z39.50 a una sola base de datos	56
Figura 5.3: Dinámica de los servidores Z39.50.....	57
Figura 5.4: Submódulos del traductor interrogador relacionados con la búsqueda.....	60
Figura 5.5: Traductor interrogador relacionado con la respuesta	61
Figura 5.6: Grupo de submódulos del traductor cliente Z39.50 que construyen la interrogación para varios servidores Z39.50.....	62
Figura 5.7: Grupo de submódulos del traductor cliente Z39.50 responsables de la respuesta.....	62
Figura 5.8: Grupo de submódulos del traductor proveedor relacionados con la elaboración de la interrogación.....	63
Figura 5.9: Grupo de submódulos del traductor proveedor responsables de construir la respuesta.....	64
Figura 5.10: Subsesiones que deben mantenerse para mantener la sesión entre los extremos	65
Figura 5.11: Implementación de subsesión interrogadora	66
Figura 5.12: Implementación de subsesión Z39.50	68

Figura 5.13: Implementación de subsesión proveedora.....	69
Figura 5.14: Manejo de sesiones interrogadoras concurrentes	70
Figura 5.15: Manejo de sesiones Z39.50 concurrentes.....	71
Figura 5.16: Manejo de sesiones proveedoras concurrentes	72
Figura 5.17: Integración de traductores y sesiones concurrentes en el MIZ....	73

www.bdigital.ula.ve

1 INTRODUCCIÓN

El tema de esta tesis es la recuperación de información bibliotecaria distribuida en servidores heterogéneos de información, el cual se trata desde dos puntos de vista, desde el usuario que requiere la información y desde la base de datos que la provee. Del lado del usuario, se estudia la necesidad de consultar información proveniente de servidores heterogéneos mediante la interfaz del sistema local. Del lado de la base de datos, se estudia como proveer información a múltiples usuarios utilizando la interfaz de base de datos del sistema proveedor. La confluencia de estas dos capacidades: la de interrogar múltiples servidores heterogéneos con una sola interfaz y la de proveer información con un solo conjunto de oraciones, genera un alto grado de integración entre sistemas.

El mundo en el que vivimos da cabida a la variedad y la heterogeneidad en todos los ámbitos del quehacer humano. Esta diversidad es el resultado de diferentes formas de enfocar y resolver problemas. Las bibliotecas no escapan a esta realidad. Existe un sin número de fuentes de información y sistemas que los manejan. Mediante Internet, un usuario no sólo se circunscribe a su sistema bibliotecario local para buscar información, sino que, también, puede acceder a otras bibliotecas que le sean útiles. Para acceder a una biblioteca, se requiere conocer la sintaxis y comandos de su interfaz. Esto ha hecho que la interacción entre múltiples sistemas no sea tan fácil como se quisiera.

Para manipular un sistema bibliotecario se necesita conocer su interfaz. A raíz de las numerosas interfaces de las diferentes bibliotecas alrededor del mundo, surgen planteamientos para facilitar la interacción. Uno de ellos es uniformizar el lenguaje de consulta. Con esta idea no se propone que todos los sistemas bibliotecarios sean iguales, sino que se comuniquen con el mismo lenguaje.

Un problema a resolver consiste en que la interrogación a varias fuentes de información necesita del uso de múltiples formas de búsqueda; lo que implica que el usuario debe dominar la sintaxis y los comandos propios de cada interfaz o que, internamente, el sistema local deba conocer la estructura de cada base de información para interrogarla usando la sintaxis apropiada. Esta tesis muestra un mecanismo único de búsqueda y recuperación de información con el cual el usuario no resuelve la heterogeneidad de las fuentes de información, sino una herramienta de software intermediaria entre los diversos sistemas.

Esta tesis presenta un modelo de integración que usa el protocolo Z39.50 como lenguaje común de interrogación y de respuesta. El mecanismo de comunicación entre clientes y servidores de información bibliotecaria es a través del protocolo Z39.50, el cual se está convirtiendo en un estándar para el intercambio de información entre usuarios de bases de información. El modelo usa una sola sintaxis de búsqueda para obtener información de fuentes heterogéneas y, permite al proveedor de información, manejar un sólo lenguaje para entender la petición y hacer la recuperación.

El principal resultado de esta tesis es una propuesta del modelo de integración bibliotecaria y su implementación exitosa. Se evidencia, entonces, que aplicaciones heterogéneas pueden interactuar entre sí haciendo transparente la complejidad inherente a la interoperabilidad de sistemas.

Adicionalmente se tienen otros logros: un buen aprovechamiento de las capacidades del protocolo Z39.50; manejo de peticiones concurrentes a varios servidores; búsquedas concurrentes en varias bases de información; establecimiento y manejo de sesión entre el usuario que inicia la búsqueda y el servidor de base de datos.

La tesis se estructura así: el segundo capítulo plantea el problema de la multiplicidad de interfaces y algunos enfoques para resolverlo; el tercer capítulo presenta el protocolo Z39.50; el cuarto capítulo propone un modelo que resuelve el problema planteado y, el quinto capítulo describe la implantación de este modelo en la aplicación bibliotecaria Alejandría[1].

2 PLANTEAMIENTO FORMAL DEL PROBLEMA Y POSIBLES SOLUCIONES

El desarrollo de este trabajo se enmarca dentro del sistema Alejandría, el cual es un sistema de manejo de información que tiene los siguientes elementos:

- Administrador del material bibliográfico.
- Motor de búsqueda y recuperación de información.
- Interfaz humano computadora para consulta y despliegue de información.
- Interfaz de bases de datos.

www.bdigital.ula.ve

Una fase que se pretende con Alejandría es que sea capaz de interactuar con otros sistemas bibliotecarios. Para ello se plantean ciertos requerimientos que se describen en este capítulo. Alejandría es el contexto en el que se verifica la resolución del problema que surge como resultado de estos requerimientos.

2.1 Entorno bibliotecario

En una biblioteca pueden identificarse diferentes procesos y servicios:

Procesos:

- Catalogación del material.
- Adquisición.

Servicios:

- Préstamos.
- Bases de datos en CD-ROM.
- Divulgación de información.
- Acceso a bases de datos no bibliográficas.
- Búsquedas de catálogos en Web.
- Intercambio de información entre bibliotecas:
 - Consultar a catálogos.
 - Proveer contenidos.

Para llevar a cabo con eficiencia estos procesos y servicios, las bibliotecas han desarrollado sistemas bibliotecarios automatizados y estándares que les han permitido intercambiar información.

Los procesos y servicios que se han automatizados son:

- Adquisición.
- Catalogación.
- Préstamos.
- Recuperación de información local y remota.

Entre los estándares desarrollados para hacer posible la cooperación interbibliotecaria están:

- Inter Library Loan (ILL)[2]: para hacer préstamos remotos.
- Machine Readable Cataloging (MARC)[3]: rige la sintaxis de los registros que se intercambian, y hace posible su importación a un sistema bibliotecario para el proceso de catalogación.
- Search and Retrieve (SR)[7]: para hacer búsqueda y recuperación de información.
- Z39.50[5]: que permite la interconexión entre bibliotecas con sistemas heterogéneos. Z39.50 utiliza los estándares anteriores.

2.2 Antecedentes de la interacción bibliotecaria

La interacción bibliotecaria ha evolucionado, desde la definición de formatos de registros para permitir su exportación e importación, hasta formas más complejas en las cuales las fronteras geográficas no son limitantes.

2.2.1 Formato MARC

MARC (Machine Readable Cataloguing), desarrollado y mantenido por la Biblioteca del Congreso de Estados Unidos, es el estándar que las bibliotecas usan para la catalogación de su material. Este formato contiene reglas para la creación de registros entendibles por cualquier sistema computarizado de recuperación bibliográfica. Los registros son las unidades de datos recuperados por un sistema y tienen campos como autor, título, editorial, materia, etcétera.

El formato MARC hace posible que las bibliotecas compartan mejor los recursos bibliográficos, adquieran datos ya catalogados, usen sistemas bibliográficos existentes en el mercado y reemplacen un sistema bajo la garantía de que sus datos sean compatibles con el nuevo sistema. En el capítulo tres se explica con más detenimiento las reglas de formación de registros bajo este formato.

Durante la década de los 60, las bibliotecas empezaron a catalogar usando el formato MARC para poder intercambiar registros con otras bibliotecas. La interacción se llevaba a cabo manteniendo catálogos electrónicos en MARC e intercambiándolos usando medios de almacenamiento electrónicos.

2.2.2 Telnet

Durante la década de los 80, sobre TCP/IP, mediante el servicio Telnet, las bibliotecas pusieron sus catálogos a la disposición de cualquier usuario autorizado. Así, un usuario remoto hacía un login remoto al sistema

bibliotecario y actuaba como si tuviera un terminal conectado a este sistema. Así podía interactuar con el catálogo y usar los servicios de la biblioteca.

Cada sistema bibliotecario tenía su propia interfaz. Para interactuar con ella, el usuario debía usar los comandos propios del sistema remoto.

2.2.3 Html/Http

En la última década del siglo pasado, las bibliotecas publicaron sus catálogos en el Web. Haciendo uso de las interfaces Web, los usuarios aprendían sus sintaxis de búsqueda y recuperación e interactuaban con el sistema bibliotecario para recuperar información.

2.3 Heterogeneidad en automatización de bibliotecas

Los sistemas bibliotecarios tienen diversos puntos de heterogeneidad:

- Interfaces.
- Sintaxis de búsqueda y recuperación.
- DBMS.
- Lenguajes de programación.
- Hardware.

Esta heterogeneidad es una característica deseable de preservar porque enriquece la diversidad de maneras de manejar la información. Sin embargo, genera limitaciones en la integración pues es necesario el aprendizaje de los elementos por cada sistema bibliotecario particular.

2.3.1 Las interfaces de un sistema bibliotecario

La interfaz humano computadora es el medio que un usuario tiene para interactuar con su sistema bibliotecario. La interfaz de la base de datos es el puente a través del cual una aplicación publica los contenidos de sus bases de datos. En la figura 2.1 se ilustran estas interfaces.

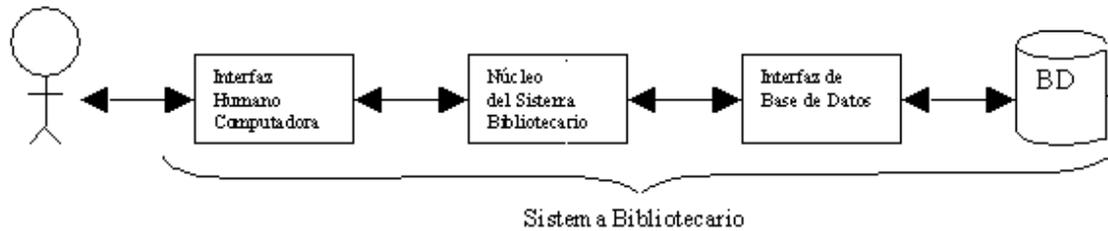


Figura 2.1: Interfaces de un sistema bibliotecario

La interfaz humano computadora está compuesta de elementos tales como:

- Idioma.
- Lenguaje de consulta que el usuario usa.
- Entorno gráfico.
- Lenguaje de interacción con el sistema bibliotecario.

La interfaz con la base de datos tiene:

- Lenguaje de interacción con la base de datos.
- Lenguaje de interacción con el sistema bibliotecario.

En la figura 2.2 se desglosa las interfaces en sus componentes.



Figura 2.2: Interacción de las interfaces con sus sistemas bibliotecarios, sus usuarios y bases de datos.

2.3.2 El problema de las interfaces

Las bibliotecas pueden tener diferentes sistemas, los cuales eventualmente implican interfaces distintas como se muestra en la figura 2.3. Esta implicación significa que:

- Un usuario final necesita interactuar con diferentes interfaces humano computadora a la hora de acceder a diferentes bases de datos.
- Los sistemas proveedores de información necesitan cambiar sus interfaces de bases de datos para publicar sus contenidos en otros sistemas.

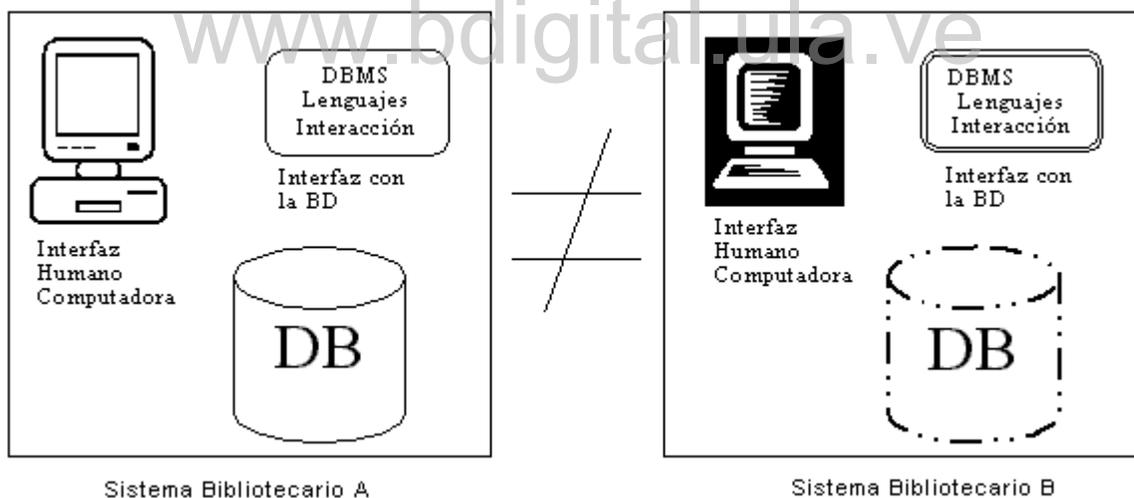


Figura 2.3 : Sistemas bibliotecarios heterogéneos

2.4 El objetivo de este trabajo

El objetivo de este trabajo es proponer una alternativa que permita la interoperabilidad entre sistemas de recuperación de información heterogéneos,

afectando lo menos posible sus interfaces humano computadora y de base de datos.

La alternativa propuesta deberá ser factible de implementar y eficiente con respecto al manejo de la heterogeneidad.

2.4.1 Los mecanismos de interoperabilidad entre sistemas bibliotecarios

Los elementos de los sistemas bibliotecario que se desean preservar son:

- Interfaz humano computadora.
- Interfaz con la base de datos.

Para facilitar el discurso, emplearemos los siguientes términos:

- Interrogadores de información.
- Proveedores de información.

El objetivo de un interrogador es brindar a sus usuarios la posibilidad de interactuar con diversos sistemas a través de su interfaz humano computadora. El objetivo del proveedor es publicar los contenidos de sus bases de datos en distintos sistemas. En ambas vistas está presente la necesidad de interoperar con sistemas heterogéneos.

Para conservar las interfaces es necesario que los sistemas hagan traducciones de sus lenguajes a los lenguajes de los sistemas con los que interactúan.

A continuación se describen tres modelos en los que el proceso de traducción es realizado de manera diferente.

2.4.2 Modelo 1: Traducción Local

En este modelo, la traducción la hace algún elemento del sistema bibliotecario local.

Si el sistema local va a ser interrogador, entonces, las consultas hechas por sus usuarios se traducen al lenguaje del sistema del que desea recuperar información. Si el sistema local va a ser proveedor, entonces, se traducen los registros de sus bases de datos al lenguaje del sistema en el que se quieren publicar. En la figura 2.4 se ilustra este modelo.

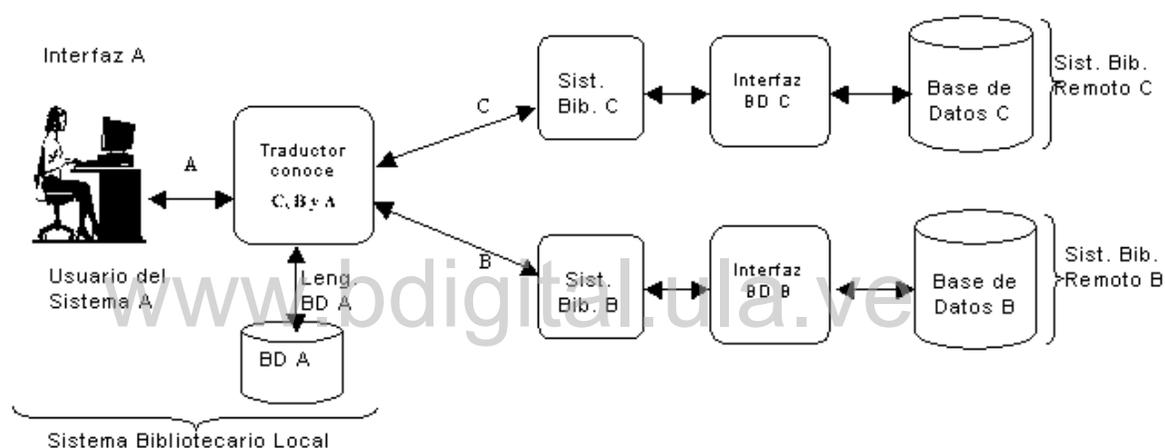


Figura 2.4: Traducción hecha por el sistema local

Las características de este modelo son:

- El traductor del sistema local tiene que conocer todos los lenguajes de los sistemas con los que interactúa.
- Si surge un nuevo sistema con el que quiera intercambiar información, hay que implantar la parte correspondiente al lenguaje del nuevo sistema.

2.4.3 Modelo 2: Traducción Remota

En este modelo, la traducción la hace algún elemento del sistema bibliotecario remoto.

El sistema que lo interroga envía las consultas en su lenguaje y el sistema remoto las traduce a su propio lenguaje. El sistema que publica los contenidos de sus bases de datos, envía los registros en su sintaxis y el sistema remoto los traduce a la sintaxis de su base de datos. En la figura 2.5 se ilustra este modelo.

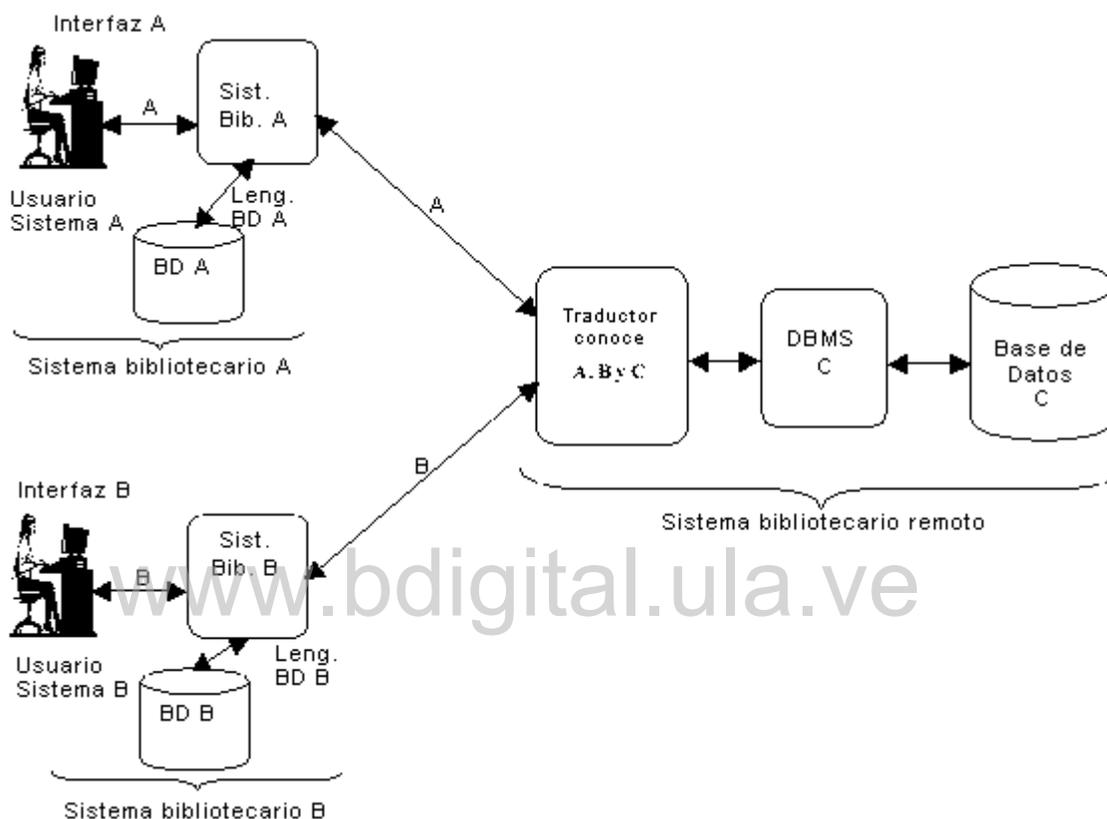


Figura 2.5: Traducción hecha por el sistema remoto

Las características de este modelo son:

- El traductor del sistema remoto tiene que conocer los lenguajes de los sistemas con los que interactúa.
- Si surge un nuevo sistema que lo interroga o que publique su bases de datos en el sistema remoto, entonces, éste debe aprender un nuevo lenguaje.

