



INSTITUTO POLITECNICO NACIONAL
CENTRO DE INVESTIGACION EN CIENCIA APLICADA Y TECNOLOGIA AVANZADA
UNIDAD ALTAMIRA



**"ARQUITECTURA DE SOFTWARE PARA LA
BÚSQUEDA Y CONSULTA DE RECURSOS
DISTRIBUIDOS SOBRE INTERNET"**

T E S I S

Que para obtener el grado de:

MAESTRO EN TECNOLOGÍA AVANZADA

Presenta:

ING. JESÚS MANUEL BORREGO IBARRA

Director de tesis:

DR. EUSTORGIO MEZA CONDE

ALTAMIRA, TAMPS.

AGOSTO DE 2008



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

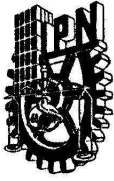
CARTA CESIÓN DE DERECHOS

En la Ciudad de Altamira el día 18 del mes de junio del año 2008, el (la) que suscribe Jesús Manuel Borrego Ibarra alumno (a) del Programa de Maestría en Tecnología Avanzada con número de registro B040823, adscrito a CICATA-IPN Unidad Altamira, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de el Dr. Eustorgio Meza Conde y cede los derechos del trabajo intitulado Arquitectura de Software para la Búsqueda y Consulta de Recursos Distribuidos sobre Internet, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección itmantec@hotmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Jesús Manuel Borrego Ibarra

Nombre y firma



INSTITUTO POLITECNICO NACIONAL SECRETARIA DE INVESTIGACION Y POSGRADO

ACTA DE REGISTRO DE TEMA DE TESIS Y DESIGNACION DE DIRECTOR DE TESIS

México, D.F. a 13 de Mayo del 2008

El Colegio de Profesores de Estudios de Posgrado e Investigación del CICATA Altamira en su sesión ordinaria No. _____ celebrada el día _____ del mes de May-08 conoció la solicitud presentada por el(la) alumno(a):

Borrego Ibarra Jesús Manuel
Apellido paterno materno nombre

Con registro:

B	0	4	0	8	2	3
---	---	---	---	---	---	---

Aspirante al grado de: **MAESTRIA EN TECNOLOGIA AVANZADA**

Se designa al aspirante el tema de tesis titulado:
ARQUITECTURA DE SOFTWARE PARA LA BUSQUEDA Y CONSULTA DE RECURSOS DISTRIBUIDOS SOBRE INTERNET

De manera general el tema abarcará los siguientes aspectos:
Diseñar e implementar una metabase de datos distribuida, diseñar e implementar un portal Web que permita al usuario realizar búsquedas y consultas de recursos distribuidos, implementar el sistema peer-to-peer mutella como mecanismo de comunicación y transferencia de recursos distribuidos, diseñar y desarrollar un programa que implemente el protocolo Gnutella para la búsqueda de recursos distribuidos, diseñar y desarrollar los protocolos de comunicación para los procesos de búsqueda y consulta de recursos distribuidos a través de Internet, desarrollar un programa servidor tipo Servent que coordine el procesamiento de consultas distribuidas e integre los resultados, desarrollar una interfaz Web que presente los resultados del proceso de extracción distribuida de recursos y los muestre en forma de gráfica o archivo de texto a través de Internet.

- 2.- Se designa como Director de Tesis al C. Profesor:
Dr. Eustorgio Meza Conde
- 3.- El trabajo de investigación base para el desarrollo de la tesis será elaborado por el alumno en:
el CICATA-IPN Unidad Altamira, Tamaulipas
que cuenta con los recursos e infraestructura necesarios.
- 4.- El interesado deberá asistir a los seminarios desarrollados en el área de adscripción del trabajo desde la fecha en que se suscribe la presente hasta la aceptación de la tesis por la Comisión Revisora correspondiente:

El Director de Tesis

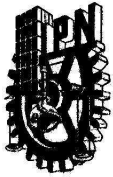
Dr. Eustorgio Meza Conde

El Aspirante

Jesús Manuel Borrego Ibarra

El Presidente del Colegio

Dr. Felipe de Jesús Carrillo Romo



**INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

ACTA DE REVISIÓN DE TESIS

En la Ciudad de Altamira Tamaulipas siendo las _____ horas del día _____ del mes de junio del 2008 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del CICATA-IPN para examinar la tesis de titulada:
Arquitectura de Software para la Búsqueda y Consulta de Recursos Distribuidos sobre Internet

Presentada por el alumno:

<u>Borrego</u> Apellido paterno	<u>Ibarra</u> Apellido materno	<u>Jesús Manuel</u> Nombre(s)
Con registro:		
B	0	4
0	8	2
		3

aspirante de:

MAESTRÍA EN TECNOLOGÍA AVANZADA

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

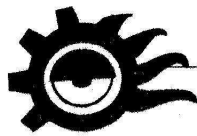
LA COMISIÓN REVISORA

Director de tesis

EUSTORGIO MEZA
Dr. Eustorgio Meza Conde

[Signature]

Dr. Javier Artero Montes de Oca Valero



[Signature]

Dr. Orzo Sánchez Montante

[Signature]

Mc. Hernán Beraza Vázquez

CENTRO DE INVESTIGACIÓN EN CIENCIA APLICADA Y TECNOLOGÍA AVANZADA DEL I.P.N. UNIDAD ALTAMIRA

[Signature]

Dr. Miguel Antonio Domínguez Crespo

Dr. Miguel Angel Arronte García

EL PRESIDENTE DEL COLEGIO

[Signature]

Dr. Felipe de Jesús Carillo Romo

Agradecimientos

Ante todo debo agradecer a mis Padres Guadalupe Ibarra Zurita y José Manuel Borrego Hernández por su cariño y apoyo para salir a delante con mis estudios.

A mi Director de tesis el Dr. Eustorgio Meza Conde por su apoyo y amistad. Gracias por ayudarme a forjar mi carácter, a tener disciplina, a no temer a nada en la vida y por enseñarme a ser más objetivo al realizar mi proyecto.

Agradezco a mi compañero y amigo M.C. Rogelio Ortega Izaguirre por sus incontables consejos y por compartir conmigo su conocimiento y experiencia. Además, le agradezco profundamente por las interminables horas de trabajo que me permitieron concluir exitosamente este proyecto de investigación.

Al Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada, Unidad Altamira por todos los recursos académicos y financieros que fueron destinados a este trabajo de Investigación y a su director el Dr. Felipe de Jesús Carrillo Romo por su apoyo.

Al Centro Nacional de Ciencia y Tecnología por la beca de estudios que me otorgó y por los recursos que aportó al proyecto Red de Observaciones y Pronósticos de Variables Oceánicas en las Costas y Puertos del Golfo de México (ROPVO-GM) del cual esta tesis forma parte.

A mi jurado de tesis: Dr. Javier Arturo Montes de Oca Valero, Dr. Orzo Sánchez Montante, Dr. Miguel Ángel Arronte García y el Dr. Miguel Antonio Domínguez por su amistad y tiempo invertido en la revisión de este trabajo de investigación.

A mis amigos Alejandro Olivares Torres, Alejandro Dávila Peña, Adán Hernández Sánchez, Hernán Peraza Vázquez, Aldo Hernandez Olivares, Juan Rodriguez Azuara, Pedro Jacobo Perez Vega y a todo el escuadron del CICATA-IPN Unidad Altamira por su amistad y apoyo.

Esta investigación recibió el apoyo del "Consejo Nacional de Ciencia y Tecnología (CONACYT)" a través del proyecto 39382-U y la "Coordinación de Posgrado e Investigación (CGPI)" de México Instituto Politecnico Nacional a través de los proyectos 20030593, 20040488, 20050768.