2.4.4 Mecanismos de implantación de interoperabilidad

Los modelos de traducción local y remoto se pueden implantar bajo los siguientes esquemas:

2.4.4.1 Integración estática

Con este esquema, el traductor está programado con todos los lenguajes de los sistemas con los que interactúa. Si surge un nuevo sistema con un nuevo lenguaje, implica que el traductor debe ser reprogramado para poder interactuar con él.

Una limitación de la integración estática es que, para cualquier sistema nuevo, el traductor debe reprogramarse para incluir el nuevo lenguaje.

El tamaño del traductor crece arbitrariamente en función de los diferentes sistemas que entran en la interacción.

2.4.4.2 Traductor inteligente

Con este esquema, el traductor está preparado para aprender los lenguajes de los sistemas con los que interactúa. Si surge un nuevo sistema, el traductor posee las suficientes reglas de aprendizaje para adquirir el conocimiento de traducir de su lenguaje a cualquier lenguaje.

La limitación del traductor inteligente es su viabilidad, pues es difícil implementar un sistema que tenga la suficiente capacidad de aprendizaje como para descubrir las primitivas de búsqueda y las sintaxis de registros de cualquier sistema.

2.4.5 Modelo 3: Uso de un lenguaje común

En este modelo, los dos sistemas que interactúan utilizan un lenguaje común para intercambiar información.

El sistema que interroga traduce su consulta al lenguaje común. El interrogado traduce del lenguaje común a su lenguaje. La respuesta sufre las

traducciones inversas. El sistema que provee información traduce los registros de su base de datos al lenguaje común y el sistema destinatario lo traduce a su lenguaje. En la figura 2.6 se presenta este modelo.

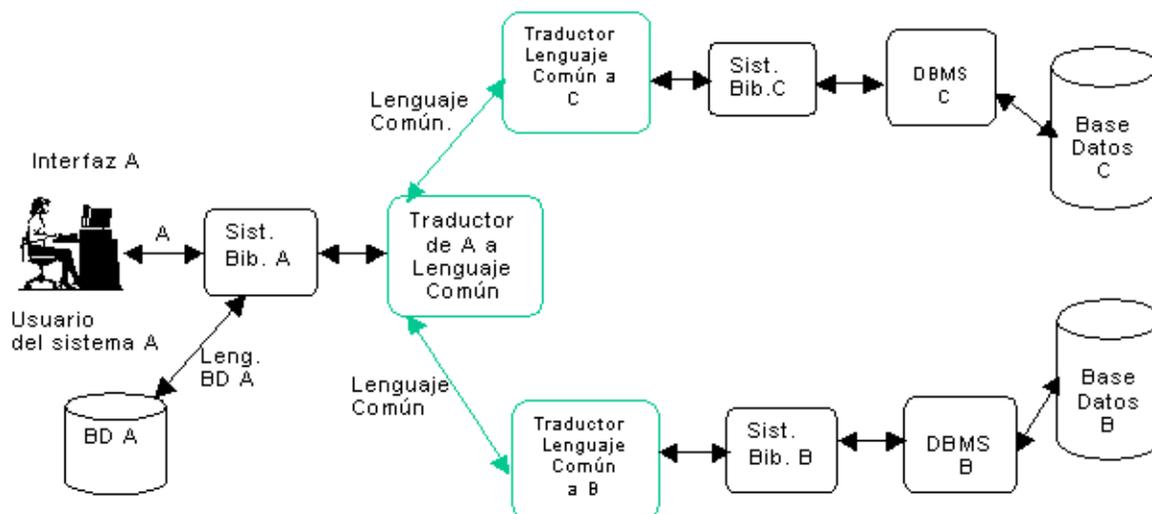


Figura 2.6: Traducción hecha por elementos traductores en cada sistema

Las características de este modelo son:

- El traductor de cada sistema tiene que conocer su propio lenguaje y el lenguaje común.
- Un nuevo sistema que quiera interactuar sólo tiene que incorporar el traductor que conozca el lenguaje común y su propio lenguaje.

2.5 Conclusión

El uso de un lenguaje común parece ser el medio más efectivo para interconectar sistemas bibliotecarios heterogéneos. Las ventajas pueden resumirse en los siguientes aspectos:

1. Transparencia de la heterogeneidad

Los sistemas que participan en la interacción no necesitan saber el lenguaje de los otros; es suficiente con conocer el lenguaje común.

Para los modelos de traducción local y remota, los sistemas deben conocer los lenguajes de los otros, lo que no hace transparente las diversas arquitecturas y lenguajes.

2. Conjunto mínimo de funciones de búsqueda y sintaxis de registros.

Cada sistema que participe en la interacción debe incluir sólo el conjunto de primitivas del lenguaje común.

En los modelos de traducción local y remota, cada sistema debe incluir el conjunto de funciones de búsqueda y sintaxis de registros de los sistemas con los que intercambia información.

3. Extensibilidad.

Un sistema nuevo sólo tiene que incluir su traducción al lenguaje común para poder interactuar con los demás. Esto disminuye el costo a la sola tarea de implantar el traductor.

En los modelos de traducción remota y local, cada sistema nuevo implica que cada uno de los que forman el conjunto tiene que conocer el nuevo lenguaje, lo que crea restricciones en la interacción.

4. Eficiencia con respecto al manejo de heterogeneidad.

Por el conjunto único de primitivas y sintaxis que los sistemas deben incluir, la solución es eficiente comparada con los modelos de traducción remota y local.

El lenguaje común elegido en este trabajo está especificado por el protocolo Z39.50, el cual será explicado en el capítulo siguiente.

3 BREVE REVISIÓN DEL PROTOCOLO Z39.50

Z39.50[13] es un estándar que hace transparente la interconexión entre diferentes sistemas de información bibliotecarios. Las mejores bibliotecas del mundo lo han incorporado y muchos fabricantes de sistemas bibliotecarios lo han adoptado. El protocolo Z39.50 es, pues, un estándar para interconectar sistemas heterogéneos.

Conforme han crecido las necesidades de intercambio de información entre los sistemas bibliotecarios, los catálogos en el Web han devenido insuficientes. El hecho de que cada biblioteca utilice su propio sistema, hace que existan muchas diferentes interfaces y sintaxis de búsqueda y recuperación; lo que genera problemas y frustraciones cuando se desea consultar varias bibliotecas.

Cada vez que se interactúa con un sistema bibliotecario, el usuario debe conocer el idioma de la interfaz, los comandos de búsqueda y recuperación. Esto crea restricciones en la intercomunicación bibliotecaria.

Como se discutió en el capítulo anterior, una solución es utilizar un lenguaje común para intercambiar información. Cada biblioteca implanta su propio sistema con su DBMS, su lenguaje y su hardware. Pero al intercambiar información lo hace en un lenguaje común.

Cada sistema manejador de información puede estar ejecutándose en cualquier plataforma e implementar su propio lenguaje de búsqueda y recuperación, pero cuando va a interactuar con otros sistemas para proveer información, lo hace en un lenguaje común previamente acordado. Todo sistema interrogador, puede tener su sintaxis de búsqueda y presentación de información, pero cuando lanza una búsqueda lo hace en este mismo lenguaje común. El lenguaje común se transforma, entonces, en el medio de

entendimiento de sistemas proveedores de información y de sistemas que la requieren.

Z39.50 surge como una alternativa porque es un protocolo que implanta un lenguaje común estándar. Las bibliotecas están cada vez más adoptando este estándar, porque facilita consultar diferentes catálogos de diferentes bibliotecas usando la misma interfaz.

3.1 El estándar Z39.50

En la literatura se usa el término "Z39.50 enabled" para indicar que un sistema posee la capacidad de interconexión a través del protocolo Z39.50. Buscar registros entre bibliotecas conectadas a través de Z39.50 es complejo a nivel de sistema, pero simple y eficiente a nivel de usuario final. El hecho de no tener que lidiar con las características particulares de cada sistema simplifica significativamente las búsquedas en muchas bibliotecas.

Lo anterior permite hoy en día disponer de una enorme capacidad de interconexión entre los sistemas bibliotecarios que hace eficiente los siguientes servicios:

- La catalogación remota con datos provenientes de diversas fuentes.
- El préstamo interbibliotecario.
- La creación de catálogos virtuales.
- La participación en catálogos virtuales de otras bibliotecas.
- La creación y distribución de bases de datos en CD, cuyo software de recuperación sigue el protocolo Z39.50.
- La divulgación de la información.
- El acceso a bases de datos no bibliográficas con capacidad Z39.50.
- La búsqueda en la Web.

Por la importancia actual y futura del Z39.50, un número creciente de bibliotecas, lideradas por la Biblioteca del Congreso de los Estados Unidos, está trabajando para mejorar la especificación del protocolo Z39.50 y los servicios derivados de su aplicación. De hecho, hoy en día, el protocolo se usa como un mecanismo genérico de búsqueda y recuperación de información y no sólo en sistemas bibliotecarios, sino también para datos no textuales como información gráfica-geográfica y para búsquedas con coincidencias de patrones. Este último uso es muy útil en búsquedas de secuencias de genes, estructuras químicas, huellas digitales, tendencias numéricas e imaginaria.

3.1.1 Reseña histórica y actualidad

El protocolo Z39.50 representa la culminación de dos décadas de debates acerca de cómo pueden ser modeladas, estandarizadas e implementadas las funciones de recuperación de información en un ambiente de sistemas distribuidos. Sus raíces se remontan a 1970 cuando un grupo de organizaciones que mantenían grandes bases de datos bibliográficas decidieron desarrollar una aplicación para hacer la catalogación de forma compartida, para lo que construyeron una base de datos bibliográfica a nivel nacional. Este programa fue llamado el “Linked Systems Project” [6] y agrupó a la Biblioteca del Congreso de E.E.U.U., el Centro de Bibliotecas Computarizadas en Línea (OCLC) y la Red de Información para Investigación en Bibliotecas.

Inicialmente, los participantes escribieron la especificación del protocolo y trabajaron en la implementación. Para 1980, el Proyecto se había enfocado casi exclusivamente en la implementación, y el trabajo sobre las especificaciones había sido movido hacia el desarrollo formal de estándares auspiciado por NISO (National Information Standar Organisation) [7].

El comité D de NISO fue establecido en 1979 y desarrolló el primer estándar que fue a votación en 1984 y no tuvo éxito. Después en 1987, el comité tuvo éxito con el ANSI Z39.50 [8]: definición del servicio de

recuperación de información y especificaciones del protocolo para aplicaciones de biblioteca.

Paralelamente al comité D de NISO, el Comité Técnico ISO 46 había estado trabajando sobre un protocolo llamado “Search and Retrieve (SR)” que era casi idéntico a Z39.50 excepto que omitía algunas funciones. Este trabajo fue estandarizado en 1991 como ISO 10162-10163 [9].

Durante 1989-1991 la Biblioteca del Congreso de E.E.U.U. (LC) fue nombrada la agencia de mantenimiento para Z39.50 por NISO. Su primer trabajo fue el de armonizar los dos estándares. El comité D de NISO fue disuelto y reemplazado funcionalmente por un grupo llamado “Z39.50 Implementors Group (ZIG) [10]”. Como resultado de estos eventos, en 1991 fue sometido a votación la segunda versión de Z39.50 que se llamó Z39.50-1992 [11], la cual, a diferencia de su predecesor, estaba fuertemente influenciada por comunidades de implementadores en varios ambientes. La segunda versión constituía un conjunto de especificaciones que incluían al estándar de ISO 10162/10163, y fue amplia y exitosamente implementada.

Entre 1992 y 1993 fue lanzado un programa llamado “Z39.50 Interoperability Testbed”[12] bajo el patrocinio de la Coalición para Información en Red en E.E.U.U. Su propósito fue facilitar el desarrollo de un gran número de implementaciones de Z39.50 que corrieran sobre TCP/IP, de manera tal que fueran accesibles a través de Internet. Este esfuerzo fue exitoso. Mientras esto ocurría, en 1991, el Z39.50 Implementors Group comenzó a trabajar en la versión 3 de Z39.50 que fue a votación en 1995 y se convirtió en el protocolo Z39.50-1995[13] o versión 3 que es, a la fecha, la actual.

En la actualidad, la Biblioteca del Congreso de los Estados Unidos mantiene un "Gateway" (<http://www.loc.gov/z3950>) que permite consultar una gran cantidad de bibliotecas a través de este protocolo. Esta lista puede explorarse para constatar la gran cantidad de bibliotecas que a nivel mundial están respaldando el estándar. Algunas de estas bibliotecas son:

- Universidad Complutense de Madrid.
- Instituto Tecnológico y de Estudios Superiores de Monterrey.
- Princeton University.
- University of Carnegie Mellon.
- Cornell University.
- Massachussets Institute of Technology.
- National Library of Canada.
- National Library of Australia.
- Oxford University.

3.1.2 Arquitectura del protocolo Z39.50

La arquitectura del protocolo Z39.50 es Cliente Servidor. Entre un cliente Z39.50 y un servidor Z39.50 se establece una sesión. En la figura 3.1, se ilustra esta arquitectura:

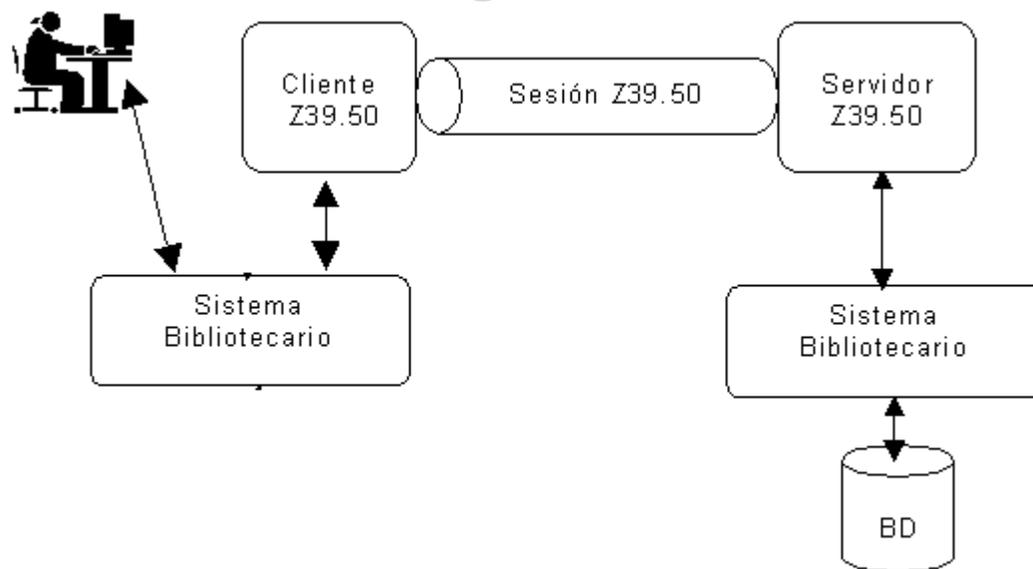


Figura 3.1: Arquitectura Cliente/Servidor del protocolo Z39.50

El usuario extremo, mediante su interfaz, puede hacer búsquedas a una base de datos remota de manera que, para él, existe una sesión entre su interfaz y la base de datos remota. Para que esto sea posible, el cliente Z39.50 y el servidor Z39.50 deben mantener una sesión a través de la cual fluye la información en el lenguaje del protocolo Z39.50.

El Sistema Bibliotecario recibe la búsqueda del servidor Z39.50 e interactúa con su base de datos para recuperar los registros que satisfacen la búsqueda; envía estos registros al servidor Z39.50, el cual vuelve a utilizar la sesión establecida con el cliente Z39.50 para enviarle los registros.

El protocolo Z39.50 sólo define las interacciones entre un cliente Z y un servidor Z. Una extensión natural de las aplicaciones que usen este protocolo es el manejo de múltiples sesiones entre un cliente Z y varios servidores Z, o que éstos atiendan a múltiples clientes Z concurrentemente. Este es uno de los aspectos más importantes del uso de Z39.50.

Visto del lado del cliente, una aplicación que aproveche las capacidades del protocolo Z39.50, debería ofrecer la habilidad de buscar en múltiples bases de datos usando siempre una misma interfaz de búsqueda. En otras palabras, brindaría la ilusión de un repositorio de información centralizado pero que realmente está distribuido. Del lado del servidor Z, la aplicación que use el protocolo Z39.50 debería ser un proveedor de la información almacenada en varias bases de datos y atender concurrentemente a varios clientes Z. En la Figura 3.2 se ilustra el uso del protocolo en un ambiente bibliotecario.

3.1.3 Servicios básicos del protocolo Z39.50

Los servicios básicos se denominan: Init, Search, Present y Close.

Init es el servicio mediante el cual el cliente Z39.50 y el servidor Z39.50 establecen una sesión. Negocian la sintaxis de la búsqueda, la sintaxis de los registros recuperados y los servicios que el servidor Z39.50 soporta.

Un servidor Z39.50 atiende a una o más bases de datos; cada una de las cuales tiene asociado un conjunto de puntos de accesos o índices que pueden ser usados para la búsqueda. El servicio Search permite al cliente transmitir una búsqueda al servidor. Una búsqueda produce un conjunto de registros llamado “conjunto resultante” que es mantenido en el servidor. El estándar no dice nada acerca de la conformación del conjunto resultante; éste puede conformarse de los registros en sí o de apuntadores a ellos.