Contenido

Lista de Figuras	iv
Lista de Tablas	vi
Resumen	vii
Abstract	viii
Glosario	ix
1 Introducción	1
1.1 Estado del Arte	2
1.2 Justificación	5
1.3 Planteamiento del problema	6
1.4 Hipótesis	7
1.5 Objetivo general	7
1.6 Objetivos específicos	8
1.7 Organización del Documento	8
2 Marco Teórico	10
2.1 Arquitectura de Software	10
2.2 Sistemas Distribuidos	11
2.2.1 Arquitectura Cliente/Servidor	11
2.2.2 Arquitecturaa Peer-to-Peer	12
2.3 Protocolos de comunicación	13
2.3.1 Sistema Mutella	14
2.3.2 Especificaciones del protocolo Genutella V0.4	15
2.3.3 JDBC	15
2.4 El Internet	15
2.4.1 La Word Wide Web (WWW)	16
2.4.2 El lenguaje HTML y localizar URL	16
2.4.3 Sistemas de Información Basados en Web (SIBW)	17
2.5 Esquemas de metadatos	18
2.6 Modelo en Cascada	19

3	Metodología	21
3.1	Metabase de Datos	21
3.2	Búsqueda Distribuida	24
3.2.1	Interfaz de Búsqueda	25
3.2.2	Intermediario Web	25
3.2.3	Componente de Comunicaciones	26
3.2.4	Presentación de Resultados	32
3.3	Consulta Distribuida	36
3.3.1	Interfaz de Consulta	37
3.3.2	Intermediario de Consulta	40
3.3.3	Extracción de Datos	42
3.3.4	Presentación de Resultados	43
4	Resultados	45
4.1	Metabase de Datos	45
4.1.1	Definición del Esquema de Metadatos	45
4.1.2	Creación de la Metabase de Datos	47
4.1.3	Creación del Diccionario de Términos	47
4.1.4	Creación del Depósito de Metadatos	48
4.1.5	Creación del Depósito Temporal de Metadatos	50
4.2	Proceso de Búsqueda Distribuida	51
4.2.1	Acceso a la Interfaz de Búsqueda	51
4.2.2	Intermediario Web	54
4.2.3	Proceso de Búsqueda Local	56
4.2.4	Proceso de Búsqueda Remota	61
4.2.5	Resultados de la Búsqueda Distribuida	62
4.3	Proceso de Consulta Distribuida	65
4.3.1	Acceso a la Interfaz de Consulta	65
4.3.2	Formulación de la Consulta	66
4.3.3	Proceso de Consulta	67
4.3.4	Proceso de Extracción de Datos	68
4.3.5	Resultados de la Consulta Distribuida	69
5	Pruebas	71
5.1	Objetivos de las pruebas	71
5.2	Descripción del Ambientes de Pruebas	72
5.3	Casos de Prueba.	75
5.3.1	Prueba No. 1.	75
5.3.2	Prueba No. 2.	78
5.3.3	Prueba No. 3.	80
5.4	Otras Pruebas.	80

6 Conclusión y discusión **83**
6.1 Conclusión 83
6.2 Alcances Logrados 85
6.3 Trabajos Futuros 86

Bibliografía **88**

Lista de Figuras

1.1	Arquitectura de herramienta de acceso a base de datos.	3
1.2	Arquitectura de herramienta de acceso a multibases de datos.	4
2.1	Arquitectura Cliente/Servidor.	12
2.2	Arquitectura Peer to Peer.	13
2.3	Ciclo de vida del modelo en Cascada.	19
3.1	Componentes de la Metabase de Datos.	21
3.2	Componentes de la Búsqueda Distribuida.	24
3.3	Componente de Comunicaciones.	26
3.4	Subcomponente de Búsqueda Local.	27
3.5	Diagrama de flujo de la función Analizador Léxico.	30
3.6	Diagrama de flujo de la función Construcción de la Instrucción SQL.	30
3.7	Diagrama de flujo que muestra el proceso para la extracción de los metadatos.	31
3.8	Subcomponente de Búsqueda Remota.	32
3.9	Ejemplo de la matriz de memoria que almacena los metadatos de los recursos que el usuario del sistema desea consultar.	35
3.10	Componentes del Proceso de Consulta Distribuida.	36
3.11	Diagrama de flujo que muestra el proceso para la creación de un conjunto de estaciones que tienen un recurso en común.	37
3.12	Ejemplo de una matriz de metadatos.	38
3.13	Secciones de la Interfaz de Consulta.	39
3.14	Diagrama de flujo que muestra el proceso de creación de la lista de estaciones.	41
3.15	Diagrama de flujo que muestra el proceso para la extracción de datos.	42
3.16	Ejemplo del diseño del archivo de texto para la presentación de resultados.	43
3.17	Ejemplo del diseño del archivo gráfico para la presentación de resultados.	44
4.1	Línea de comando utilizada para la creación de la base de datos DBmetadatos.	47
4.2	Línea de comando utilizada en la creación de la relación stations.	48
4.3	Línea de comando utilizada para la creación de la relación resource.	48
4.4	Línea de comando utilizada en la creación de la relación DataResource.	49
4.5	Línea de comando utilizada en la creación de la relación TempDataResource.	51
4.6	Interfaz de Búsqueda de Recursos Distribuidos sobre Internet.	51
4.7	Interfaz para el registro de Estaciones Mareográficas.	52
4.8	Interfaz para el registro de Diccionario de Recursos.	53
4.9	Interfaz para el registro de Compartición de Recursos Oceanográficos.	53

4.10	Funciones del Intermediario Web.	54
4.11	Fragmento de código que ejemplifica el paso de parámetros de solicitud de búsqueda local al componente Analizador Léxico.	56
4.12	Fragmento de código que ejemplifica la detección de estaciones.	57
4.13	Fragmento de código que ejemplifica la detección de recursos.	58
4.14	Fragmento de código que ejemplifica la construcción de la cadena SQL.	59
4.15	Fragmento de código que ejemplifica la ejecución de la cadena SQL.	60
4.16	Fragmento de código que ejemplifica el almacenamiento temporal de metadatos.	61
4.17	Fragmento de código que ejemplifica el paso de parámetros de solicitud de búsqueda remota al componente AnalizadorLexico.	62
4.18	Interfaz Web que muestra los metadatos que se obtuvieron del proceso de Búsqueda de Recursos Distribuidos.	63
4.19	Interfaz Web de la estación mareográfica de Tampico, Tamaulipas.	64
4.20	Fragmento de código que ejemplifica la creación de un conjunto de estaciones que tienen un recurso en común.	65
4.21	Interfaz Web para la Consulta de Recursos Distribuidos.	66
4.22	Fragmento de código que ejemplifica el proceso para la creación del formato para la lista de estaciones.	67
4.23	Fragmento de código que ejemplifica el Proceso de Extracción de Datos.	68
4.24	Interfaz Web que muestra en forma de gráfica los datos extraídos de la estación mareográfica de Tampico Tamaulipas.	69
4.25	Interfaz Web que muestra en forma de texto los datos extraídos de la estación mareográfica de Tampico Tamaulipas.	70
5.1	Ambiente de pruebas número Uno.	73
5.2	Ambiente de pruebas número Dos.	74
5.3	Ambiente de pruebas número Tres.	74
5.4	Interfaz de Búsqueda de Recursos Distribuidos sobre Internet.	75
5.5	Interfaz que muestra los resultados de la Búsqueda de Recursos Distribuidos sobre Internet.	76
5.6	Interfaz Web para la Consulta de Recursos Distribuidos.	77
5.7	Gráfica que muestra los datos extraídos de las estaciones de Mezquital, la Pesca y Tampico.	77
5.8	Interfaz que muestra los resultados de la Búsqueda de Recursos Distribuidos sobre Internet.	78
5.9	Gráfica que muestra los datos extraídos de las estaciones de Mezquital, la Pesca, Tampico y Veracruz.	79
5.10	Interfaz que muestra los resultados de la Búsqueda de Recursos Distribuidos sobre Internet.	81
5.11	Gráfica que muestra los datos extraídos de las estaciones de Mezquital, la Pesca, Tampico, Tuxpan y Veracruz.	82

Lista de Tablas

3.1	Ejemplo de una relación de metadatos para la descripción de los recursos almacenados en un nodo de la comunidad.	22
3.2	Ejemplo de lista de claves y descripción de recursos.	34
3.3	Relación de la matriz de metadatos M	38
4.1	Esquema de metadatos tomados del estandar FGDC, los cuales fueron seleccionados para la relación de metadatos.	46
4.2	Estructura de la relación stations.	47
4.3	Estructura de la relación resource.	48
4.4	Estructura de la relación DataResource.	49
4.5	Estructura de la relación TempDataResource.	50

Resumen

En éste trabajo se obtuvo una arquitectura y una herramienta de software para la búsqueda y consulta de recursos distribuidos sobre Internet. El objetivo principal de ésta herramienta es extraer los recursos compartidos por diferentes organizaciones a través de Internet, donde cada organización se considera como un nodo. La creación de la arquitectura de software incluyó el diseño y desarrollo de las siguientes 5 fases principales: un portal Web de búsqueda y uno de consulta, la metabase de datos, el proceso de búsqueda distribuida y el proceso de consulta distribuida. El portal Web de búsqueda permite al usuario realizar por medio de atributos de recursos oceanográficos la búsqueda de recursos distribuidos a través de Internet. La metabase de datos cumple con el objetivo de almacenar los metadatos de los recursos que se desean compartir por una organización. El diseño del esquema de metadatos está basado en el estandar de metadatos FGDC, el cual se implementó posteriormente en una base de datos. El proceso de búsqueda distribuida se encarga de propagar la solicitud de búsqueda del usuario y extraer los metadatos de los recursos desde los nodos. Se utilizó el sistema peer-to-peer Mutella como motor de transferencia de recursos distribuidos mediante el protocolo de comunicación Gnutella. El proceso de consulta distribuida se encarga de extraer los datos de los recursos localizados por el proceso de búsqueda distribuida de los nodos. Se diseñaron y desarrollaron interfaces Web, para la consulta de recursos distribuidos, para habilitar el mecanismo de extracción remota de datos, y para la presentación de los resultados al usuario.

El desarrollo de la herramienta demostró ser un medio de consulta de datos oceanográficos útil para la sociedad en general, ya que tiene el potencial de ayudar tanto en las actividades de investigación como en las actividades operacionales que resguardan la seguridad en los puertos y la protección civil próximas a la costa, facilitando el acceso a la información del clima oceánico en tiempo aproximado al real a través de Internet. A manera de ejemplo, se realizaron diversas pruebas con datos extraídos de las estaciones mareográficas ubicadas en Tampico, Mezquital y la Pesca Tamaulipas, administradas por la Secretaría de Marina, verificando la eficiencia del sistema propuesto.

Abstract

The present work shows the architecture and the software tool developed to perform both distributed search and distributed consultation of resources on Internet. The main objective of this tool is to extract the resources shared by different organizations through Internet, where each organization is considered as a node. The creation of the software architecture includes the design and development of the next 5 main phases: a web searching portal and web consultation, a data metabase, distributed search process and distributed consultation. The web search portal allows user to carry out searching of distributed resources through Internet by means of attributes of oceanographic resources. The data metabase fulfills the objective of storing metadatas of the shared resources from an organization. The design of the metadata scheme is based on the metadata standard FGDC, which was implemented later as a database. The process of distributed search will propagate the search query the application of the user and extracts the metadatas of the resources from the nodes. The Mutella peer-to-peer system has been used as the transfer engine motor of the distributed resources using Gnutella as communication protocol. The process of distributed consultation extracts the data of the resources located by the process of distributed search of the nodes. The following web interface were designed: web for consultation of distributed resources, remote extraction of data, and results presentation to user's.

The developed tool has demonstrated to be a useful medium for consultation of oceanographic resources. It can be use not only for scientific studies but as other mechanism to take safe decisions relating operations in port, civil protection and others, taking into a account the near real time ocean to the data that can be achieved with this tool. As an example, it was carried out a real tests by extracting data from tidal stations, which are located in Tampico, Mezquital and La Pesca Tamaulipas, all of them were carry out by Secretaría de Marina and it was verified the efficiency of the proposed system.

Glosario

Actividad intermitente de los nodos. Proceso en donde las conexiones de los nodos ROPVO se interrumpen, o cesan y luego continúan o se repiten.

Base de Datos. Es un conjunto de datos operacionales almacenados, que pueden ser usados por aplicaciones de sistemas de alguna empresa en particular.

Carácter. Cualquier signo tipográfico. Puede ser una letra, un número, un signo.

Componente. Parte discreta de un sistema capaz de operar independientemente, pero diseñada, construida y operada como parte integral del sistema.

Comunidad ROPVO. Conjunto de organizaciones ó instituciones que comparte el interés de compartir recursos e información oceanográfica a través de internet.

DBI. La interfaz a base de datos (DBI, por sus siglas en inglés), es un conjunto de funciones que permiten a los programas realizados con el lenguaje de programación perl establecer conexión a diferentes bases de datos y ejecutar sentencias SQL.

Función. Es el término para describir una secuencia de órdenes que hacen una tarea específica de una aplicación más grande.

Interfaz. Término usado en los conceptos de hardware y software que permite la conexión entre dos o más elemento de la misma naturaleza.

JDBC. La Conectividad de Bases de Datos Java (JDBC, por sus siglas en inglés) es un conjunto de funciones que permiten a los programas Java establecer conexión a diferentes bases de datos y ejecutar sentencias SQL. JDBC permite a los programas Java interactuar con la mayoría de las base de datos compilada en SQL. Actualmente la mayoría de los sistemas manejadores de bases de datos relacional

soportan SQL. JDBC fue desarrollado por JavaSoft que es una sucursal de Sun Microsystems.

Metadato. Los metadatos son datos altamente estructurados que describen información de la información o datos sobre los datos. Los metadatos describen quién, cómo y cuando han sido obtenidos un conjunto de datos en particular. Los metadatos son esenciales para entender la información almacenada en un Sistema de Información.

Multitarea. Facultad de ejecutar más de una tarea a la vez en uno o varios procesadores.

MySQLConnect. Establece una conexión a un servidor MySQL. Todos los argumentos son opcionales, y si no hay, se asumen los valores por defecto.

Nodo. Una computadora, unión o punto de conexión en una red. Puede referirse a un solo dispositivo, es decir, un interruptor limitador particular.

Proceso. Conjunto de actividades realizadas en forma secuencial que permiten transformar uno o más insumos en un producto o servicio.

Recurso. es un dato o conjunto de datos, página Web, imágenes, etc.

Relación. Es el término matemático dado para una tabla. Una relación esta constituida de atributos, dominios y tuplas. Una relación de dominios $D_1, D_2, .. D_n$ (no necesariamente todos distintos) consiste de un encabezado y de un cuerpo.

Servent. Aplicación de software que realiza la función de cliente y servidor simultáneamente.

Socket. Interfaz que ofrece un mecanismo de comunicación general entre dos procesos cualquiera que pertenezcan a un mismo sistema o ha dos sistemas diferentes.

Tubería. Son conexiones entre procesos. La salida de un proceso se encadena con la entrada de otro, esto permite procesar los datos en una sola línea de comandos.

Capítulo 1

Introducción

Hoy en día los sistemas de información proporcionan la comunicación y poder de análisis que muchas empresas e instituciones requieren para llevar a cabo la administración de la información a una escala global. Prueba de ello, es el creciente interés que se ha observado en importantes organizaciones alrededor del mundo como: la Red de Datos Marinos en el Mediterráneo (MEDAR, por sus siglas en inglés), el Directorio de Datos Marinos (EDMED, por sus siglas en inglés), la Red de Observaciones Oceánicas en las Costas de Texas (TCOON, por sus siglas en inglés) y en particular la Red de Observaciones y Predicciones de Variables Oceánicas (ROPVO) en el Golfo de México (GM) entre otros [1].