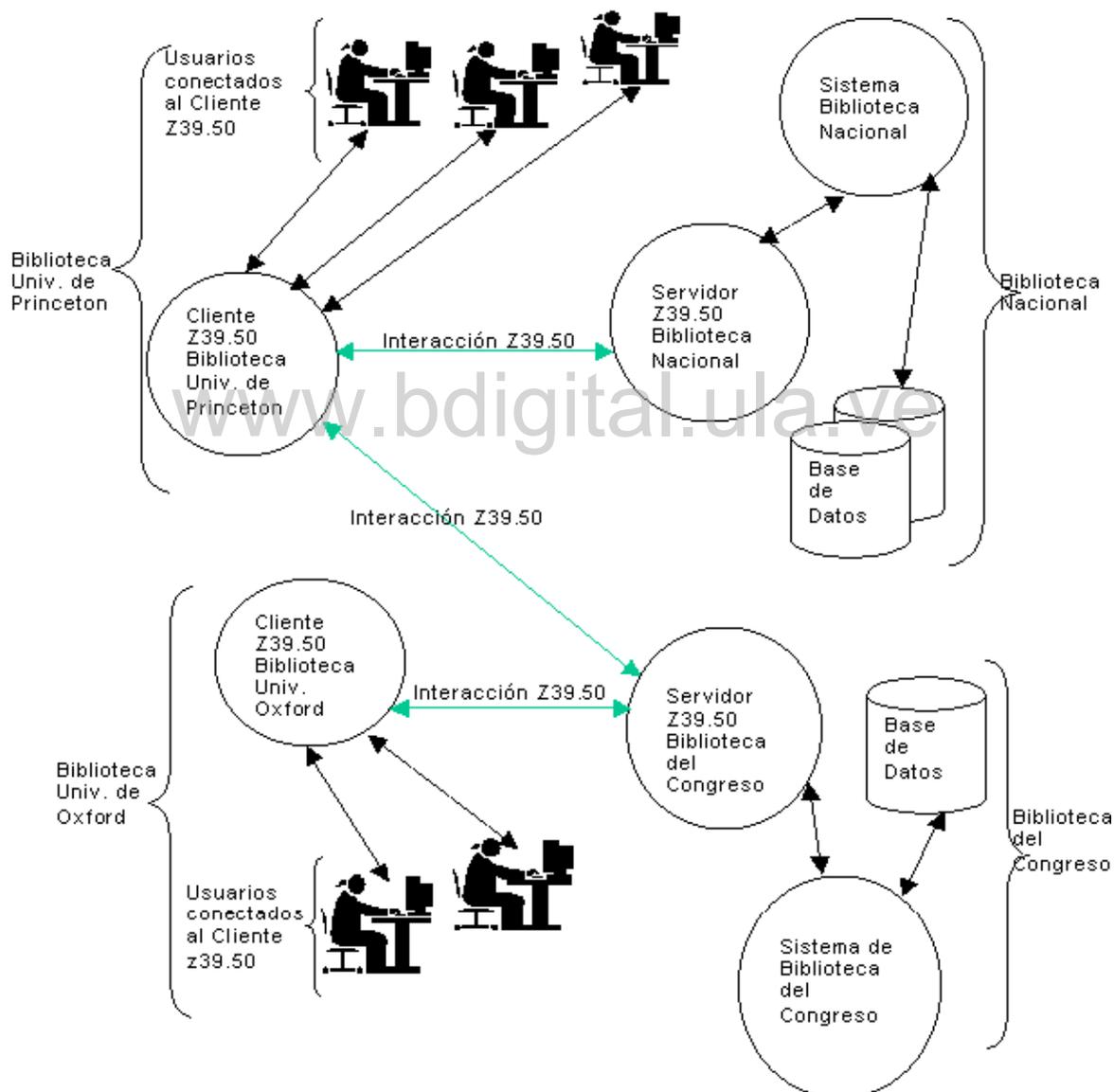


Figura 3.2: Ejemplo de uso del protocolo Z39.50 en bibliotecas

El cliente Z puede enviar una búsqueda indicando una base de datos. La sintaxis de la búsqueda es la de una sentencia con términos y operadores booleanos. Adicionalmente, se envían parámetros que indican si se quiere hacer sólo una búsqueda sin recuperación de registros o una búsqueda con recuperación de un subconjunto de registros. El servidor Z responde con una cuenta de registros identificados y posiblemente algunos o todos los registros. Con la información recibida, el cliente Z decide si desea recuperar el resto de registros y asume que los registros seleccionados forman un conjunto resultante y ordenado que reside en el servidor Z. Dentro del servidor, los registros pueden ser referenciados por su posición dentro del conjunto y recuperados en varias peticiones. El conjunto resultante se mantiene en el servidor Z hasta que el cliente decida iniciar una nueva búsqueda o cerrar la sesión.

Los registros en el conjunto resultante pueden ser sucesivamente recuperados por el cliente mediante el servicio Present. Se indica los registros específicos del conjunto resultante que van a ser recuperados. Hay facilidades para manejar espacios más grandes en el servidor para registros muy grandes y para su transferencia. También se proveen facilidades para gestionar búsquedas muy intensas: el servidor transmite reportes sobre el estado de una búsqueda para que el cliente pueda cancelar o proseguir.

La sesión es terminada mediante el servicio Close. Este servicio es invocado por el cliente Z39.50 o por el servidor Z39.50.

3.1.4 Servicios secundarios del protocolo Z39.50

Z39.50 contiene facilidades para el ordenamiento y eliminación de los conjuntos resultantes. También existe un conjunto de servicios llamados “servicios extendidos”, para que un cliente Z invoque remotamente un procedimiento en el servidor.

El protocolo también define lo siguiente:

1. Un lenguaje para hacer las peticiones que incluye una especificación para los atributos que califican a los términos de búsqueda.
2. Varias sintaxis de registros que son usadas para dar formato a los registros transferidos del servidor al cliente: MARC, SUTRS, GRS-1 entre otras.
3. Una facilidad llamada EXPLAIN que permite a los cliente la obtención de un rango amplio de información del servidor tales como los índices de cada base de datos.

3.1.5 Gramática de servicios y registros

Los servicios Search y Present son invocados iterativamente durante una sesión entre cliente Z39.50 y servidor Z39.50. Usan operadores y operandos regidos por gramáticas que validan los constructos de estos servicios.

La gramática del Search usa una notación sufija. Los operandos son los términos que se quieren buscar. Los atributos que califican a estos operandos pertenecen a un registro estándar llamado Bib-1[14]. La gramática del Present usa referencias a los registros por su posición en el conjunto resultante.

En el medio bibliotecario, la gramática más popular para expresar los registros recuperados por el servidor Z39.50 está regida por el formato MARC. Cuando el cliente Z39.50 pide un Search con recuperación de registros o un Present, los registros que recibe están expresados en formato MARC.

3.2 El formato MARC

El protocolo Z39.50 utiliza el formato MARC para estructurar los registros recuperados, los cuales son transmitidos por el servidor Z39.50 hacia el cliente Z39.50. El cliente que los recibe puede extraer o componer datos para construir su salida. La ventaja del formato MARC es que provee un estándar para denominar cada campo y su contenido. Por ejemplo, el campo

autor siempre es el número 100, el título siempre es el 245, la materia siempre es el 650, etcétera.

3.2.1 Estructura del registro MARC

Un registro MARC es una cadena plana de caracteres ASCII. Cada registro está dividido lógicamente en campos, los cuales están divididos en subcampos. Además hay unos indicadores por cada campo para dar información adicional. Hay campos para el autor, el título, la editorial, etc. Los nombres de los campos se representan con un identificador de 3 dígitos que antecede al contenido del campo. Un sistema que interprete el formato MARC debe dar un nombre a cada identificador de campo. Así, cada identificador tiene una semántica igual para todo catálogo y una etiqueta que depende de cada sistema.

El Servicio de Distribución de la Catalogación de la Biblioteca del Congreso distribuye una lista detallada de todos los identificadores de campo en la publicación de dos volúmenes titulada “MARC 21 Format for Bibliographic Data” [15] y en otra publicación resumida titulada “MARC 21 Concise Formats” [16].

Los indicadores son dos caracteres que se ubican a continuación del identificador de campo. Se pueden usar uno, otro, o ambos para indicar características adicionales del campo. En algunos campos, por ejemplo, el 020 y el 300, no se usa ninguno. Cuando un indicador no se usa, éste es sustituido por un numeral (#).

Un carácter indicador es un número del 0 al 9. Aun cuando dos indicadores juntos pueden parecer un número de dos dígitos, ellos son realmente dos indicadores de un dígito cada uno.

La mayoría de los campos contienen varias porciones relacionadas de datos. Cada porción se llama subcampo y cada uno de estos es precedido por un código. Los campos desde el 001 al 009 no tienen subcampos. Por ejemplo, el campo dedicado a la descripción física de un libro, identificado por el 300, incluye un subcampo para el número de páginas, uno para otros detalles físicos como la información de ilustración y uno para la dimensión en centímetros.

El código que identifica a un subcampo está conformado por una letra minúscula precedido por un delimitador, que es un carácter usado para separar subcampos y puede ser una arroba (@), un dólar (\$) ú otro. En la figura 3.3 se ilustra la estructura de un registro MARC.

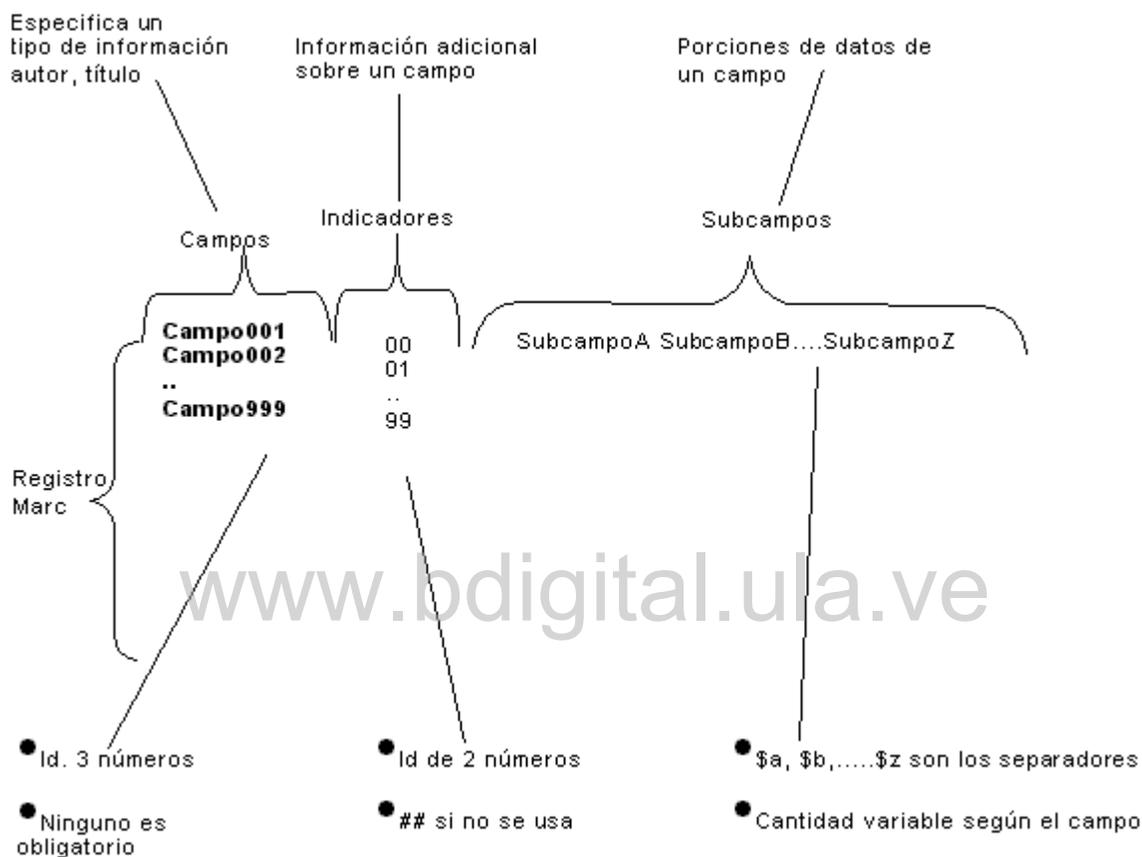


Figura 3.3: Estructura de un registro MARC

3.2.1.1 Ejemplo de Registro MARC

Este registro fue tomado del servidor Z de la Biblioteca del Congreso de E.E.U.U. Los campos que están resaltados se explican a continuación:

```

001 1192676
005 19971022104611.2
008 971021s1996 ve b 001 0bspa c
035 $9 (DLC) 97207395

```

906 \$a 7 \$b cbc \$c pccunad \$d u \$e ncip \$f 19 \$g y-gencatlg

955 \$a CU 10-21-97

010 \$a 97207395

020 \$z 980638667

040 \$a CU **\$c** CU

042 \$a pcc

043 \$a s-ve---

050 14 \$a F2323.M6 \$b P65 1996

100 2 \$a Polanco Alcántara, Tomás, **\$d** 1927-

245 10 \$a Francisco de Miranda : **\$b** bosquejo de una biografía : Don Juan o Don Quijote? / **\$c** Tomás Polanco Alcántara.

250 \$a 1. ed. venezolana.

260 \$a Caracas : \$b EG, \$c 1996.

300 \$a 779 p. ; \$c 23 cm.

504 \$a Includes bibliographical references.

600 10 \$a Miranda, Francisco de, \$d 1750-1816.

650 0 \$a Revolutionaries \$z Venezuela \$x Biography.

651 0 \$a Venezuela \$x History \$y War of Independence, 1810-1823.

001: es un número de control, normalmente el identificador unívoco del registro en el catálogo de la base de información.

005: fecha y hora de la última transacción

010 \$a: es el número de control de la Biblioteca del Congreso (LCCN), \$a es el número LCCN

020 \$z: es el número de libro en el estándar internacional (ISBN), \$z significa que el número es inválido.

040 \$a \$c: es la fuente de catalogación, \$a es la agencia original de catalogación, \$c es la agencia de transcripción.

100 2 \$a \$d: es la entrada principal al registro, 2 es un indicador obsoleto desde 1990, \$a es el nombre personal y \$d es la fecha de nacimiento

245 10 \$a \$b \$c: título del material, 1 es un indicador de que el título es un punto de entrada al registro, 0 es un indicador de que no hay caracteres de relleno, es decir que el título debe clasificarse por su primera letra, \$a es el propio título, \$b es un subtítulo y \$c es una oración que puede indicar un responsable.

3.2.1.2 Transmisión del registro MARC

El servidor Z39.50 transmite los registros MARC como una cadena plana de caracteres ASCII. El cliente Z39.50 tiene que interpretar esta cadena para obtener cada uno de los registros con sus datos. Cada registro MARC tiene tres partes: un Líder, un Directorio y los Datos. El Líder dice la longitud del registro en número de caracteres. El Directorio es un índice en el que hay una entrada por cada campo. Los Datos contienen la información de cada campo.

La estructura del registro MARC es como sigue:

- El Líder está compuesto por los 24 caracteres iniciales. Cada posición, desde la 0 hasta la 23 tiene un significado.
- El Directorio indica cuales son los identificadores de campo y sus posiciones dentro del registro. La posición se obtiene contando los caracteres que existen desde el primer carácter donde empieza la información bibliográfica hasta el inicio del identificador de campo.
- Después del Directorio se encuentran los Datos que son la información propiamente dicha del registro MARC. En la figura 3.4 se ilustra esta estructura.

Para resumir:

- Hay una longitud máxima de registro.
- Hay una longitud máxima de campo.
- El fin de cada registro está determinado por su longitud, la cual se indica en las primeras cinco posiciones del líder.
- El tipo de subcampo depende de cada campo.

- Los indicadores que se usan dependen de cada campo
- Ningún campo ni subcampo es obligatorio.
- El software que interpreta este formato maneja criterios por cada campo sobre los indicadores y subcampos que puede encontrar.
- La razón de ser de una estructura que no es fija, y lleva en si misma los identificadores de los campos, es la necesidad de dar cabida a la gran diversidad de información de catalogación que existe en las bibliotecas del mundo.

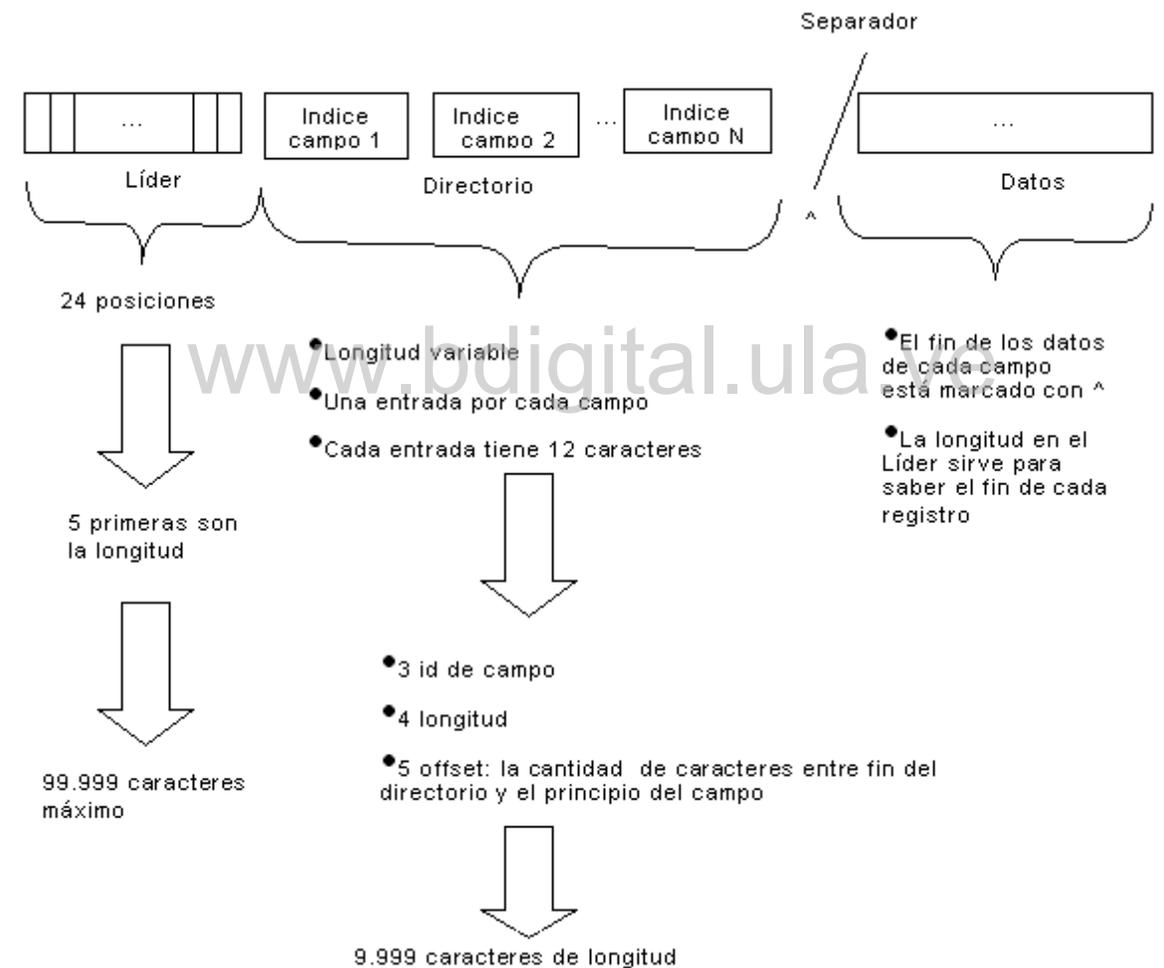


Figura 3.4: Transmisión de registros MARC

3.3 Conclusión

El protocolo Z39.50 es un estándar que permite la interoperabilidad de sistemas heterogéneos. Especifica una serie de servicios y gramáticas para permitir la búsqueda y recuperación de registros.

Entre el cliente y el servidor Z39.50 se mantiene una sesión a través de la cual fluye la información.

Un sistema bibliotecario que requiera intercambiar información con otros sistemas debe tener un cliente Z39.50 para buscar información, y un servidor Z39.50 para proveer información.

Es deseable extender el uso del protocolo para permitir que el cliente Z haga interrogaciones concurrentes a varios servidores Z, y así mismo, que el servidor Z acepte peticiones concurrentes de varios clientes Z para proveer información.

www.bdigital.ula.ve

4 UN MODELO DE INTEGRACIÓN BIBLIOTECARIA CON Z39.50

En los capítulos anteriores hemos visto que las bibliotecas se automatizan y tienen sistemas de recuperación de información. Eventualmente, los sistemas difieren en elementos tales como el hardware, las interfaces, los manejadores de bases de datos, los lenguajes de recuperación de información, las sintaxis de registros, etcétera.

Las bibliotecas prestan servicios a usuarios y realizan procesos internos, muchos de los cuales necesitan de cooperación entre bibliotecas, proveedores de información, vendedores de software, etcétera. Para poder interactuar, se necesita intercambiar información. Si cada sistema es diferente, el intercambio es restringido porque las estructuras de los datos de cada sistema son diferentes y las interfaces de usuario son distintas.

Una forma de disminuir el grado de limitación en la interacción entre sistemas bibliotecarios distintos es lograr que mantengan sin cambios sus interfaces. Esto es deseable porque, tanto para el usuario final como para las bases de datos, es transparente la interacción entre sistemas heterogéneos.

Hemos visto algunos esquemas bajo los cuales se puede llevar a cabo la interoperabilidad entre sistemas heterogéneos sin afectar sus interfaces. El esquema que más ventajas brinda es el intercambio de información en un lenguaje común.

Grandes bibliotecas, como la Biblioteca del Congreso de E.E.U.U., organizaciones como NISO e ISO, han desarrollado estándares de integración de bibliotecas. Uno de los primeros desarrollados es el formato MARC descrito

en el capítulo 3. Del mismo modo, otro estándar que ha tenido gran auge es el protocolo Z39.50, también explicado en el capítulo 3.

Con el fin de facilitar la integración entre sistemas distintos, sin afectar sus interfaces, este trabajo propone un modelo de integración de bibliotecas mediante el protocolo Z39.50. Este modelo se denomina “MIZ” y de ahora en adelante nos referiremos a él usando este nombre.

En el contexto del MIZ, una aplicación es un sistema de recuperación de información que tiene los siguientes elementos:

- Una interfaz humano computadora que permite hacer búsquedas de registros y despliegue de resultados.
- Una interfaz de base de datos que permite manejar las bases de datos.
- Una o más bases de datos.

4.1 Consideraciones de diseño del MIZ

La esencia del MIZ es proveer un enfoque para que las aplicaciones bibliotecarias desarrollen la capacidad de interactuar sin que sus usuarios y sus bases de datos lidien con la heterogeneidad de los sistemas con los que interactúan. En la figura 4.1 se ilustran dos sistemas: uno que busca información a través del MIZ y otro que recupera información a través del MIZ.

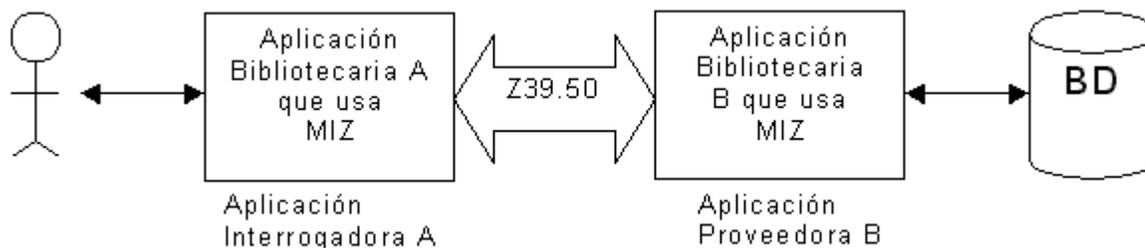


Figura 4.1: Interacción de las aplicaciones bibliotecarias A y B a través del MIZ.

La comunicación entre estas dos aplicaciones es remota y se hace mediante el protocolo Z39.50. La aplicación A usa el MIZ para interrogar y la

aplicación B usa el MIZ para proveer. En adelante las llamaremos MIZ Interrogador y MIZ Proveedor, respectivamante.

El requisito base de diseño del MIZ es la transparencia de la complejidad del intercambio de información para:

- El usuario que pide información.
- Las bases de datos que proveen información.

Este requisito implica que el modelo tiene que conservar la interfaz humano computadora y la interfaz de las bases de datos de la aplicación.

El modelo incluye al protocolo Z39.50 para intercambiar información, pues hemos mostrado que es el estándar de interoperabilidad entre sistemas bibliotecarios.

En la figura 4.2 se ilustran los requerimientos de diseño mencionados hasta ahora:

- Conservar las interfaces.
- Usar Z39.50.

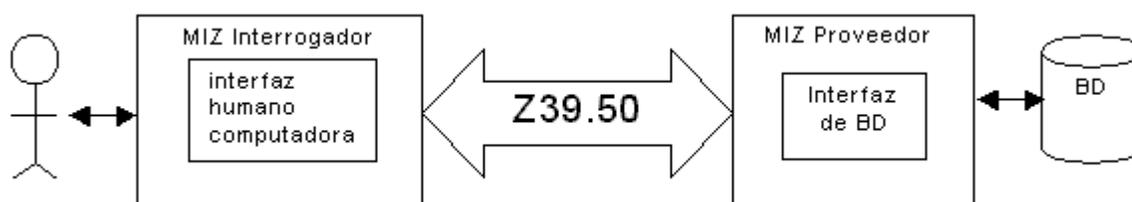


Figura 4.2: Uso del protocolo Z39.50 y de las interfaces humano computadora y de base de datos en el MIZ.

El MIZ se fundamenta en dos módulos principales denominados MIZ Interrogador y MIZ Proveedor respectivamente. Ambos módulos se comunican entre sí a través de Z39.50. El MIZ Interrogador es responsable de traducir los comandos de la interfaz de usuario del sistema bibliotecario a una sesión Z39.50 con un extremo remoto. El MIZ Proveedor es responsable de atender solicitudes remotas, emitidas por un MIZ Interrogador o un cliente Z39.50, a comandos de la base de datos del sistema local.

La inclusión del protocolo Z39.50 significa que el modelo debe contemplar submódulos para el cliente Z39.50 y el servidor Z39.50, tal como se ve en la figura 4.3:

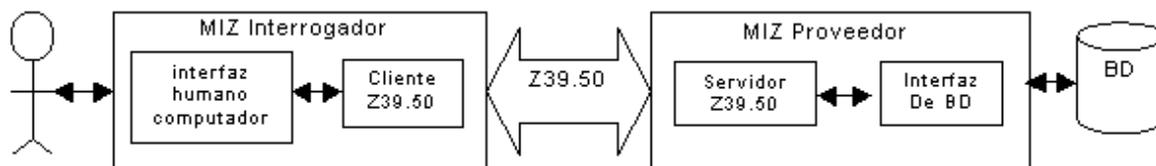


Figura 4.3: Cliente Z39.50 y Servidor Z39.50 dentro del MIZ.

Es importante que cada elemento que interactúe efectúe un grupo único de funciones, lo que significa que:

- Debe implementar el mínimo número de primitivas.
- Evitar redundancia.
- No debe implementar funciones pertinentes a otros elementos.

www.bdigital.ula.ve

Hasta el presente, según la figura 4.3, los submódulos identificados son:

- **Interfaz humano computadora:** interacción con el usuario.
- **Cliente Z39.50:** comunicación con los servidores Z39.50.
- **Servidor Z39.50:** comunicación con los clientes Z39.50.
- **Interfaz de Base de Datos:** interactuar con las bases de datos.

La interfaz humano computadora de la aplicación interrogadora tiene su propio lenguaje de búsqueda de registros. El cliente Z39.50 maneja el lenguaje del protocolo Z39.50. Esto sugiere una traducción entre la interfaz y el submódulo Z. Igualmente, en la aplicación proveedora, el servidor Z39.50 y la interfaz de la base de datos manejan el lenguaje del protocolo y el lenguaje de la base de datos respectivamente: luego, de este lado también se necesita una traducción. En la figura 4.4 se ven los puntos de traducción necesarios.

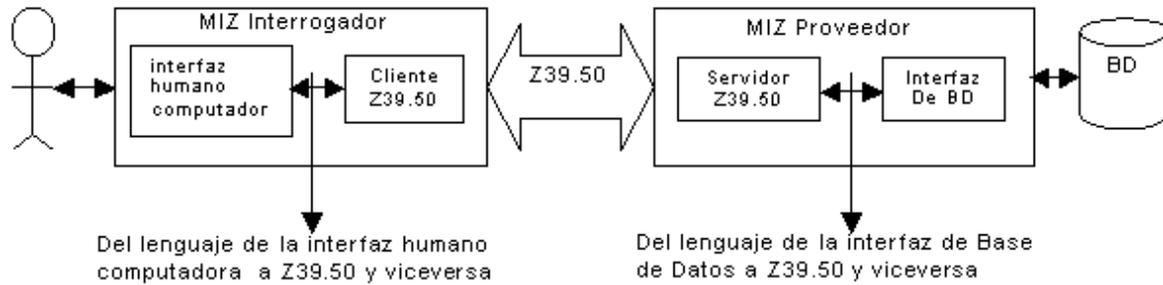


Figura 4.4: Traductores en el MIZ.

Las traducciones pueden hacerlas cualesquiera de los elementos que entran en la interacción; empero se estarían incumpliendo algunos de los requerimientos de diseño formulados hasta ahora. Si las interfaces son traductoras, entonces se estaría faltando al principio de no afectarlas; si el cliente Z39.50 y el servidor Z39.50 son los traductores, entonces estos elementos no estarían realizando un grupo único de funciones.

Estas consideraciones llevan a un nivel más de refinamiento en el que se introducen elementos traductores de lado y lado, tal como se ve en la figura 4.5.

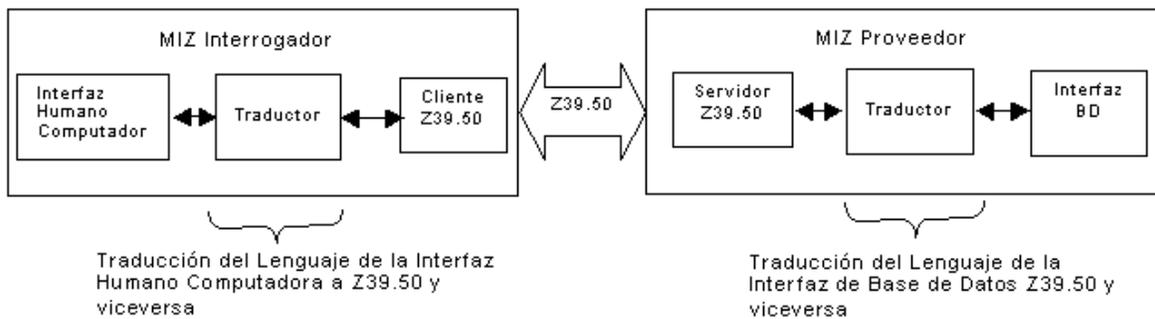


Figura 4.5: MIZ Interrogador y MIZ Proveedor con traductores

4.2 Arquitectura final del MIZ

La arquitectura final del MIZ tiene dos lados, uno que interroga y otro que provee. Cada uno de ellos tiene tres submódulos tal como se ve en la figura 4.6.

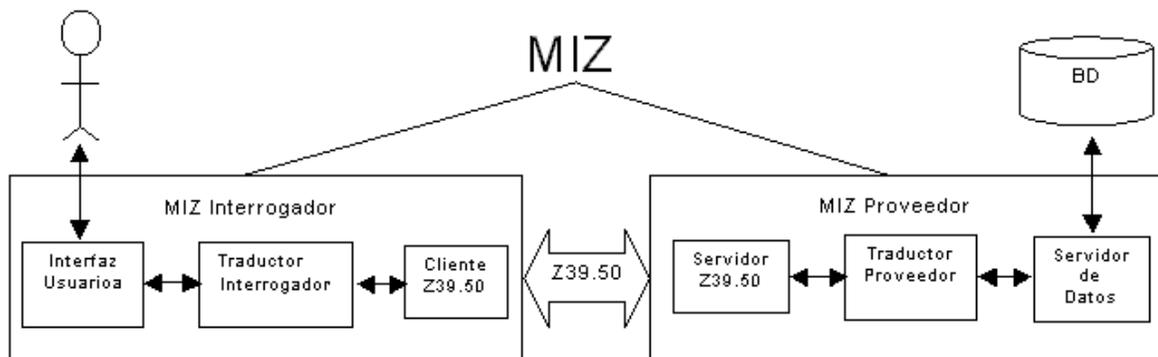


Figura 4.6: Arquitectura final del MIZ.

Cada submódulo del modelo de la figura 4.6 tiene funciones y estructuras bien específicas:

1. Interfaz Usuario:

- Pertenece a la aplicación.
- Comprende una parte vista por el usuario y una parte vista por la aplicación.
- Su función es interactuar con el usuario.

2. Traductor Interrogador:

- Debe ser implementado.
- Comprende una parte vista por la Interfaz de Usuario y una parte vista por el Cliente Z39.50.
- Su función es mediar entre la Interfaz Usuario y Cliente Z39.50.

3. Cliente Z39.50:

- Comprende una implementación del protocolo Z39.50 como Cliente, y una parte que debe ser implementada para manejar las peticiones a varios servidores Z39.50.
- Su función es interactuar con los Servidores Z39.50.

4. Servidor de Datos:

- Pertenece a la aplicación y es la interfaz con las bases de datos.
- Comprende una parte vista por la aplicación bibliotecaria y una parte vista por las base de datos.
- Su función es interactuar con la Base de Datos.

5. Traductor Proveedor:

- Debe ser implementado.
- Comprende una parte vista por el servidor Z39.50 y una parte vista por el Servidor de Datos.
- Su función es mediar entre el Servidor Z39.50 y el Servidor de Datos.

6. Servidor Z39.50:

- Es una implementación del protocolo Z39.50 como servidor y una parte que debe ser implementada para atender peticiones de varios clientes Z39.50.
- Su función es interactuar con los clientes Z39.50.

La estructura de los submódulos del MIZ se ve en la figura 4.7.

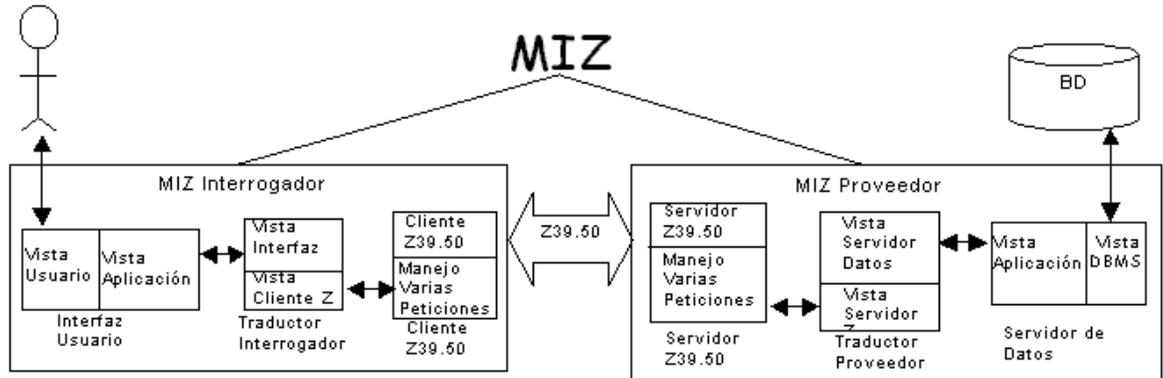


Figura 4.7: Vista de cada componente del MIZ

Los seis elementos componentes del modelo necesitan de lenguajes para poder interactuar entre ellos, tal como se ve en la figura 4.8:

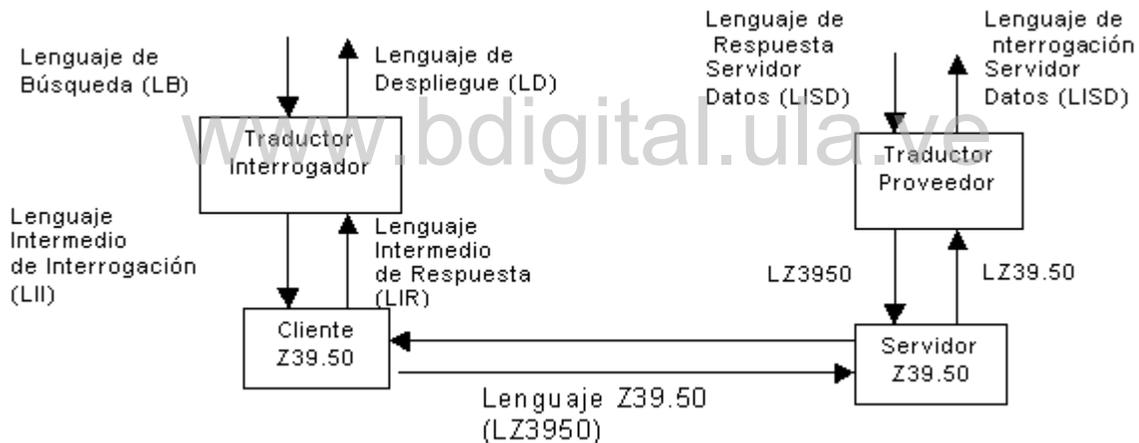


Figura 4.8: Lenguajes de interacción entre los componentes del MIZ

Lenguaje de Búsqueda: es el lenguaje que usa la Interfaz de Usuario para hacer una búsqueda en su aplicación.

Lenguaje de Despliegue: es el lenguaje que utiliza la Interfaz de Usuario para expresar los resultados de una consulta en su aplicación.

Lenguaje Intermedio de Interrogación: es provisto por el MIZ. Contiene al lenguaje de búsqueda del protocolo Z39.50 y, además, constructos útiles para que el Cliente Z39.50 maneje peticiones a varios servidores Z39.50.

Lenguaje Intermedio de Respuesta: es provisto por el MIZ. Contiene los constructos para identificar los registros de respuesta y el servidor del que provienen, así como los datos del registro en si mismo.

Lenguaje Z39.50: es el lenguaje que provee el protocolo para expresar búsquedas y registros (MARC, texto).

Lenguaje de Interrogación del Servidor de Datos: es el lenguaje que maneja la aplicación para recuperar datos de sus bases de datos.