En la actualidad se están realizando esfuerzos por administrar y compartir recursos oceanográficos entre TCOON y ROPVO-GM. TCOON es una infraestructura compuesta de mareógrafos y satélites, computadoras y el Sistema de Información Basado en Web (SIBW), Pharos. TCOON fué construido y es operado por el Conrad Blucher Institute Division of Nearshore Research (CBI-DNR) en la Texas A&M University-Corpus Christi (TAMUCC). TCOON está integrado por 42 plataformas para la recolección de datos de nivel del mar y meteorológicos que se mantienen bajo los estándares del Servicio Oceánico Nacional (NOAA, por sus siglas en inglés)[2]. El sistema Pharos es una herramienta de software que permite poner a disposición del público en general datos oceanográficos en tiempo aproximado al real [3].

Por otra parte ROPVO es una infraestructura en desarrollo que está compuesta de mareógrafos, equipo

modem celular y computadoras. Actualmente los datos recolectados por sus mareógrafos son administrados con el sistema Pharos donado por TAMUCC. Las organizaciones que participan en el proyecto ROPVO son el Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada Unidad Altamira del Instituto Politecnico Nacional (CICATAUA-IPN), TAMUCC, Universidad Nacional Autónoma de México (UNAM), Secretaría de Marina (SEMAR) y Servicio Meteorológico Nacional (SMN) a los cuales nos referiremos en este documento como la *comunidad ROPVO*. Cada uno de los organismos cuenta con sus propios recursos oceanográficos que son bases de datos, archivos de texto, imágenes, gráficas, sonido y video. Es importante aclarar que el proyecto ROPVO-GM es financiado por el Consejo Nacional de Ciencias y Tecnología (CONACyT) a través del CICATAUA-IPN en Altamira, Tamaulipas, realizado por E. Meza Conde et al. [4]. Este trabajo de tesis forma parte del proyecto ROPVO-GM. El objetivo de este trabajo fue desarrollar e implementar una arquitectura de software para la búsqueda y consulta de recursos oceanográficos distribuidos en la comunidad ROPVO, que se incorporará al proyecto ROPVO-GM y que servirá como motor de transferencia de recursos distribuidos a través de Internet mediante el uso de metadatos de los recursos oceanográficos.

Nota: Dada la naturaleza del tema y del proyecto de investigación, en éste manuscrito se emplean muchos términos en inglés, algunos de ellos traducidos, pero la mayor parte formarán parte del texto sin traducción alguna.

1.1 Estado del Arte

La primera Herramienta de Consulta Basada en Ejemplos (QBE, por sus siglas en inglés) para una Base de Datos en Internet fué realizada por F. Rasgado [5]. Esta herramienta permite consultar bases de datos generadas con el Sistema Manejador de Bases de Datos Distribuidas (SiMBaDD). Una de las razones por la que las bases de datos son implementadas con SiMBaDD se debe a que la herramienta debe presentar al usuario información detallada de la definición de la base de datos, además de información adicional, para que el usuario seleccione las tablas y columnas que le pueden servir para formular su consulta. La

información adicional se refiere a datos como unidades de medida, formato de fecha y descripción de las columnas y de las tablas.

La arquitectura de esta herramienta, figura 1.1, incluye el uso de un servidor intermediario, un servidor de bases de datos, un servidor Web y un applet que es una interfaz Web. En este caso la función principal del intermediario es reexpedir la consulta definida en la interfaz QBE hacia el SiMBaDD. Si existe un resultado, éste es enviado al intermediario, quien a su vez lo reenvía al applet. Otro motivo de la existencia del intermediario es para resolver el problema de la conexión con bases de datos que no están en la misma máquina en la que reside el servidor Web de donde se descargó el applet de la interfaz Web. Sin el servidor intermediario las bases de datos no podrían ser consultadas por las restricciones de seguridad de los applets.

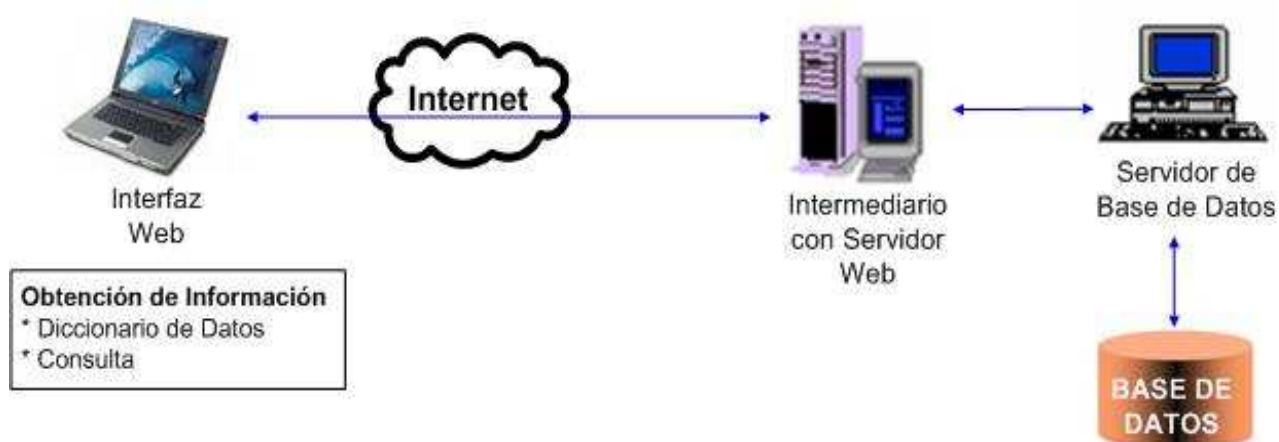


Figura 1.1: Arquitectura de herramienta de acceso a base de datos.

Continuando con los estudios relacionados en el lenguaje de QBE en el año 2000, se diseñó una Herramienta QBE para Múltibases de Datos en Internet realizado por May Arrijoja [6]. Esta herramienta permite que usuarios inexpertos, consulten varias bases de datos al mismo tiempo. Esta versión permite realizar conversiones de unidades de medida y de formatos de fechas, además de que la interfaz le permite al usuario escoger entre las tablas de las distintas Bases de Datos que le interesen para formular su consulta.

A diferencia del intermediario anterior escrito en C para Unix, éste fue escrito completamente en Java

y provee cuatro servicios fundamentales: obtención de metadatos, obtención de unidades de conversión, ejecución de la consulta global y conversión de unidad de medida y de formato de fecha. Además soporta las peticiones de varios clientes, en otra palabras es multitarea. Un solo intermediario brinda servicio a más de una interfaz QBE. La arquitectura general de la herramienta, presentada en la figura 1.2, considera tres componentes principales que son: la Interfaz QBE, Intermediario y Sistema Manejador de Bases de Datos (SMBS). La arquitectura muestra la creación de hilos para atender a los diferentes usuarios de la herramienta, muestra también cuáles son las peticiones de servicio que la interfaz QBE solicita al intermediario, indica que las comunicaciones con los SMBD se realizan vía JDBC, y que en el intermediario está instalado el SMBD comercial MS SQL Server con una base de datos cuyo nombre es Intermediario. La herramienta utiliza la base de datos Intermediario para almacenar los resultados de las subconsultas y realizar la consulta final.