Lenguaje de Respuesta del Servidor de Datos: es el lenguaje que usa la aplicación para expresar los registros recuperados de sus bases de datos.

www.bdigital.ula.ve

4.2.1 Modelado de objetos del MIZ en UML

Clases Componentes del MIZ

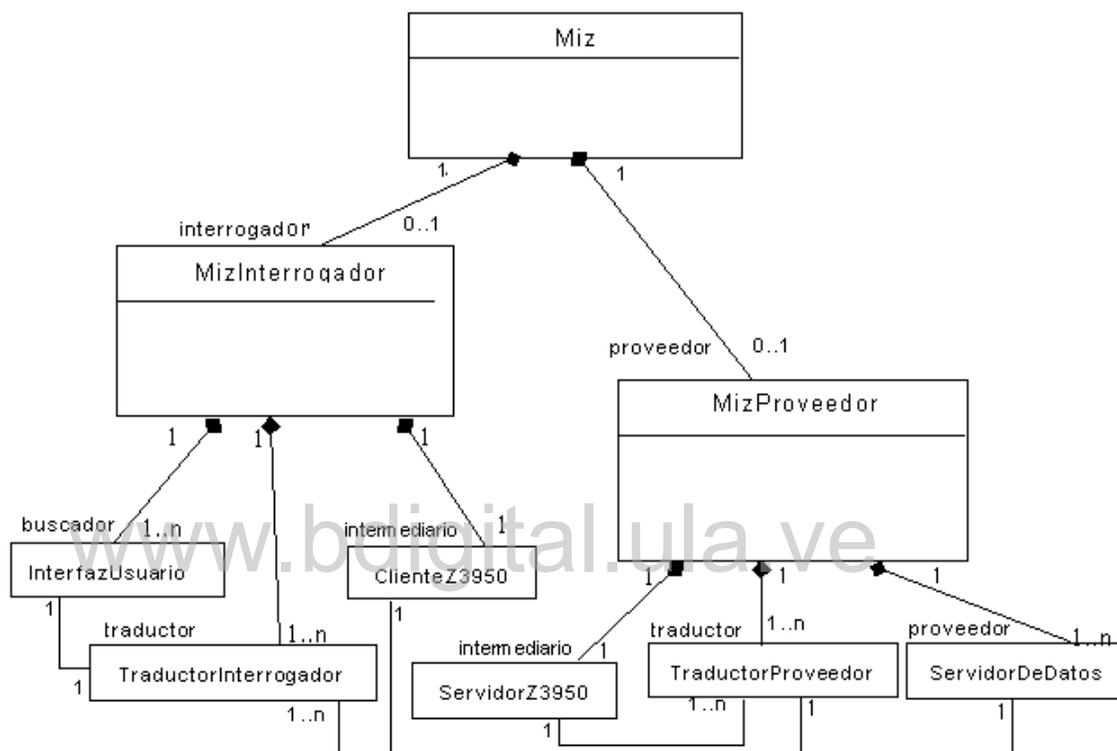


Figura 4.9: Clases componentes del MIZ

El MIZ está compuesto de uno o ningún interrogador y de uno o ningún proveedor, de tal manera que una aplicación puede ser interrogadora, proveedora o ambas.

El MIZ Interrogador puede estar compuesto de interfaces de usuarios distintas que cumplen el rol de buscadoras. Para cada interfaz distinta existe un traductor interrogador que traduce del lenguaje de la interfaz de usuario a Z39.50 y viceversa. Existe un solo cliente Z39.50 que cumple el rol de intermediario Z.

El MIZ Proveedor puede estar compuesto de servidores distintos de datos que cumplen el rol de proveedores. Para cada servidor de datos distinto existe un traductor proveedor que traduce del lenguaje del servidor de datos a Z39.50 y viceversa. Existe un solo servidor Z39.50 que cumple el rol de intermediario Z.

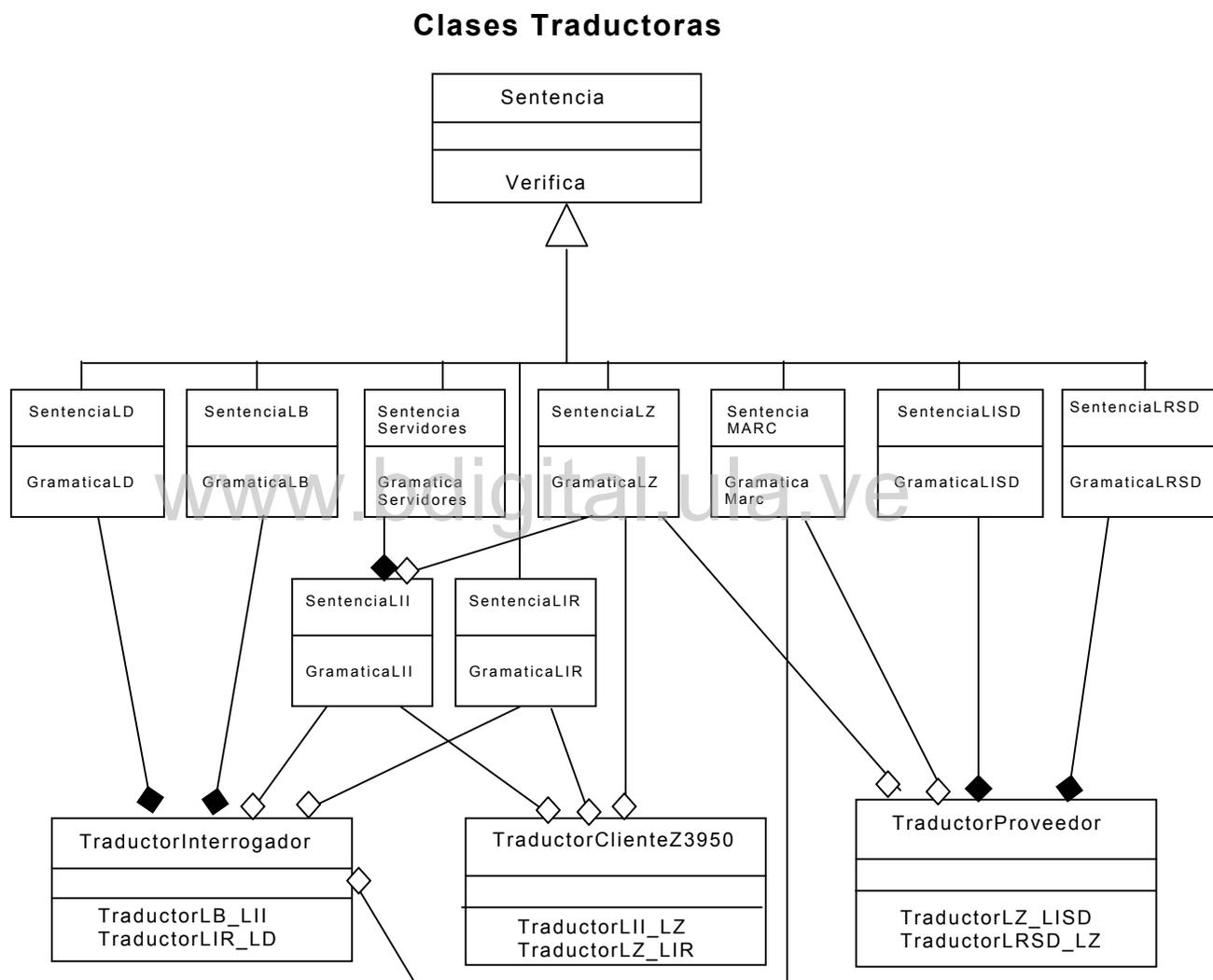


Figura 4.10: Clases traductoras del MIZ

Las tres clases traductoras están compuestas por los analizadores de sentencias de los lenguajes con los que interactúan. La clase SentenciaServidores expresa los nombres de múltiples servidores Z39.50 y sus bases de datos. La clase SentenciaMARC es un constructor e interpretador del

formato MARC. La clase TraductorClienteZ3950 forma parte de la clase ClienteZ39.50 que se ve en la figura 4.11.

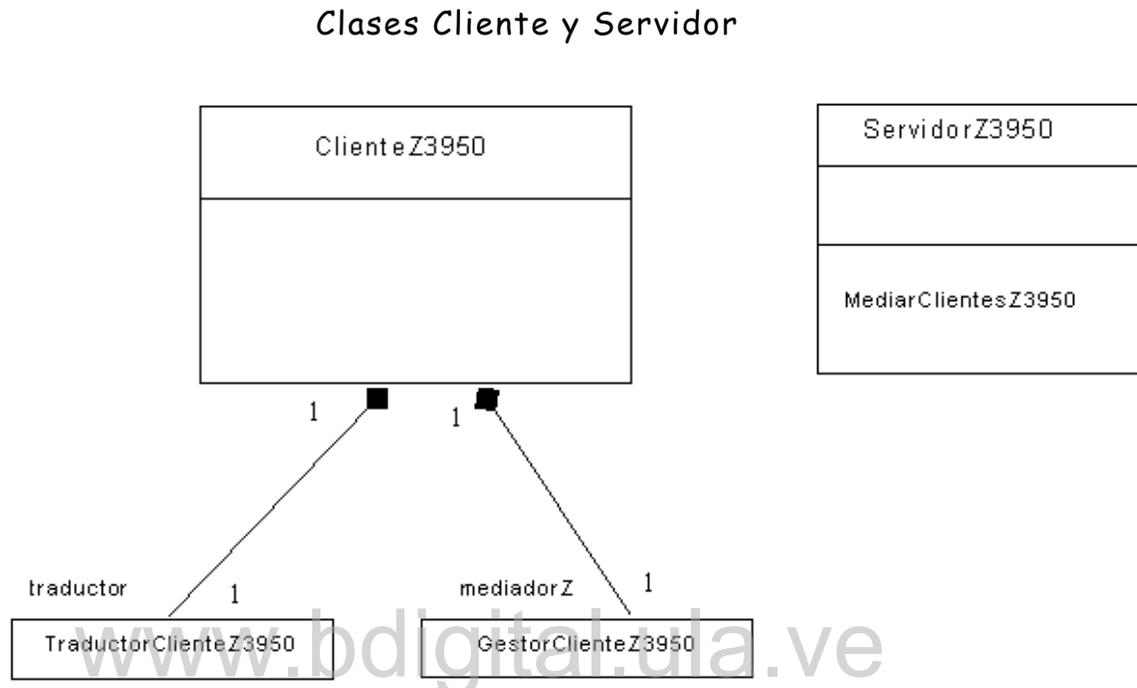


Figura 4.11: Clases cliente y servidor Z39.50

La clase **ClienteZ3950** está compuesta de un traductor y un mediador con los servidores Z39.50. Esta clase debe traducir porque maneja peticiones concurrentes a varios servidores Z39.50. Desglosa una sola instrucción proveniente de los elementos anteriores en varias instrucciones en Z39.50. A cada servidor Z39.50 requerido le envía una instrucción de búsqueda. Cuando recibe las respuestas de estos servidores, construye una sola sentencia con los resultados y la envía al traductor interrogador.

La clase **ServidorZ3950** recibe una instrucción de búsqueda de cada cliente Z39.50 y envía de respuesta una sola sentencia de resultados por cada cliente.

4.2.2 Lenguajes del MIZ

El MIZ provee las gramáticas para dos lenguajes que son útiles porque facilitan el manejo de peticiones a múltiples servidores y sus respuestas.

El Lenguaje Intermedio de Interrogación (LII) es usado por el traductor interrogador porque recibe una sentencia de búsqueda de la interfaz usuario. La sentencia de búsqueda, indica los términos a ser buscados, los servidores Z39.50 y las bases de datos a las que el usuario quiere consultar. El traductor Interrogador usa la gramática de LII para transformar la sentencia en una instrucción que tiene una porción en lenguaje de búsqueda del protocolo Z39.50, y otra que expresa los servidores y sus bases de datos. En la figura 4.12 se ven las partes constituyentes de una instrucción en LII.

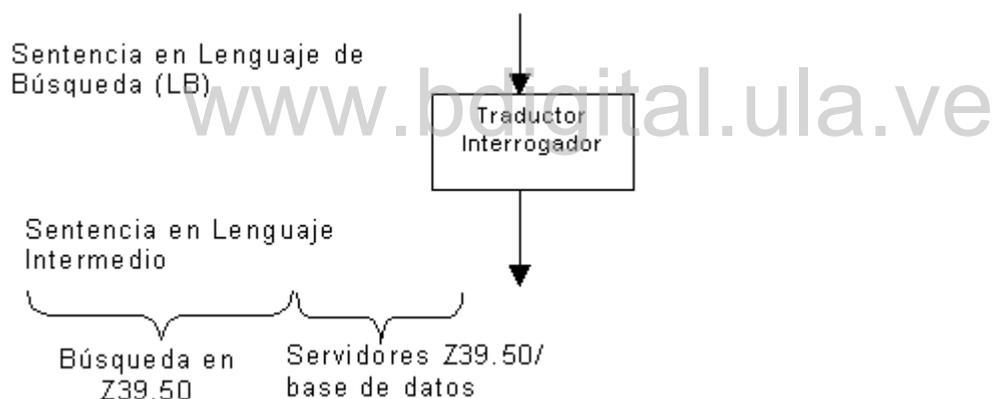


Figura 4.12: Transformación de una instrucción en LB a LII

La gramática de LII es:

```
<lenguaje intermedio de interrogación> ::= <sintaxis de búsqueda de Z3950>#  
<condiciones> <servidores>  
<condiciones> ::= <sesión># <primeros># <siguientes># <anterior># <desde>  
#<continua>  
<servidores> ::= #<dirección ip>:<puerto>/<basedatos> <servidores> | <vacío>  
<sesión> ::= <cadena>  
<primeros> ::= <dígito><dígito><dígito>
```

```

<siguientes> ::= <primeros>
<anterior> ::= s | n
<desde> ::= <entero>
<continua> ::= s | n
<dirección ip> ::= <cadena>
<puerto> ::= <entero>
<basedatos> ::= <cadena>
<vacío> ::= '\0'
<dígito> ::=
<entero> ::= <dígito>|<dígito><entero>
<cadena> ::= {caracteres ASCII}

```

En el Anexo I, sección A se da información adicional sobre esta gramática.

El Lenguaje Intermedio de Respuesta (LIR) es usado por el Cliente Z39.50 porque recibe una respuesta de cada servidor Z39.50 al que interrogó. Estas respuestas llegan de manera asíncrona y el cliente Z39.50 debe armar una oración que contenga los registros numerados, el origen de cada uno de ellos, sus datos, el total encontrado por cada servidor y mensajes pertinentes a la comunicación. Para elaborar esta respuesta usa la gramática de LIR, tal como se ven en la figura 4.13.

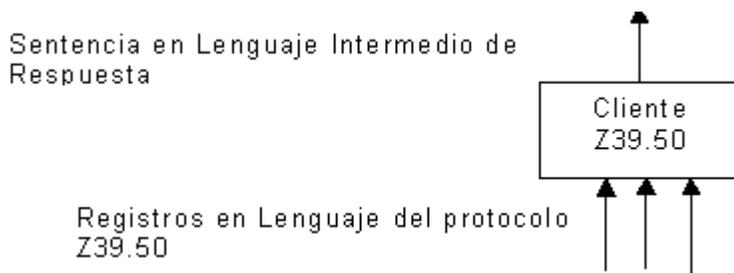


Figura 4.13: Transformación de registros en LZ39.50 a LIR

La gramática de LIR es:

```
<lenguaje intermedio de respuesta> ::= TOKENINICIO <contenido> TOKENFIN
<contenido> ::= <tokenexiste> <resto> <contenido> |
               <tokenmensajecon> <contenido> |
               <vacío>
<resto> ::= <tokenregistro> <tokenvalor> <resto> |
            <tokenmensajeres> <vacío> |
            <vacío>
<tokenvalor> ::= TOKENVALOR <cadena> <vacío>
<tokenmensajecon> ::= TOKENMENSAJECON <cadena> <vacío>
<tokenmensajeres> ::= TOKENMENSAJERES <cadena> <vacío>
<tokenregistro> ::= TOKENREGISTRO <entero> <vacío>
<tokenexiste> ::= TOKENEXISTE <entero> registros en <cadena> <vacío>
<vacío> ::= '\0';
<dígito> ::=
<entero> ::= <dígito>|<dígito><entero>
<cadena> ::= {caracteres ASCII}
```

En el Anexo A, sección B se da información adicional sobre esta gramática.

4.3 Conclusión

El MIZ es un modelo que facilita la integración de sistemas heterogéneos de recuperación de información.

Su esencia es:

- La preservación de la interfaz humano computadora y de la interfaz de las bases de datos de la aplicación, para lo cual modela:
 - Traductores en el lado interrogador: uno para cada interfaz usuario distinta y uno para el cliente Z39.50.
 - Traductores en el lado proveedor: uno para cada interfaz de base de datos distinta.
- El uso del protocolo Z39.50 para intercambio de información entre cliente y servidor.
- La extensión del protocolo Z39.50 para permitir que un cliente intercambie información con varios servidores, para lo cual provee gramáticas para:
 - El Lenguaje Intermedio de Interrogación.
 - El Lenguaje Intermedio de Respuesta.

Las peticiones en lenguaje Z39.50 especifican una base de datos por lo que, a nivel de interrogación, el cliente Z39.50 formula una petición Z por cada base de datos distinta y, a nivel del proveedor, se crea un servidor Z39.50 por cada base de datos distinta.

La sincronización de múltiples peticiones del lado del interrogador la lleva a cabo el cliente Z39.50. Las peticiones concurrentes a cada base de datos son sincronizadas por el servidor de datos.

5 UNA IMPLANTACIÓN DEL MIZ

Hemos visto como la integración bibliotecaria ha evolucionado desde el intercambio de registros usando el formato MARC, pasando por protocolos como Telnet y Html/Http hasta llegar al protocolo Z39.50, el cual ha probado ser maduro y hoy en día está ampliamente difundido.

Hemos analizado la problemática que surge de la interacción entre distintos sistemas bibliotecarios por la necesidad de conocer sus lenguajes de consulta y recuperación de registros.

Vimos que para disminuir el grado de limitación en la interacción entre sistemas distintos, es deseable conservar sus interfaces humano computadora y de base de datos, para lo cual es necesario que las aplicaciones hagan traducciones de sus lenguajes a los lenguajes de los otros sistemas.

Planteamos tres modelos en los que las traducciones se hacen de diferentes maneras. Mostramos que el modelo óptimo en el manejo de la complejidad del intercambio de información entre sistemas heterogéneos, es el que tiene un lenguaje común para la interacción.

El lenguaje común elegido en este trabajo, es el especificado por el protocolo Z39.50 del cual hemos estudiado sus características principales.

El contexto de desarrollo de este trabajo es el sistema de recuperación de información Alejandría, lo que impone ciertos requerimientos. Para satisfacerlos, se propuso un modelo de integración mediante el protocolo Z39.50, el cual tiene un enfoque modular para ofrecer la capacidad de intercambiar información a una aplicación recuperadora de información. Este modelo lo hemos llamado MIZ.

El MIZ facilita que una aplicación interactúe con sistemas distintos bajo las siguientes características:

- Conservando su interfaz humano computadora.
- Conservando su interfaz con las bases de datos.

- Usando el protocolo Z39.50.

En este capítulo se describe la implementación del MIZ en Alejandría, y se verifica que Alejandría cumpla el rol de interrogador y proveedor de información a sistemas distintos mediante el protocolo Z39.50.

Los principales problemas resueltos en la implantación son los relacionados con la traducción y el mantenimiento de sesión.

5.1 Traducción

El MIZ tiene tres módulos traductores, dos en el lado interrogador y uno en el lado proveedor. La función principal de los traductores es servir de mediadores: entre la interfaz humano computadora y el cliente Z39.50 para el interrogador; entre la interfaz de base de datos y el servidor Z39.50 para el proveedor.

El traductor interrogador traduce la interrogación enviada por la interfaz de usuario en su lenguaje de búsqueda (LB) al Lenguaje Intermedio de Interrogación (LII), provisto por el MIZ. Cuando recibe la respuesta del cliente Z39.50 en Lenguaje Intermedio de Respuesta (LIR), también provisto por el MIZ, la traduce al lenguaje de despliegue de la interfaz de usuario (LD). Estas traducciones se ilustran en la figura 5.1.