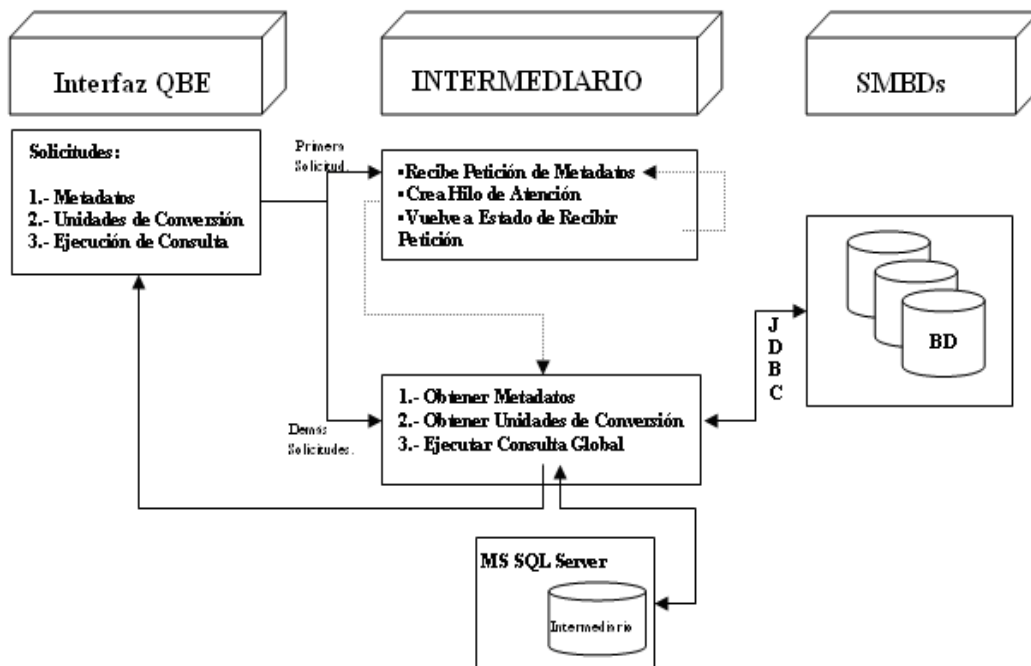


Figura 1.2: Arquitectura de herramienta de acceso a multibases de datos.

La herramienta cuenta con un mecanismo para traducir la consulta definida gráficamente por el usuario en lenguaje QBE a SQL, que se envía al intermediario y los resultados de la consulta son enviados al usuario. Una parte fundamental de la herramienta es el uso de metadatos. Dado el desconocimiento del esquema

de las bases de datos por parte de los usuarios, es necesario presentarles la mayor cantidad de información acerca de esos esquemas, de aquí la importancia del uso de metadatos, ya que éstos describen las tablas y columnas de las bases de datos. El trabajo descrito por A. May [6] sólo permite consultar bases de datos distribuidas, en tanto que nuestro trabajo permite realizar búsquedas y consultas de recursos distribuidos.

Se observa que tanto la herramienta de software distribuido realizado por F. Rasgado y A. May se enfocan al acceso a bases de datos distribuidos a través de Internet. La diferencia que existe entre los proyectos realizados por ellos dos y éste trabajo de tesis radica en que la arquitectura de software desarrollada se enfoca a realizar búsquedas y consultas de recursos distribuidos a través de Internet. Otra diferencia notable es en la forma de emplear los metadatos. Los proyecto realizado por F. Rasgado y A. May utiliza los metadatos para describir el contenido de los campos de las tablas de las bases de datos, mientras que en éste trabajo de tesis, los metadatos se utilizan para describir recursos oceanográficos distribuidos en la comunidad ROPVO.

1.2 Justificación

Los avances en la tecnología de la comunicación, las redes de computo, el almacenamiento y acceso distribuidos de datos están revolucionando internacionalmente a las interacciones científicas[]. Instituciones importantes en Estados Unidos como el Centro Nacional de Datos Oceanográficos (NODS), El Centro Mundial de datos para la Oceanografía, y el Centro para el Análisis de Información sobre Bióxido de Carbono y TCOON actualmente estan compartiendo sus recursos a través de Internet. Por otra parte, en México se están realizando esfuerzos por recopilar y compartir recursos entre instituciones de la comunidad ROPVO, sin embargo, no se está llevando a cabo estrategias que permitan la compartición de recursos entre instituciones. Este trabajo de investigación propone estandarizar los procesos de almacenamiento de los recursos de las instituciones en bases de datos. Esto permitirá diseñar e implementar una arquitectura de software que permitirá realizar búsquedas y consultas de recursos distribuidos en la comunidad y que estará disponible para el medio científico y al público en general a través de Internet. En estos momentos se están enfocando los esfuerzos en la recopilación de datos del nivel del mar y datos meteorológicos a lo

largo de la costa y de los puertos del Golfo de México. Estos datos serán administrados por diferentes nodos ROPVO-GM. Posteriormente se incorporará el Centro de Información TCOON, esto ampliará los márgenes de información de datos y fomentará el intercambio de ideas y apoyo mutuo entre naciones.

1.3 Planteamiento del problema

El Internet se ha convertido en uno de los medios de comunicación más usado en el mundo que permite compartir información. La comunidad virtual ROPVO es una comunidad que realizará la administración de los recursos oceanográficos mediante el SIBW-Pharos. Actualmente el sistema Pharos permite realizar búsquedas y consultas de datos oceanográficos en una sola computadora. Sin embargo la comunidad ROPVO desea compartir recursos oceanográficos almacenados en diferentes computadoras conectadas a través de Internet, y Pharos tiene el problema de que no cuenta con un mecanismo que permita realizar búsquedas y consultas distribuidas de recursos oceanográficos sobre Internet. Como resultado de este trabajo se agrega al sistema Pharos la implementación de la ASBUCORDI.

Cada uno de los integrantes de la comunidad ROPVO es una organización independiente, por lo que hay que establecer un mecanismo adecuado de *Búsqueda* y *Consulta* entre los organismos de la comunidad ROPVO. Esto es, se debe considerar que las organizaciones de la comunidad virtual se agregan y remueven constantemente, por lo cual el mecanismo de búsqueda y consulta debe poder adaptarse a esta dinámica. Además se deben de tomar en cuenta los problemas de comunicación sobre Internet que se pueden presentar al realizar la búsqueda y consulta de recursos oceanográficos distribuidos. De esta manera, se hace necesario que la comunidad ROPVO cuente con una estructura lógica, que se lleva a cabo mediante el diseño de un esquema de metadatos, para la organización distribuida de sus recursos oceanográficos. Lo anterior plantea el problema de realizar un análisis que nos permita organizar los recursos oceanográficos de la comunidad ROPVO bajo un esquema de metadatos, que pueda ser útil para el mecanismo de búsqueda y consulta distribuida de recursos. Una vez diseñado el esquema de metadatos, se procederá a introducir estos en una base de datos que llamaremos de ahora en adelante Metabase de Datos.

Una vez organizados los metadatos de los recursos compartidos en una metabase de datos, un mecanismo de búsqueda distribuida requiere resolver el problema de encontrar a través de los metadatos los recursos solicitados a la comunidad virtual ROPVO. Esto es, a través de un mecanismo de consulta de recursos oceanográficos se deberá permitir a los usuarios consultar los recursos oceanográficos localizados por el mecanismo de búsqueda. Para ello, el mecanismo de consulta deberá estar disponible a través de Internet y deberá permitir que usuarios inexpertos puedan consultar los recursos oceanográficos a través del Web. Esto es, se debe permitir a los usuarios consultar recursos oceanográficos, de una manera transparente [7], almacenando los metadatos de los recursos en metabases de datos distribuidas en la comunidad ROPVO. Con respecto a los recursos de bases de datos administradas por el sistema Pharos, sus resultados deberán ser presentados al usuario a través de la Web, en forma de texto, o en gráficas.

1.4 Hipótesis

Con base en una arquitectura peer-to-peer es factible diseñar e implementar un mecanismo de software que permita la búsqueda y consulta de recursos oceanográficos distribuidos en una comunidad virtual, utilizando una herramientas de software que sean totalmente libre y metabases de datos como esquemas de conectividad de datos.

1.5 Objetivo general

Desarrollar e implementar una arquitectura de software para la búsqueda y consulta de recursos oceanográficos distribuidos que se incorporará al proyecto ROPVO-GM y que servirá como motor de transferencia de recursos a través de Internet mediante el uso de metadatos.

1.6 Objetivos específicos

Los objetivos específicos del proyecto de investigación consisten en el diseño e implementación de:

- Una metabase de datos distribuida donde se almacenen los metadatos de los recursos.
- Un programa que implemente el protocolo Gnutella [8] para la búsqueda de recursos distribuidos a través de metadatos.
- Un portal Web que permita al usuario realizar, por medio de atributos de recursos oceanográficos, búsquedas de recursos distribuidos a través del Internet.
- Un programa servidor tipo Servent que coordine el procesamiento de consultas distribuidas e integre los resultados.
- Un portal Web para la presentación de los resultados de consultas distribuidas en forma de texto o gráficas.
- Integrar el mecanismo de búsqueda y consulta distribuida de recursos oceanográficos en Pharos.
- Realizar pruebas sobre Internet

1.7 Organización del Documento

Este manuscrito se encuentra dividido en cinco capítulos, cuyo contenido se describe a continuación.

En el capítulo 2 se describe el marco teórico relacionado con el proyecto y que sirve de soporte para su desarrollo. Se describe qué es un Sistema Distribuido, la Arquitectura Cliente/Servidor [9] y la Arquitectura Peer-to-Peer [10]. Igualmente describe el sistema Peer-to-Peer Mutella y conceptos relacionados con el Internet. Finalmente se presentan los diferentes tipos de estándares de esquemas de metadatos y los protocolos de comunicación.

El capítulo 3 muestra la metodología de la Arquitectura de Software propuesta, el diseño de la Metabase de Datos y además se describen los procesos de Búsqueda y Consulta Distribuida de recursos a través de Internet.

El capítulo 4 muestra los resultados del diseño de la arquitectura de software, la definición del esquema de metadatos, los métodos para la creación de la Metabase de Datos, el Diccionario de Términos, el Depósito de Metadatos y el Depósito Temporal de Metadatos. Igualmente se presenta el diseño de la Interfaz de Búsqueda y el diseño del Intermediario Web con las líneas de código que realizan el proceso Búsqueda Local, Analizador Léxico y propagación de la solicitud de búsqueda. Finalmente se muestra el diseño de la Interfaz de Consulta con las líneas de código que realizan el procesamiento de la Consulta y Extracción de Datos.

En el capítulo 5 se describen las pruebas realizadas a la implementación de la arquitectura de software propuesta. Los objetivos de las pruebas, el ambiente de pruebas y los casos de pruebas realizados.

Capítulo 2

Marco Teórico

2.1 Arquitectura de Software

En la última década cambió la visión que los desarrolladores tienen sobre la estructura de un sistemas de software. Esta nueva visión se llamó arquitectura. Desde los pequeños programas hasta los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables según su arquitectura. Este nuevo aspecto hace posible el estudio de sistemas ya implementados así como el desarrollo de nuevos sistemas. El tema arquitectura de software es aún nuevo en nuestros días ya que no se han establecido bases sólidas en su modo de uso y existen diferentes definiciones de arquitectura de software. De ahora en adelante nos referiremos a arquitectura de software como *AS*. Una definición reconocida es la de Clements [12], quien la define como una vista del sistema que incluye los componetes principales del mismo y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. Un componente es un conjunto de procesos. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión de los detalles.

La *AS* es una forma de diseño de software que se manifiesta en el proceso de creación de un sistema. Mary Shaw y David Garlan [13] sugieren que las estructuras de *AS* deben incluir una organización a grandes razgos y una estructura global de control, protocolos para la comunicación, la sincronización y el acceso a datos, la asignación de funcionalidad a elementos del diseño, la distribución física, la composición

de los elementos de diseño, escalabilidad, rendimiento y selección entre alternativas de diseño. Después, David Garlan y Mary Shaw [14] clasifican a la *AS* en cinco modelos: modelos estructurales, modelos de framework, modelos dinámicos, modelos de proceso y modelos funcionales.

En este trabajo utilizaremos el modelo estructural para describir el diseño de nuestra Arquitectura de Software para la Búsqueda y Consulta de Recursos Distribuidos sobre Internet. La Arquitectura de Software a diseñar empleará componentes de mecanismos locales y componentes de mecanismos distribuidos sobre Internet.

2.2 Sistemas Distribuidos

Un sistema distribuido es aquél al que sus usuarios ven como un ordinario sistema centralizado; sin embargo, se ejecuta en diferentes e independientes CPUs [15]. En un sistema distribuido los componentes de software y hardware locales se comunican y coordinan sus acciones solamente por el envío de mensajes. Las computadoras conectadas a esta red, pueden estar espacialmente separadas por alguna distancia, por ejemplo las computadoras pueden estar en continentes diferentes, pertenecer a una institución en particular o estar en el mismo cuarto. Un sistema distribuido conlleva las consecuencias de ser concurrente, manejar algún tiempo global y fallas de independencia. A continuación presentamos dos modelos de arquitectura para el diseño de sistemas distribuidos.

2.2.1 Arquitectura Cliente/Servidor

El término cliente/servidor fue primeramente usado en los años 80 para referirse a las computadoras personales en la red y empezó a tener aceptación a finales de la década de los 80, ver figura 2.1. La arquitectura cliente/servidor es una infraestructura versátil y distribuida, basada en mensajes que está pensada para mejorar la flexibilidad, interoperabilidad y escalabilidad de los sistemas [16]. Un *cliente* se define como un

solicitador de servicios y un *servidor* se define como el proveedor de servicios [17].



Figura 2.1: Arquitectura Cliente/Servidor.

2.2.2 Arquitecturaa Peer-to-Peer

Los sistemas Peer-to-Peer(P2P) se refieren a una clase de sistemas distribuidos que funcionan de una manera descentralizada. Clay Shirky el autor del artículo "What is P2P and What is not" define una red Peer-to-Peer como un sistema distribuido de aplicaciones que toma ventajas de los recursos, almacenamiento, ciclos, contenido y presencia humana disponible a través de Internet [18], se refiere a una red que no tiene clientes ni servidores fijos, sino una serie de nodos que se comportan simultáneamente como clientes y como servidores de otros nodos conectados a la red.

La tecnología Peer-to-Peer (P2P) ha ganando terreno en los últimos años [19]. P2P llamó la atención cuando apareció el programa Napster [20] que permitía compartir archivos musicales. Aunque no era una aplicación P2P pura, éste, cumple con algunas de las características de los sistemas Peer-to-Peer. Con la misma idea de compartir archivos musicales, aparecieron otras aplicaciones como Gnutella [21], eDonkey [22], y eMule [23]. Existen instituciones que han enfocado la tecnología P2P a otras áreas de investigación, por ejemplo en las ciencias, los proyectos SETI@Home [24], Folding@Home [25], Genome@Home [26] y Grid project [27], que han mostrado cómo la tecnología P2P es muy útil para resolver problemas complejos, costosos y de carácter distribuido.

Se puede entender que un peer es un participante en un sistema P2P. Estos participantes están activos en una aplicación, de manera que pueden proporcionar servicios a otros participantes en el grupo. Una ventaja comparativa con respecto a otros sistemas clásicos, es que los archivos o los procesos están completamente descentralizados. Una de las metas de la tecnología P2P es tomar ventaja de los recursos que existen en la red, ver figura 2.2.