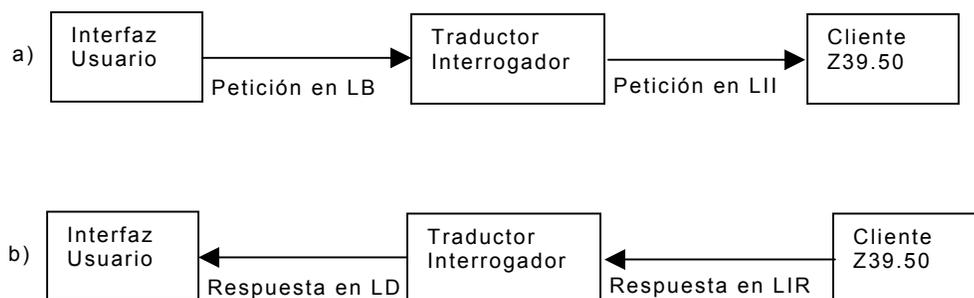


Figura 5.1: Dinámica del Traductor Interrogador.

a) Traductor Interrogador haciendo una petición. b) Traductor Interrogador recibiendo una respuesta

El traductor proveedor recibe una instrucción en Z39.50 proveniente del servidor Z39.50 y traduce al lenguaje de la interfaz de su base de datos que, en el MIZ, la llamamos servidor de datos. Cuando recibe la respuesta hace la traducción inversa. Abreviamos los lenguajes de respuesta e interrogación del servidor de datos como LISD y LRSO respectivamente.

Del lado proveedor no se necesita un lenguaje que integre varias salidas porque el servidor Z39.50 fabrica una salida por cada cliente Z39.50 que le interroga. Cada petición en lenguaje Z39.50 especifica una sola base de datos, tal como se ve en la figura 5.2.

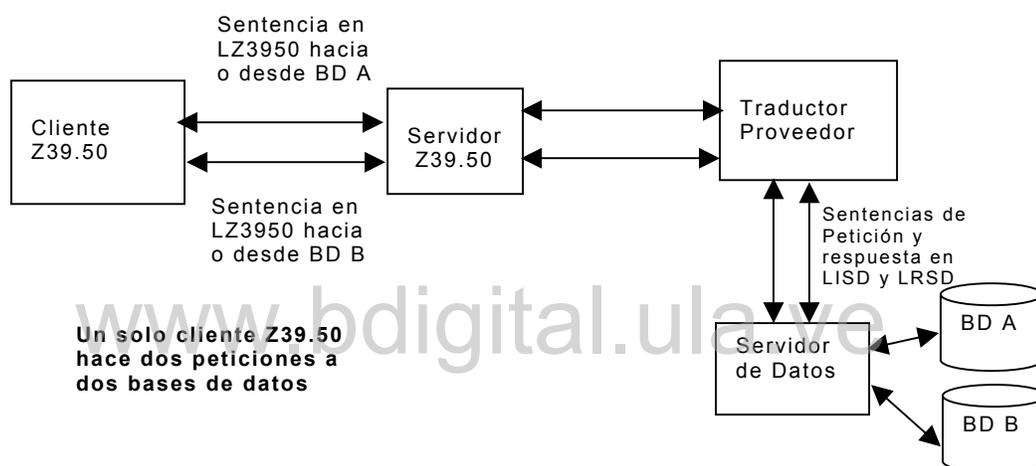


Figura 5.2: Petición en Z39.50 a una sola base de datos

Adicionalmente, el cliente Z39.50 tiene un submódulo de traducción puesto que recibe una sola sentencia de interrogación en LII que la traduce en varias instrucciones Z39.50: una para cada servidor Z. Después, recibe asíncronamente conjuntos de registros de cada uno de los servidores Z en lenguaje Z39.50 y los integra en una sola salida secuencial que especifica el origen, el número y el contenido de cada registro. El lenguaje en el que se expresa esta salida es el LIR. El tiempo de espera por una respuesta está definido por configuración, si un servidor Z39.50 no responde en este lapso de tiempo, el Cliente Z39.50 construye un mensaje en LIR. En la figura 5.3 se aprecia el submódulo de traducción del cliente Z39.50.

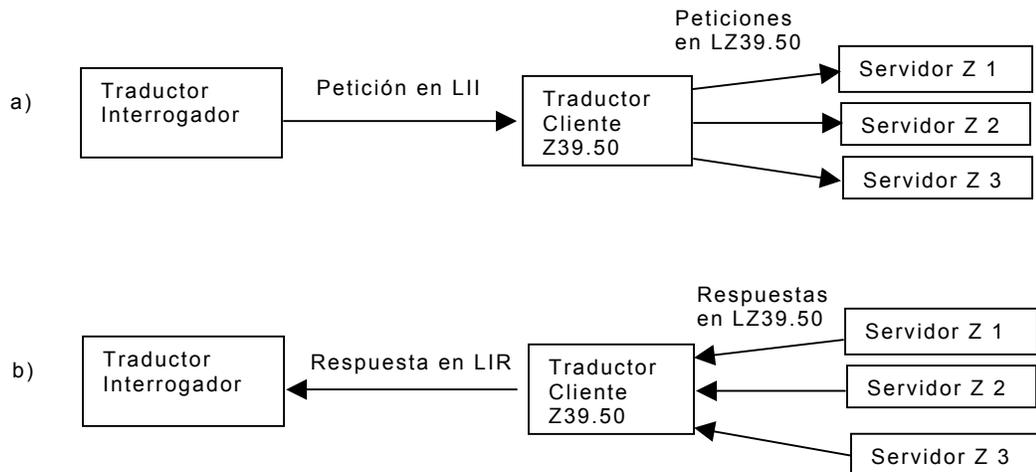


Figura 5.3: Dinámica de los servidores Z39.50.

a) Peticiones a varios servidores Z. b) Recepción de respuestas asíncronas de los servidores Z.

www.bdigital.ula.ve

5.1.1 Requerimientos de traducción

Para facilitar el discurso, usaremos la palabra origen para calificar la gramática y el lenguaje de la sentencia que un traductor recibe, y la palabra destino para calificar el lenguaje de la sentencia que un traductor produce.

Los tres traductores: el interrogador, el proveedor y el cliente Z39.50 tienen en común los siguientes requerimientos:

- Chequear la sintaxis de las sentencias que reciben en la gramática origen.
- Interpretar estas sentencias de acuerdo al lenguaje destino.
- Convertir esta sentencia al lenguaje destino.
- Rechazar una sentencia si no es reconocida.

5.1.1.1 Requerimientos del traductor interrogador

- Conocer las gramática de los siguientes lenguajes:

- Lenguaje de búsqueda de la interfaz de usuario.
- Lenguaje de despliegue de la interfaz de usuario.
- Lenguaje intermedio de interrogación que comprende las siguientes sintaxis:
 - Sintaxis del lenguaje de búsqueda del protocolo Z39.50.
 - Sintaxis del lenguaje de especificación de servidores Z39.50 y sus bases de datos, descrita por el MIZ.
- Lenguaje intermedio de respuesta.
- Formato MARC.
- Recibir una sentencia de interrogación en LB y producir otra de interrogación en LII.
- Recibir una sentencia de respuesta en LIR y producir otra de respuesta en LD.

5.1.1.2 Requerimientos del traductor del cliente Z39.50

- Conocer las gramáticas de los siguientes lenguajes:
 - Lenguaje intermedio de interrogación.
 - Lenguaje intermedio de respuesta.
 - Lenguaje Z39.50.
- Recibir una sentencia de interrogación en LII y producir varias sentencias de interrogación en LZ39.50: una por cada servidor Z con el que interactúe.
- Conocer el estado de la sesión con cada servidor Z39.50 para cambiar pertinentemente la sentencia de interrogación origen.
- Recibir asíncronamente una respuesta por cada servidor Z en LZ39.50 y producir una sola sentencia de salida en LIR.

5.1.1.3 Requerimientos del traductor proveedor

- Conocer las gramáticas de los siguiente lenguajes:

- Lenguaje de interrogación del servidor de datos.
- Lenguaje de interrogación del servidor de datos.
- Lenguaje de respuesta del servidor de datos.
- Lenguaje Z39.50.
- Formato MARC.
- Recibir una sentencia de interrogación en LZ39.50 y generar otra sentencia de interrogación en LISD.
- Recibir una sentencia de respuesta en LRSD y producir otra sentencia de respuesta en LZ39.50.

5.1.2 Diseño de los traductores

La interfaz de usuario de Alejandría es una interfaz Web por lo que la sintaxis del lenguaje de búsqueda sigue este estándar. El lenguaje de despliegue es HTML. El lenguaje de interrogación del servidor de datos es SQL. El lenguaje de respuesta del servidor de datos es un conjunto de estructuras complejas de datos propias de Alejandría, cada una de las cuales conforma un registro numerado y ordenado secuencialmente.

Los traductores están conformados por dos grupos de submódulos: los que traducen la sentencia de interrogación y los que traducen la sentencia de respuesta.

Las clases a las que nos referiremos en el diseño de los traductores están descritas en el capítulo 4.

5.1.2.1 Diseño del traductor interrogador

El traductor interrogador pertenece a la clase TraductorInterrogador, la cual tiene dos métodos: TraductorLB_LII y TraductorLIR_LD para traducir la interrogación y la respuesta, respectivamente.

El método TraductorLB_LII se implementa con los siguientes submódulos:

- **Verificador de sintaxis del lenguaje de búsqueda:** recorre secuencialmente la sentencia LB buscando los términos de

búsqueda y sus valores, así como, también, los servidores Z39.50 y las bases de datos requeridos.

- **Interpretador de términos LB en Z39.50:** por cada término reconocido, sea éste de búsqueda o la dupla (servidor,base de datos), busca su equivalente en LZ39.50.
- **Constructor de LII:** cuando tiene la equivalencia de todos los términos de búsqueda en LZ39.50, construye una oración válida en LZ39.50, a la que le adiciona la sintaxis de los servidores Z y sus bases de datos según se describe en la gramática de LII.

En la figura 5.4 se ilustra este grupo de submódulos.



Figura 5.4: Submódulos del traductor interrogador relacionados con la búsqueda

El método TraductorLIR_LD se implementa con los siguientes submódulos:

- **Verificador de sintaxis LIR:** es un motor recursivo que recorre la sentencia LIR reconociendo:
 - Grupos de registros y su servidor Z de origen
 - Cantidad de registros componentes de cada grupo
 - Registros
 - Mensajes
- **Interpretador de registros MARC y/o texto en LD:** por cada registro reconocido, interpreta los campos MARC y busca la expresión equivalente en HTML.
- **Constructor de LD:** cuando tiene el equivalente en HTML, construye una salida en lenguaje de despliegue de Alejandría.

En la figura 5.5 se ilustra este grupo de submódulos:



Figura 5.5: Traductor interrogador relacionado con la respuesta

La clase TraductorInterrogador está compuesta por las clases SentenciaLD, SentenciaLB, SentenciaLII, SentenciaLIR y SentenciaMARC las cuales tienen las gramáticas de cada uno de sus lenguajes e implementan los métodos de verificación de sintaxis de sus gramáticas.

5.1.2.2 Diseño del traductor cliente Z39.50

El traductor cliente Z39.50 pertenece a la clase TraductorClienteZ3950, la cual tiene dos métodos: TraductorLII_LZ y TraductorLZ_LIR para traducir la interrogación y la respuesta, respectivamente.

El método TraductorLII_LZ se implementa con los siguientes submódulos:

- **Identificador de servidores Z39.50:** busca secuencialmente en la sentencia LII, cada uno de los servidores Z39.50 y sus bases de datos.
- **Identificador de sentencia de búsqueda LZ39.50:** analiza la sentencia LII para identificar la instrucción de búsqueda en Z39.50.
- **Constructor de sentencia de búsqueda LZ39.50:** determina el estado de la sesión con cada uno de los servidores Z39.50; de acuerdo a esto, construye una sentencia de búsqueda y/o recuperación LZ39.50 para cada uno de estos servidores partiendo de la sentencia de búsqueda origen.

En la figura 5.6 se ilustra este grupo de submódulos.

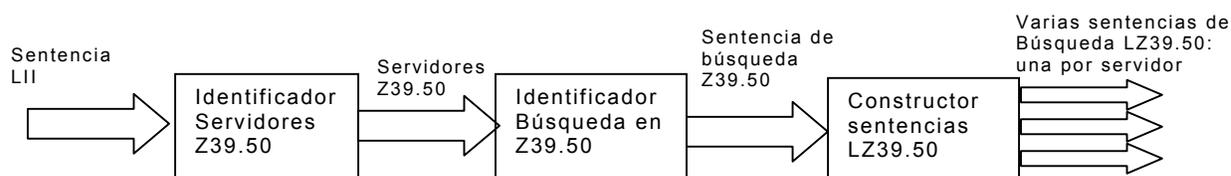


Figura 5.6: Grupo de submódulos del traductor cliente Z39.50 que construyen la interrogación para varios servidores Z39.50

El método TraductorLZ_LIR se implementa con los siguientes submódulos:

- **Extractor de registros en cadena LZ39.50:** por cada servidor que responde, recorre secuencialmente la cadena de caracteres LZ39.50 para extraer los registros o el mensaje del servidor Z39.50.
- **Constructor de sentencia LIR:** cuando se ha recibido la respuesta de todos los servidores o, en su defecto, determinado que no hay respuesta, construye una sentencia LIR que está compuesta de grupos de registros. Cada grupo puede constar de algunos de los siguientes elementos:
 - Nombre del servidor Z origen.
 - Cantidad de registros.
 - Identificador secuencial para cada registro.
 - Registro.
 - Mensajes pertinentes.

En la figura 5.7 se ilustra este grupo de submódulos.



Figura 5.7: Grupo de submódulos del traductor cliente Z39.50 responsables de la respuesta.

La clase TraductorClienteZ3950 está compuesta por las clases SentenciaLII, SentenciaLIR y SentenciaLZ que implementan los métodos de verificación de sus respectivas gramáticas.

5.1.2.3 Diseño del traductor proveedor

El traductor proveedor pertenece a la clase TraductorProveedor, la cual implementa dos métodos: TraductorLZ_LISD y TraductorLRSD_LZ para traducir la interrogación y la respuesta, respectivamente.

El método TraductorLZ_LISD se implementa con los siguientes submódulos:

- **Verificador de sintaxis de la sentencia LZ39.50:** analiza la estructura del protocolo Z39.50 en la que se expresa la búsqueda, para identificar los términos y sus valores, así como, también, la base de datos.
- **Interpretador de términos LZ39.50 en LISD:** por cada término reconocido, busca su equivalente en el lenguaje de interrogación del servidor de datos (SQL).
- **Constructor de sentencia LISD:** cuando la equivalencia está completa, construye una sentencia válida en SQL.

En la figura 5.8 se presenta este grupo de submódulos.



Figura 5.8: Grupo de submódulos del traductor proveedor relacionados con la elaboración de la interrogación.

El método TraductorLRSD_LZ se implementa con los siguientes submódulos:

- **Extractor de registros Alejandría:** recorre secuencialmente la estructura de datos Alejandría retornada y extrae los registros, sus campos y valores.
- **Traductor a MARC:** por cada registro extraído hace la conversión a MARC.
- **Constructor de registros LZ39.50:** cuando todos los registros están en formato MARC, construye la respuesta en lenguaje Z39.50.

En la figura 5.9 se presenta este grupo de submódulos.

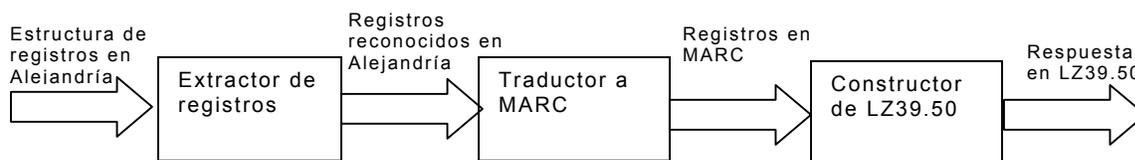


Figura 5.9: Grupo de submódulos del traductor proveedor responsables de construir la respuesta.

La clase TraductorInterrogador está compuesta por las clases SentenciaLISD, SentenciaLRSD, SentenciaLZ39.50 y SentenciaMARC las cuales tienen las gramáticas de cada uno de sus lenguajes e implementan los métodos de verificación de sintaxis de sus gramáticas.

5.2 Sesión

Definamos una sesión como una conexión virtual entre dos procesos con la capacidad de recordar el estado de éstos.

Para implementar exitosamente el MIZ, es necesario mantener una sesión entre la interfaz de usuario y el DBMS de las aplicaciones interrogadora y proveedora de información, respectivamente.

Cuando un usuario consulta una base de datos a través de su interfaz, la respuesta esperada es:

- Una cuenta del total de registros encontrados.
- Un subconjunto de estos registros.

Si el usuario pide un subconjunto de registros, los elementos que interactúan para satisfacer esta consulta necesitan conocer el estado de la interacción entre la interfaz de usuario y el DBMS. Esto hace imprescindible el establecimiento de una sesión entre los dos elementos.

Por otro lado, una aplicación proveedora de información recupera registros a petición de un cliente Z39.50. Si un cliente Z39.50 pide iterativamente registros, los elementos que interactúan para satisfacer el requerimiento necesitan saber de dónde se recuperan estos registros: del conjunto resultante o de la base de datos, luego, tienen que conocer el estado de la interacción. Esto hace necesario el establecimiento de sesión entre el extremo que interroga y el extremo que provee.

La implantación de una sesión pasa primero por identificar cuáles son los elementos que intervienen en la interacción. En la figura 5.10 se identifican estos elementos:

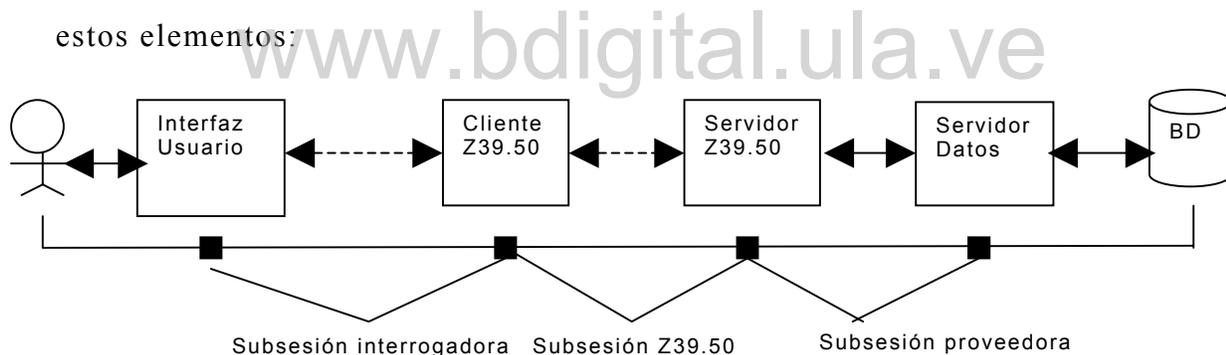


Figura 5.10: Subsesiones que deben mantenerse para mantener la sesión entre los extremos

En la figura 5.10 se nota que para mantener la sesión entre los extremos es necesario mantener subsesiones entre los elementos que interactúan; las hemos llamado subsesión interrogadora, subsesión Z39.50 y subsesión proveedora.

El principal problema que resolver es, entonces, el mantenimiento de las subsesiones.

5.2.1 Diseño de subsesión interrogadora

La subsesión interrogadora se identifica con un nombre único al que llamaremos *ID Interrogación*, el cual es escogido por la interfaz de usuario.

Para encaminar la sesión entre la interfaz de usuario y el cliente Z39.50, se crea un proceso llamado traductor interrogador, el cual recibe la interrogación de la interfaz de usuario junto con su *ID Interrogación*.

El traductor interrogador busca si existe la sesión *ID Interrogación*, si existe, entonces, la usa y sino, la crea. La creación de sesión consiste en iniciar un proceso llamado cliente Z39.50 como un servidor que atiende únicamente al servicio llamado *ID Interrogación*.

Hasta aquí tenemos que:

- Toda comunicación entre cliente Z39.50 e interfaz de usuario es encaminada por el traductor interrogador.
- Toda petición de la interfaz de usuario identificada como *ID Interrogación* es atendida por un cliente Z39.50.
- Los procesos traductor interrogador y cliente Z39.50 son los componentes del MIZ Interrogador que estudiamos en la sección de Traducción.

Usar esta sesión significa que toda interrogación o respuesta entre la interfaz de usuario y su cliente Z39.50 fluye a través de ella.

La sesión se cierra si no es usada por un cierto tiempo. Cerrarla consiste en terminar el proceso cliente Z39.50 correspondiente.

En la figura 5.11 se ilustra la implementación de la subsesión interrogadora.

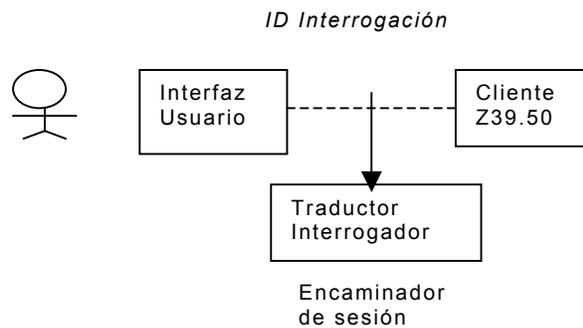


Figura 5.11: Implementación de subsesión interrogadora

5.2.1.1 *Cómo se recuerda el estado de la sesión*

- El cliente Z39.50 porque es un servidor de esa única sesión.
- El traductor interrogador por el *ID Interrogación*.
- La interfaz de usuario por el *ID Interrogación*.

5.2.2 **Diseño de subsesión Z39.50**

La subsesión Z39.50 es mantenida entre el cliente Z39.50 y el servidor Z39.50.

El cliente Z39.50 recibe la consulta por la subsesión interrogadora y la desglosa en una o más consultas a distintos servidores Z39.50. Por cada servidor Z39.50, el cliente Z39.50 busca si existe una sesión.

Si existe la sesión, entonces, el cliente Z39.50 la usa, sino existe, la crea. La creación de sesión está especificada por el protocolo Z39.50.

El servidor Z39.50 es un servicio que se ejecuta en un puerto tcp. Cuando recibe la petición de creación de sesión, crea un *thread* para atenderla.

El cliente Z39.50 identifica la sesión por un nombre único al que llamaremos *ID Z39.50*. El servidor Z39.50 identifica esta sesión porque existe un *thread* para mantenerla.

Hasta ahora tenemos que:

- El cliente Z39.50 mantiene varias sesiones: una por cada servidor Z39.50, cada cual identificada con un *ID Z39.50* diferente.
- El servidor Z39.50 crea un *thread* por cada cliente Z39.50 que solicite sesión.
- Toda comunicación entre cliente y servidor Z39.50 está especificada por el protocolo Z39.50.

Usar esta sesión significa que toda interrogación y respuesta, entre un cliente y un servidor, fluye a través de la sesión *ID Z39.50* para el cliente, y el

puerto tcp asignado al *thread* para el servidor. Cada servidor corre en un puerto tcp diferente, el standard dice que es el puerto 210.

La sesión Z39.50 se cierra por petición del cliente Z39.50.

En la figura 5.12 se muestra la implementación de esta sesión.

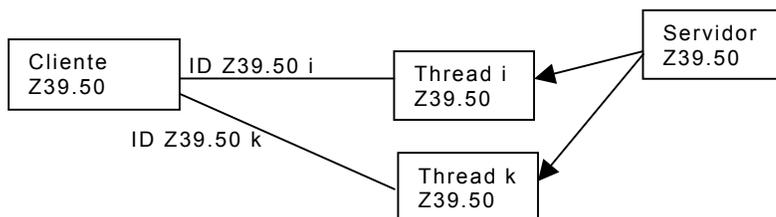


Figura 5.12: Implementación de subsesión Z39.50

5.2.2.1 Cómo se recuerda el estado de la sesión

- El cliente Z39.50 por el *ID Z39.50* de cada sesión.
- El servidor Z39.50 porque existe un *thread* por sesión.

5.2.3 Diseño de subsesión proveedora

La subsesión proveedora se mantiene entre el servidor Z39.50 y el DBMS.

El servidor de datos establece sesiones permanentes con las bases de datos a través del DBMS. Estas sesiones están especificadas por el protocolo de comunicación con bases de datos; el servidor de datos las identifica con un nombre único al que llamaremos *ID BaseDatos*.

El *thread Z39.50* recibe la petición e intenta satisfacerla con los registros del conjunto resultante. Si hay registros que no están en él, se comunica con el servidor de datos usando el identificador *ID BaseDatos*.

El servidor de datos busca la sesión, si existe, entonces, la usa, sino la crea. Crear una sesión es establecer una conexión permanente con el DBMS e identificarla con el *ID BaseDatos*.

Hasta ahora tenemos que:

- Toda comunicación entre el *thread Z39.50* y el DBMS fluye a través del servidor de datos.

- El servidor de datos sirve de encaminador de la sesión con el DBMS.

Usar esta sesión consiste en que toda petición y respuesta entre el *thread* Z39.50 y el DBMS fluye a través de ella.

Esta sesión se cierra por petición del *thread* Z39.50 o porque no se usa por un tiempo. Cerrarla significa eliminar la conexión entre servidor de datos y DBMS.

En la figura 5.13 se ilustra la implementación de esta sesión.

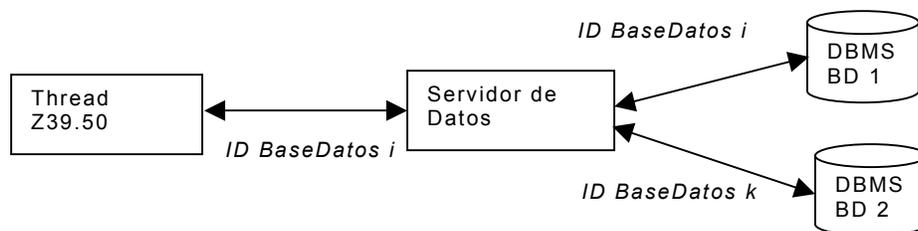


Figura 5.13: Implementación de subsesión proveedora

5.2.3.1 Cómo se recuerda el estado de la sesión

- El servidor de datos por el *ID BaseDatos*.
- El *thread* Z39.50 por el *ID BaseDatos*.

5.2.4 Manejo de sesiones concurrentes

La **conurrencia de subsesiones interrogadoras** es posible porque todo proceso interfaz de usuario tiene asignados:

- Un proceso cliente Z39.50 que lo atiende exclusivamente .
- Un proceso traductor interrogador que encamina sólo esta sesión.

De esta manera, el sistema MIZ interrogador tiene n interfaces de usuario interactuando con un número igual de traductores interrogadores y clientes Z39.50, como se ve en la figura 5.14:

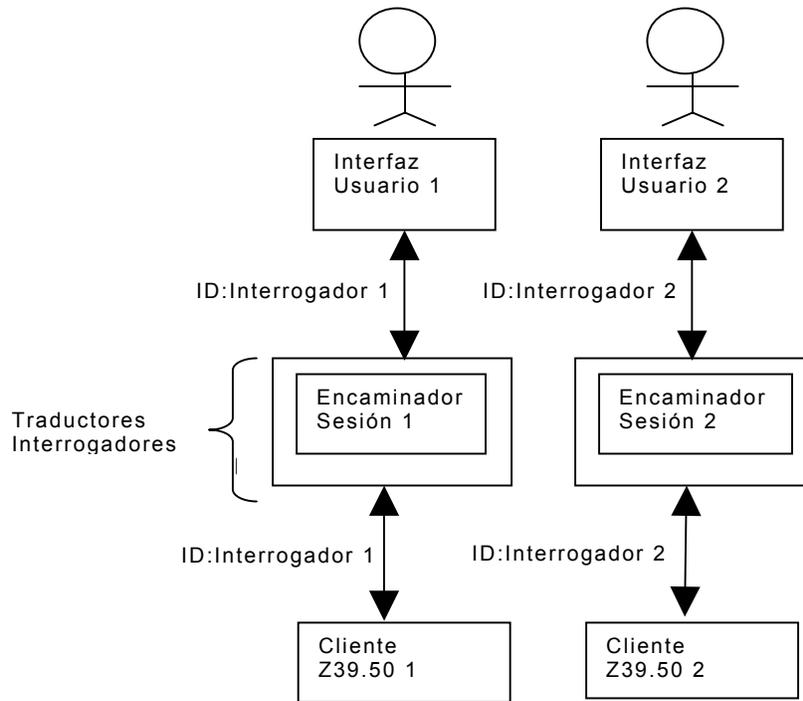


Figura 5.14: Manejo de sesiones interrogadoras concurrentes

La concurrencia de subsesiones Z39.50 es posible porque:

- Cada cliente Z39.50 mantiene múltiples sesiones con los servidores Z39.50 requeridos por su interfaz de usuario
- El cliente Z39.50 crea una sesión por cada dupla distinta compuesta por servidor Z39.50 y base de datos
- Para cada petición de un cliente Z39.50 el servidor Z39.50 crea un *thread* que atiende únicamente esta petición.

De esta manera, el sistema MIZ Interrogador tiene n clientes Z39.50 que mantienen, cada uno de ellos, *múltiples* sesiones con servidores Z39.50. El sistema MIZ Proveedor tiene k *threads* Z39.50 atendiendo igual número de peticiones. Un *thread* Z39.50 atiende una petición proveniente del MIZ Interrogador o de cualquier cliente Z39.50.

En la figura 5.15 se ilustra este esquema de concurrencia.

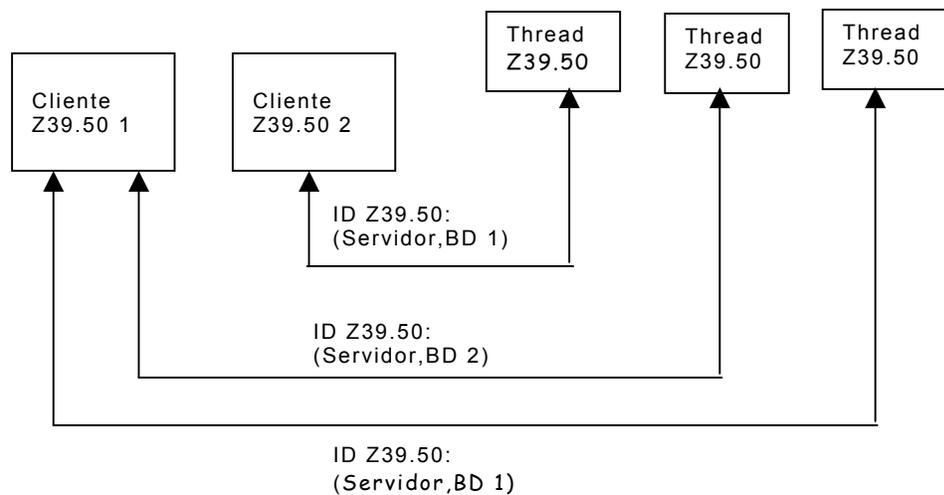


Figura 5.15: Manejo de sesiones Z39.50 concurrentes

La concurrencia de subsesiones proveedoras es posible porque:

- El servidor de datos mantiene n sesiones permanentes con n bases de datos.
- El servidor de datos recibe varias peticiones concurrentes provenientes de los *threads* Z39.50 y las serializa implementando una sección crítica de código.

De esta manera, el sistema MIZ Proveedor tiene n *threads* y un servidor de datos que tiene m sesiones a m bases de datos, tal como se ve en la figura 5.16.

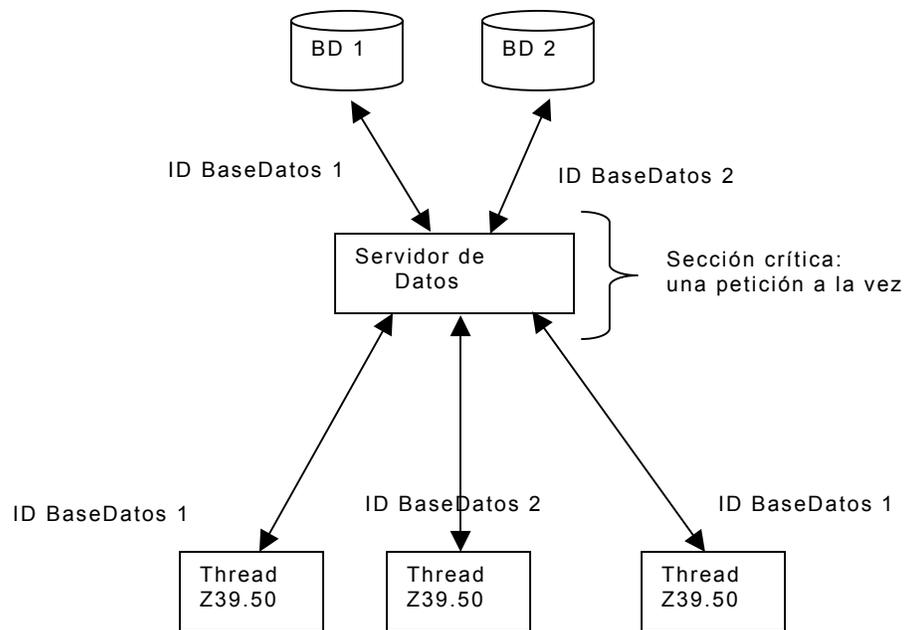


Figura 5.16: Manejo de sesiones proveedoras concurrentes

5.2.5 Integración

La integración de los traductores y el manejo de sesiones concurrentes tiene las características siguientes:

- El proceso traductor interrogador tiene dos submódulos: el traductor y el encaminador de sesión.
- El proceso cliente Z39.50 tiene dos submódulos: el traductor y el manejador de varias sesiones Z39.50.
- El *thread* Z39.50 tiene dos submódulos: el traductor proveedor y el manejador de una sesión Z39.50.
- El servidor de datos mantiene tantas sesiones como bases de datos sirva.

En la figura 5.17 se aprecia el esquema integrado:

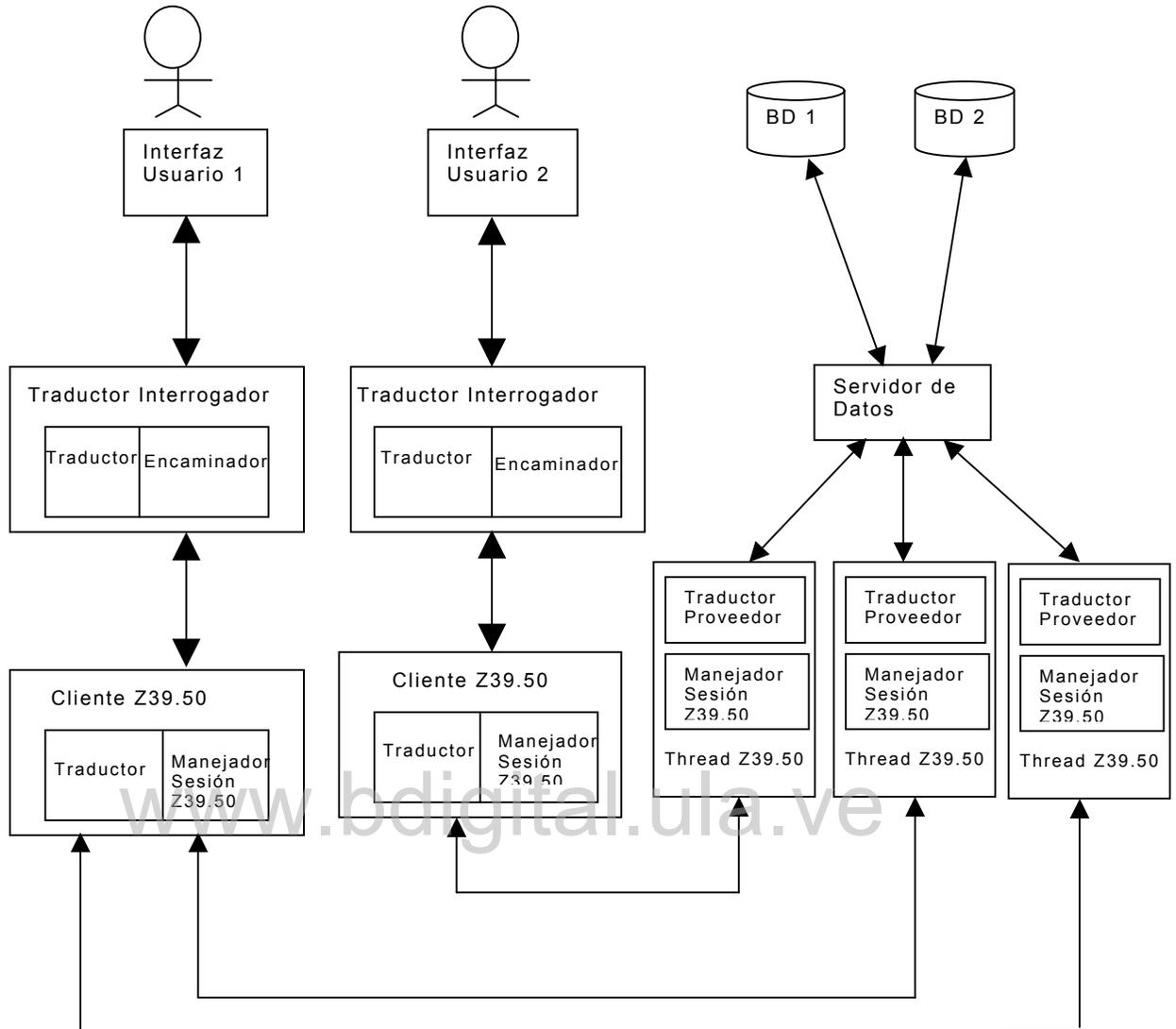


Figura 5.17: Integración de traductores y sesiones concurrentes en el MIZ

5.3 Implantación y pruebas

El entorno de implantación tiene las siguientes características:

- La interfaz de usuario es Web y se comunica vía http con el traductor interrogador. En el Anexo II se muestran ejemplos de estas interfaces.
- El traductor interrogador y el cliente Z39.50 se ejecutan en un entorno Windows 95, 98, NT, 2000 o XP.

- El traductor interrogador se comunica con el cliente Z39.50 por el protocolo de comunicación entre procesos Dynamic Data Exchange (DDE) de Windows [17].
- El servidor Z39.50 se ejecuta en un entorno Windows y presta el servicio en cualquier puerto tcp asignado; el puerto estándar para el protocolo Z39.50 es el 210. Cada *thread* Z39.50 comparte el entorno del proceso padre.
- El servidor de datos Alejandría se ejecuta en el servidor Z39.50.
- El DBMS se ejecuta local o remotamente.
- La comunicación entre la interfaz de base de datos Alejandría y el DBMS es a través del protocolo ODBC [18].

5.3.1 Bibliotecas utilizadas

El traductor interrogador enlaza estáticamente las bibliotecas del sistema Alejandría:

- Interpretador de lenguaje de búsqueda.
- Presentación de registros.
- Interpretador MARC.