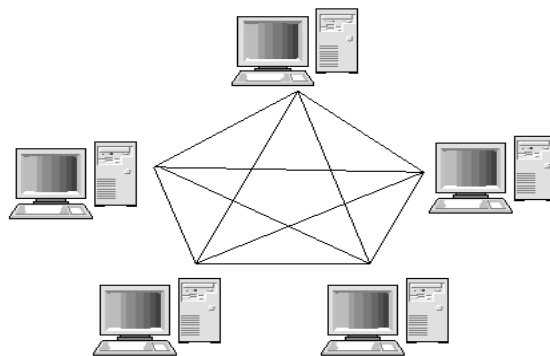


Figura 2.2: Arquitectura Peer to Peer.

2.3 Protocolos de comunicación

Respecto a los estudios realizados sobre los protocolos de comunicaciones de Internet podemos mencionar el Protocolo de Internet (IP, por sus siglas en inglés), que fue diseñado para interconectar sistemas basados en redes de intercambio de paquetes. El protocolo IP permite el intercambio de bloques de datos, denominados datagramas, entre nodos conectados a una red, cada uno de ellos tiene una dirección única denominada dirección IP. La función de IP es encaminar datagramas a través de una serie de redes interconectadas. Este proceso se lleva a cabo pasando los datagramas desde un módulo Internet hacia otro, hasta alcanzar el destino deseado. Se entiende por módulo Internet aquellos servicios que lleva a cabo la capa de red dentro de una arquitectura TCP/IP. Otro protocolo de Internet es el Protocolo de Control de Transmisión (TCP, por sus siglas en inglés). El protocolo TCP es el método más eficiente y seguro para el tráfico de red entre un cliente y un servidor o entre subredes en general. Aunque TCP se presenta como

parte del grupo de protocolos de Internet (TCP/IP), es un protocolo de propósito general que se puede adaptar para utilizarlo con otros sistemas de entrega. Las conexiones TCP funcionan de una forma muy parecida a como lo hacen las conexiones vía telefónica: el que está a un lado de la línea inicia una comunicación y ésta debe ser aceptada por el usuario que se encuentra al otro lado. Cuando un cliente decide establecer una comunicación con un servidor, es necesario que ambos estén de acuerdo en participar, de lo contrario la comunicación no se podrá llevar a cabo. Una conexión TCP viene identificada por una pareja de sockets, es decir, una dirección IP y un número de puerto en cada extremo. La ventaja de este método es que un único nodo es capaz de mantener diferentes conexiones TCP a través de un mismo puerto. Esto es posible debido a que los paquetes que recibe el nodo se diferencian uno de otros porque utilizan sockets distintos. Por ejemplo: un servidor Telnet permite que diferentes clientes accedan al puerto 23, que es el puerto estándar para este servicio. Todos los clientes se conectan al mismo socket (dirección IP del servidor + puerto 23), sin embargo, el servidor es capaz de distinguir a los diferentes clientes debido a que cada uno de ellos utiliza un socket local diferente.

2.3.1 Sistema Mutella

El sistema Mutella es un software tipo peer-to-peer utilizado para el intercambio de archivos de texto, sonido, programas, imágenes y videos. El sistema P2P Mutella se utilizará como motor de búsqueda de recursos distribuidos en la implementación de nuestra Arquitectura de Software. Este protocolo inicialmente fue desarrollado por Max [31]. El sistema Mutella tiene la ventaja de ser un software con código fuente libre (open source software); esto quiere decir que se permite descargar el código, estudiarlo y realizar las modificaciones pertinentes para adaptarlo a nuestras necesidades. El código fuente puede ser descargado desde su página oficial <http://mutella.sourceforge.net>. La tarea del sistema P2P Mutella es propagar la solicitud de búsqueda de recursos en la comunidad ROPVO.

2.3.2 Especificaciones del protocolo Gnutella V0.4

Gnutella es un protocolo para búsquedas distribuidas. Aunque el protocolo Gnutella soporta un paradigma tradicional de búsqueda de clientes/servidores centralizados, Gnutella se distingue en ser un modelo Peer-to-Peer descentralizado. En este modelo, cada nodo es un servidor y cliente a la vez, a esta característica de Gnutella es conocida como *Servents*. Los *Servents* proveen interfaces para los usuarios, puede consultar y ver los resultados de búsqueda, mientras que al mismo tiempo pueden aceptar consultas de otros *Servents* comparando un conjunto de datos locales respondiendo favorablemente si la comparación es verdadera. Debido a su naturaleza distribuida, una red de *Servents* que implemente el protocolo Gnutella es altamente tolerante a fallo, las operaciones de una red no será interrumpida si un subconjunto de *Servents* queda fuera de servicio [30].

2.3.3 JDBC

La Conectividad de Bases de Datos Java (JDBC, por sus siglas en inglés) es un conjunto de funciones que permiten a los programas Java establecer conexión a diferentes bases de datos y ejecutar sentencias SQL [36]. JDBC permite a los programas Java interactuar con la mayoría de las base de datos compilada en SQL. Actualmente la mayoría de los sistemas manejadores de bases de datos relacional soportan SQL. JDBC fue desarrollado por JavaSoft que es una sucursal de Sun Microsystems.

2.4 El Internet

El Internet es un sistema distribuido con una basta colección de redes de computadoras de diferentes tipos interconectadas entre sí. La interacción de los programas en ejecución de las computadoras conectadas a la red se logra por el paso de mensajes, empleando una característica común de comunicación. El diseño y construcción de los mecanismos de comunicación de Internet llamados también protocolos Internet es la principal técnica que se ha logrado para dirigir mensajes permitiendo la ejecución de un mensaje de un programa en cualquier parte a cualquier otro programa. El Internet es también un sistema distribuido que

permite a los usuarios, hacer uso de servicios tales como el WWW, email y transferencia de archivos. El conjunto de servicios es finalmente abierto, éste se puede extender para añadir nuevos tipos de servicios y computadoras servidoras [32].

2.4.1 La Word Wide Web (WWW)

La WWW es un sistema en evolución que permite acceder a servicios y recursos publicados a través del Internet [28]. A través del browser para la web, tal como Netscape e Internet Explorer, los usuarios pueden ver y recuperar diferentes tipos de documentos, escuchar audio, ver video e interactuar con un ilimitado conjunto de servicios. La Web está basada principalmente en tres componentes de tecnología estandar que son el Lenguaje de Marcado de Hipertexto (HTML, por sus siglas en inglés), el Localizador de Recursos Uniformes (URL, por sus siglas en inglés) y una arquitectura de sistema.

2.4.2 El lenguaje HTML y localizar URL

El HTML es un lenguaje estándar con el que se escriben los programas (documentos) en el World Wide Web. El HTML define formatos, vínculos y otros aspectos de disposición de textos, imágenes y objetos dentro de una página Web [29]. El lenguaje HTML especifica el diseño y contenido de las páginas, esto es, la manera en que las páginas serán desplegadas por los exploradores Web. Una página Web contiene encabezados, párrafos, tablas e imágenes. El URL identifica los documentos y otros recursos almacenados como parte de la Web, esto es, el propósito del URL es permitir al explorador Web la localización de estos recursos. La arquitectura de sistema cliente/servidor cuenta con reglas bien definidas para la interacción entre aplicaciones, un ejemplo es el Protocolo de Transferencia de Hipertexto (HTTP, por sus siglas en inglés) mediante el cual los exploradores Web y otras aplicaciones obtienen documentos y otros recursos de los servidores Web.