El cliente Z39.50 usa las bibliotecas:

- YAZ [19] que es una implementación del protocolo Z39.50, enlazada dinámicamente.
- ZOOM [20] que es una interfaz para YAZ, enlazada dinámicamente.
- DDEML [17] implementación del protocolo DDE, enlazada dinámicamente.

El servidor Z39.50 usa la biblioteca YAZ enlazada dinámicamente.

El traductor proveedor usa las bibliotecas:

- Recuperación de registros de Alejandría, enlazada dinámicamente.
- Escritor de MARC de Alejandría, enlazada estáticamente.
- YAZ, enlazada dinámicamente.

El servidor de datos usa la biblioteca de conexión a bases de datos de Alejandría, enlazada dinámicamente.

5.3.2 Compilador y programas

El compilador usado para los módulos programados es MS Visual C/C++ 6.0.

Los módulos programados y sus líneas de código son:

- Traductor interrogador: 3260 líneas
- Cliente Z39.50: 1611 líneas
- Traductor proveedor: 511 líneas
- Servidor de datos: 386 líneas

5.3.3 Pruebas

Las principales pruebas del MIZ Proveedor han sido realizadas por Indexdata [21], que es una empresa danesa que implementa el protocolo Z39.50. La biblioteca YAZ es desarrollada por esta empresa. Indexdata tiene un robot que prueba los servidores Z39.50 registrados en su sitio Web, cuya dirección es <http://www.indexdata.dk/targettest/c.php>. La implementación del MIZ Proveedor Alejandría está registrado en este sitio y puede ser verificada buscando su entrada y probando:

- La entrada es Ciudad Alejandría.
- Esta entrada tiene tres bases de datos: bolivar, revistaActual y Igula.

El robot de Indexdata prueba todos los días al MIZ Proveedor Alejandría, y hasta la fecha, el promedio de disponibilidad es de 97%.

Existe un sitio de servidores de prueba mantenido por la Biblioteca del Congreso de Estados Unidos, en el que está registrado el MIZ Proveedor Alejandría con dos servidores Z39.50 que proveen información de tres y dos bases de datos, respectivamente.

- La dirección es:
<http://lcweb.loc.gov/z3950/agency/resources/testport.html>
- La entrada es Hacer Sistemas: Alejandría Z39.50.

Las principales pruebas del MIZ Interrogador son realizadas por los usuarios que se conectan a la Biblioteca del Congreso de Estados Unidos, para buscar implementaciones relacionadas con el protocolo. EL MIZ Interrogador está registrado en un sitio de implementadores Z39.50 mantenido por la Biblioteca del Congreso.

- La dirección es:
<http://lcweb.loc.gov/z3950/agency/register/entries.html>
- La entrada es Hacer Sistemas.
- La dirección del MIZ Interrogador Alejandría registrado en este sitio es: <http://ciudad-alejandria.com/serverz>

En Saber ULA [4], existe otra implementación del MIZ Interrogador Alejandría que puede ser probado en la dirección:

<http://www.saber.ula.ve/servez>

Adicionalmente, el MIZ ha sido sometido a algunas pruebas de esfuerzo tales como:

- Crear múltiples sesiones interrogadoras que, concurrentemente, hacen consultas; esto con el fin de verificar la capacidad de atención a varios usuarios que interactúan con el MIZ Interrogador.
- Consultar a varios servidores Z del MIZ Proveedor para verificar concurrencia, tiempo de respuesta y que los registros de diferentes servidores y bases de datos no se mezclen.

5.4 Conclusión

La implementación del MIZ Alejandría ha sido exitosa porque verifica los siguientes aspectos:

- La interoperabilidad de Alejandría con otros sistemas como interrogador y proveedor de información.
- La viabilidad de implementar el MIZ siguiendo sus principios:
 - Conservación de las interfaces humano computadora y de base de datos de Alejandría.
 - Uso del protocolo Z39.50 para lograr la integración.
 - Uso de traductores.
- Manejo exitoso de sesiones concurrentes entre interfaz de usuario y DBMS.

www.bdigital.ula.ve

6 CONCLUSIÓN

Los aportes de esta tesis se resumen en dos aspectos:

- La propuesta de un modelo que integra sistemas bibliotecarios heterogéneos mediante el protocolo Z39.50 que hemos llamado MIZ.
- La implantación exitosa de este modelo.

La principal bondad del modelo es brindar un esquema para que las aplicaciones bibliotecarias adquieran la capacidad de integrarse con otros sistemas.

El modelo resuelve la interoperabilidad entre sistemas distintos usando traductores, gracias a los cuales se pueden conservar las interfaces humano computadora y de bases de datos de la aplicación que incorpora el modelo.

El modelo provee las gramáticas de dos lenguajes que ayudan a extender el uso del protocolo Z39.50, para que un cliente Z39.50 interroge a varios servidores Z y pueda fabricar una sola salida para el usuario extremo.

La implantación del modelo facilita la comprensión de su estructura y verifica que, cuando una aplicación recuperadora de información lo incorpora, adquiera la capacidad de interactuar con sistemas distintos, superando la limitación de la multiplicidad de lenguajes e interfaces.

La implantación del modelo requiere el manejo de sesiones y concurrencia como dos aspectos insoslayables para lograr una implantación exitosa.

En lo que concierne al modelo, la mejora de este trabajo puede decirse que es su extensión para incluir:

- El manejo de sesiones.
- La concurrencia de estas sesiones.

Aunque en la implementación estos son aspectos resaltantes, el modelo no muestra explícitamente cómo se pueden resolver estas características.

El proveedor MIZ Alejandría puede ampliarse para incorporar más características del protocolo Z39.50 tales como:

- Préstamo remoto.
- Actualización remota de bases de datos.
- Más operaciones con los conjuntos resultantes.

El interrogador MIZ Alejandría podría, también, ampliarse para permitir catalogación remota.

www.bdigital.ula.ve

7 ANEXO I

7.1 Sección A- Lenguaje Intermedio de Interrogación

El significado de las etiquetas del LII así como sus valores por omisión y su obligatoriedad o no, se especifica a continuación:

Etiqueta	Significado	Valor por omisión	Oblig.
<sintaxis de búsqueda de Z3950>	Gramática de búsqueda del lenguaje Z39.50	@attr 1=4 manuel	Si
<condiciones>	Parámetros de configuración para el servidor Z39.50	*	Si
<servidores>	Dirección de los servidores Z39.50 en los que se va a buscar	*	No
<sesión>	Id de la sesión interrogadora	Fecha, hora y dirección IP de la máquina en la que se ejecuta la interfaz de usuario	Si
<primeros>	Número de registros a recuperar en la primera consulta	5	Si
<siguientes>	Número de registros a recuperar en las consultas siguientes	5	Si
<anterior>	Se busca en el conjunto resultante existente	N	Si
<desde>	Número de registro en el conjunto resultante a partir del cual se recupera	1	Si
<continua>	La siguiente recuperación es a partir del último registro recuperado	S	Si
<dirección ip>	Dirección ip del servidor Z39.50	N/A	No
<puerto>	Puerto tcp en el que atiende el servidor Z39.50	N/A	No
<basedatos>	Nombre de la base de datos en la que se quiere buscar	N/A	No
<vacío>	Cadena vacía	'/0'	Si
<cadena>	Conjunto de caracteres ASCII	**	Si
<dígito>	Conjunto de números del 0 al 9	**	Si
<entero>	Número entero	**	Si

*El valor por omisión es el correspondiente a los elementos que componen esta etiqueta.

**El valor por omisión de esta etiqueta depende del valor de la etiqueta que la utilice.

7.2 Sección B – Lenguaje Intermedio de Respuesta

El significado de las etiquetas del LIR así como sus valores por omisión y su obligatoriedad o no, se especifica a continuación:

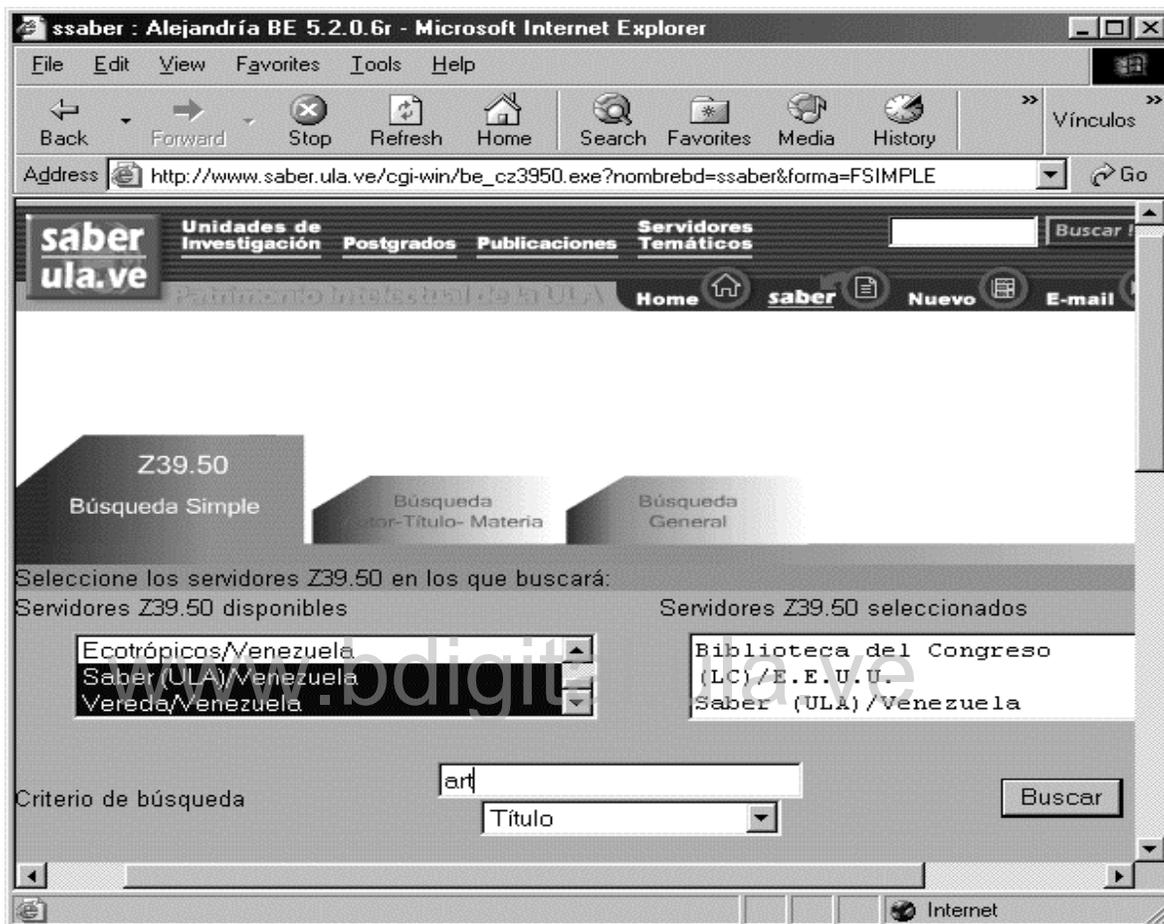
Etiqueta	Significado	Valor por omisión	Oblig.
TOKENINICIO	Marca de inicio de la cadena que contiene todos los registros recuperados	TOKENINICIO	Si
TOKENFIN	Marca de fin de la cadena que contiene todos los registros recuperados	TOKENFIN	Si
<contenido>	Cadena que contiene todos los registros recuperados	*	Si
<tokenexiste>	Número de registros que satisfacen la consulta en un servidor Z39.50	*	Si
<resto>	Cadena de registros recuperados no interpretada todavía	*	Si
<tokenmensajecon>	Mensajes relativos al estado de un servidor Z39.50	*	Si
<vacio>	Cadena vacía	‘/0’	Si
<tokenregistro>	Identificador de registro	*	Si
<tokenvalor>	Contenido del registro	*	Si
<tokenmensajer>	Mensajes relativos a la composición del registro	*	Si
TOKENVALOR	Marca de inicio del contenido del registro	TOKENVALOR	Si
<cadena>	Conjunto de caracteres ASCII	**	Si
TOKENMENSAJECON	Marca de inicio del mensaje relativo al estado del servidor Z39.50	TOKENMENSAJECON	Si
TOKENMENSAJERES	Marca de inicio del mensaje relativo a la composición del registro	TOKENMENSAJERES	Si
TOKENREGISTRO	Marca de inicio de la identificación del registro	TOKENREGISTRO	Si
TOKENEXISTE	Marca de inicio del número de registros por servidor z39.50	TOKENEXISTE	Si
<digito>	Conjunto de dígitos del 0 al 9	**	Si
<entero>	Número entero	**	Si

*El valor por omisión es el correspondiente a los elementos que componen esta etiqueta.

**El valor por omisión de esta etiqueta depende del valor de la etiqueta que la utilice.

8 ANEXO II

Ejemplo de uso de la interfaz de Saber ULA para buscar en tres bibliotecas:



El resultado de la búsqueda usando la misma interfaz:

Resultado de la consulta Z39.50 a la base de información: Vereda/Venezuela
Se buscó en SABER-ULA

- 1 [\(ZKM\) Center for Art and Media](#)
- 2 ¿Quieres conocer algunos tipos de arte interactivo?
[Vereda](#)
- 3 010101: Art in Technological Times
[Arte Cibernético, Exposiciones de Arte Cibernético](#)
- 4 [A Parte Rei](#)
[A Parte Rei](#)
- 5 [AARON, Kurzweil CyberArt Technologies, Inc.](#)
[Net Art, Browser Art](#)

Resultado de la consulta Z39.50 a la base de información: Saber (ULA)/Venezuela
Se buscó en SABER-ULA

- 1 [1er Encuentro de egresados de Historia del Arte](#)
Mérida, 3 al 6 de Diciembre de 2002
- 2 [Arte A Priori. Encuentro de egresados de Historia del Arte](#)
Mérida, del 3 al 6 de Diciembre de 2002
- 3 [Concierto de María Rivas con la banda Cuarto Tiempo](#)
Mérida, Viernes 6 de Diciembre de 2002
- 4 [Concierto: Arte por la paz](#)
Mérida, sábado 22 de Febrero de 2003
- 5 [Encuentro nacional de artesanos de la madera](#)
Mérida, del 28 al 31 de Mayo de 2002

Resultado de la consulta Z39.50 a la base de información: Biblioteca del Congreso (LC)
Se buscó en SABER-ULA

- 1 010101 : art in technological times.
San Francisco Museum of Modern Art, San Francisco, Calif. : 2001.
- 2 10 aspects of Estonian life.
Eesti Raamat, Tallinn, 1967.
- 3 10-nin no āato direkt-āatachi = Ten art directors in contemporary Japan /
Seibundō Shinkōsha, Tōkyō : , Shinsō Shōwa 58 [1983]
- 4 10p00s Premio nazionale di architettura Trevi Flash Art Museum /
Giancarlo Politi, [Milano] : [1999]
- 5 100 contemporary prints at the Jewish museum. Pratt Graphic Art center
1964.

--> (1) 2 3 4 5 6 7 8 9 10 ...

Formato de los registros Desde el registro Registros a
Alejandría número recuperar Recuperar

9 BIBLIOGRAFÍA Y REFERENCIAS

- [1] José Gregorio Silva, Alejandria una red transinstitucional de servicios de teleinformación, un modelo para el desarrollo de aplicaciones sobre bases de información y una metodología ágil para el desarrollo de software. Tesis doctoral de la Universidad Central de Venezuela, noviembre del 2001. <http://alejandria.hacer.ula.ve>
- [2] ISO 10160 -- Information and Documentation - Interlibrary Loan Application Service Definition for Open Systems Interconnection 1991. ISO 10161 -- Information and Documentation - Interlibrary Loan Application Protocol Specification for Open Systems interconnection 1991.
- [3] MARC 21 Specifications for Record Structure, Character Sets, and Exchange Media. Library of Congress.
- [4] Saber Ula, Patrimonio intelectual de la Universidad de Los Andes en Internet.
<http://www.saber.ula.ve>
- [5] ANSI/NISO Z39.50-1995. Revision of ANSI/NISO-1992. NISO Press. ISSN: 1041-5653 National Information Standards series. ISBN: 1-880124-22-X
- [6] The Linked Systems Project : a networking tool for libraries / compiled and edited by Judith G. Fenly and Beacher Wiggins. Dublin, Ohio : OCLC Online Computer Library Center, c1988. viii, 138 p. : ill. ; 23 cm.
- [7] National Information Standards Organization (NISO). American National Standard Z39.50, Information Retrieval Service Definition and Protocol Specifications for Library Applications (New Brunswick, NJ: Transaction Publishers; 1988).
- [8] ANSI/NISO Z39.50 (1988), versión 1.
- [9] ISO 10162 International Organization for Standardization (ISO). Documentation -- Search and Retrieve Service Definition, 1992. ISO 10163 International Organization for Standardization (ISO). Documentation -- Search and Retrieve Protocol Definition. 1992

- [10] Register of Z39.50 Implementors.
<http://lcweb.loc.gov/z3950/agency>
- [11] ANSI/NISO Z39.50-1992 (version 2) Information Retrieval Service and Protocol: American National Standard, Information Retrieval Application Service Definition and Protocol Specification for Open Systems Interconnection, 1992
- [12] Clifford Lynch, The Z39.50 Information Retrieval Standard. Part 1, A Strategic View of Its Past, Present and Future. En D-Lib Magazine . April, 1997. ISSN 1082-9873
- [13] Information retrieval (Z39.50) application service definition and protocol specification : approved May 10, 1995 by the American National Standards Institute. Developed by the National Information Standards Organization., Bethesda, Md. NISO Press,1995. National information standards series,
- [14] Z39.50 Attribute Architecture,
Version 1.1 July 9, 1999
- [15] MARC 21 format for bibliographic data : including guidelines for content designation prepared by Network Development and MARC Standards Office, Library of Congress, in cooperation with Standards and Support, National Library of Canada.1999 ed.
Washington [D.C.] : Library of Congress, Cataloging Distribution Service, 1999.
- [16] MARC 21 concise formats prepared by Network Development and MARC Standards Office, Library of Congress.
2002 ed.Washington, D.C. : Library of Congress, 2002.
- [17] MSDN Library Visual Studio 6.0.
- [18] Microsoft ODBC software development kit [computer file].
Version 1.0.[Redmond, Wash.] : Microsoft, c1992.
2 computer disks ; 3 1/2 in. + 1 programmer's reference (xv, 548 p. : ill. ; 28 cm.) + SDK guide (vi, 81 p. : ill. ; 28 cm.)
- [19] YAZ.
<http://www.indexdata.dk/yaz/>
- [20] ZOOM. Yaz user's guide and reference.
<http://www.indexdata.dk/yaz/doc/zoom.options.php>

- [21] Indexdata.
<http://www.indexdata.dk/>
- [22] Mary E. Jackson, The Application of the ILL Protocol to Existing ILL Systems, 63rd IFLA General Conference - Conference Programme and Proceedings - August 31- September 5, 1997.
- [23] Charles Lakos and John Lamp The Incremental Modelling of the Z39.50 Protocol using object petri Nets, Computer Science, University of Adelaide and Management Information Systems, Deakin University.
- [24] Connecting and Sharing: the Emerging Role of Z39.50 in Library Networks. Paper by Andrew Wells, Judith Pearce, Linda Groom, Bronwyn Lee, National Library of Australia, presented at the VALA Conference, Melbourne, 28th-30th January 1998.
- [25] Terrence W. Pratt, Lenguajes de Programación, Prentice-Hall Latinoamericana, S.A., Segunda Edición, México 1987.

www.bdigital.ula.ve