2.4.3 Sistemas de Información Basados en Web (SIBW)

Actualmente está emergiendo una nueva tecnología para la administración de la información sobre Internet que son los sistemas de información basados en Web (SIBW) [33]. Los SIBW tienen las siguientes características:

- El cliente Web, que es nombrado primera capa, es una interfaz de usuario estándar que permite al usuario recuperar información en la forma de páginas HTML o que permite activar aplicaciones.
- El Servidor Web, que es nombrado segunda capa, procesa y redirecciona la petición del usuario, recolecta los resultados y envía estos al cliente en la forma de páginas HTML.
- Puesto que la primera capa es la única responsable para interactuar con el usuario, la funcionalidad de la aplicación es implementada como un conjunto de servicios distribuidos en diferentes servidores. Debido a que una parte importante de los servicios debe ser oculta para el usuario, esto obliga a establecer políticas para la asignación de servicios.
- Las aplicaciones basadas en Web operan por lo regular en estaciones de trabajo. Los procesos del servidor son ejecutados en máquinas de servidores dedicados. En gran manera el desempeño de la aplicación es influenciada por la máquina servidor.
- La comunicación entre las capas relacionadas con el usuario está basada en el protocolo HTTP. Por esa razón la identificación de los recursos se lleva a cabo a través del uso del URL.
- Un sistema típico basado en WWW consiste de un sistema de base de datos, un lenguaje de consulta de base de datos y varios componentes tales como scripts y servidores WWW que trabajan conjuntamente. Estos sistemas pueden tener una gran cantidad de información confidencial que requiere complejas políticas de seguridad.

2.5 Esquemas de metadatos

Se han realizado diversos estudios concernientes al diseño y desarrollo de esquemas de metadatos. Entre estos se encuentran los esquemas realizados por la National Information Standards Organization (NISO) [34]:

- DUBLIN. El objetivo original del diseño Dublin fue definir un conjunto de elementos y algunas reglas que deben ser usadas por las organizaciones para describir sus propios recursos disponibles en la Web.
- TEI. La Iniciativa de Codificación de Texto (TEI, por sus siglas en Inglés), es un proyecto internacional que desarrolla pautas para el marcado de texto electrónico tales como novelas y poesías.
- METS. La Codificación de Metadatos y Estándares de Transmisión (METS, por sus siglas en Inglés) fue desarrollado para diseñar estándares de estructuras de datos para describir bibliotecas de objetos digitales complejos. METS es un esquema basado en el Lenguaje Extensible de Marcado para crear documentos XML.
- MODS. Los elementos del esquema para la Descripción de Objetos de Metadatos (MODS, por sus siglas en Inglés) son más ricos que el diseño Dublin, ya que estos elementos son más compatibles con los datos de librería en comparación con el estándar de diseño Dublin.
- LOM. El esquema de Aprendizaje de los Objetos de Metadatos (LOM: por sus siglas en Inglés) propone metadatos de aprendizaje para permitir el uso y rehusado de tecnologías que soporten el aprendizaje de recursos tal como el entrenamiento basado en computadoras y el aprendizaje a distancia.
- CDWA. Las categorías para la Descripción de Trabajos de Arte (CDWA, por sus siglas en Inglés) fue desarrollado para describir y acceder a la información de los objetos e imágenes de trabajos de arte.
- MPEG. El grupo de Expertos de Imágenes en Movimiento (MPEG, por sus siglas en inglés) ha desarrollado una serie de estándares para codificar la representación de audio y video digital.

- FGDC. El contenido Estándar de Metadatos Geospaciales Digitales (FGDC, por sus siglas en inglés) incluye datos topográficos y demográficos, sistema de información geográfico y ayuda computarizada de cartografía basada en archivos que son usados en una gran variedad de áreas, incluyendo estudios del uso del suelo, rastreos de cambios globales y climatológicos e imágenes de satélite.
- ISO 19115. Finalmente el estándar para metadatos de información geográfica ISO 19115, define un esquema internacional para describir servicios e información geográfica.

Aunque los elementos del esquema internacional del estándar ISO 19115 se apegan más a las necesidades requeridas para realizar este trabajo, se debe de tomar en cuenta que el esquema internacional del estándar ISO 19115 no es un esquema libre. De esta manera, el esquema del estándar FGDC será empleado en este trabajo para la creación de los metadatos de la base de datos que contendrá los recursos oceanográficos pertenecientes a la comunidad ROPVO y que serán puestos a disposición del público en general a través de Internet.

2.6 Modelo en Cascada

El modelo en Cascada es usado en la implementación de sistemas de información y abarca las siguientes actividades: ingeniería y análisis del sistema, análisis de los requisitos del software, diseño, codificación, prueba y mantenimiento, ver figura 2.3.

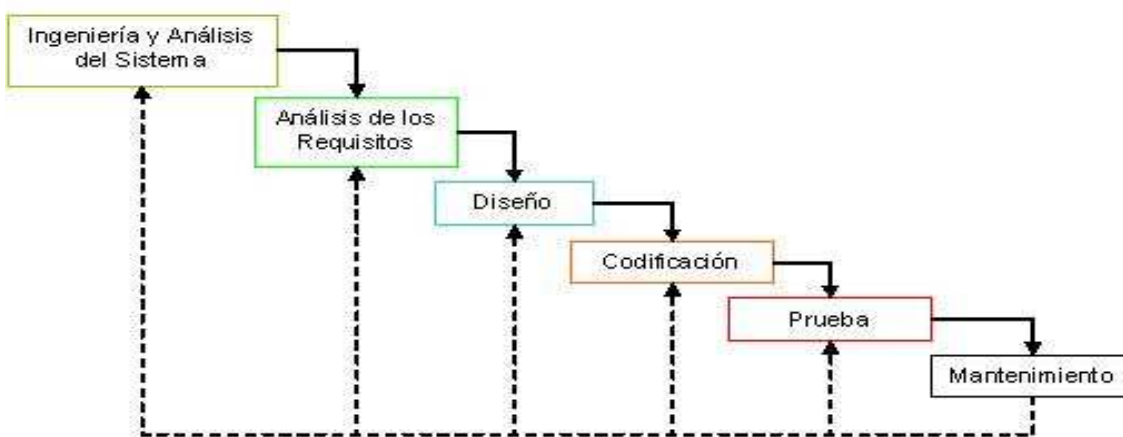


Figura 2.3: Ciclo de vida del modelo en Cascada.

- Ingeniería y Análisis del Sistema: Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.
- Análisis de los requisitos del software: el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.
- Diseño: el diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.
- Codificación: el diseño debe traducirse en una forma legible para la máquina.
- Prueba: La prueba se centra en la lógica interna del software. Se realizan pruebas que aseguren que la entrada de datos al sistema, producen los resultados que realmente se requieren.
- Mantenimiento: Los cambios ocurrirán debido a que hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

Desventajas:

- Los proyectos reales raramente siguen el flujo secuencial que propone el modelo, siempre hay iteraciones y se crean problemas en la aplicación del paradigma.
- Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requisitos. El ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos productos.

La ventaja de este mtodo radica en su sencillez ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.

Capítulo 3

Metodología

Este trabajo de tesis propone el diseño y desarrollo de una ASBUCORDI. La arquitectura permitirá a las comunidades virtuales compartir recursos. Su implementación requiere del análisis y diseño de una Metabase de Datos y de los procesos de Búsqueda y Consulta de Recursos Distribuidos.

3.1 Metabase de Datos

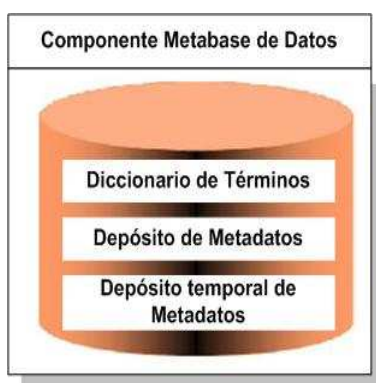


Figura 3.1: Componentes de la Metabase de Datos.

La figura 3.1 presenta el componente Metabase de Datos que tiene por objeto almacenar los metadatos de los recursos pertenecientes a un nodo de la comunidad. Entiendase por *metadato* como la información necesaria para describir un recurso, *metabase de datos* como la base de datos que almacena los recursos de un nodo de la comunidad y por un *nodo de la comunidad*, una computadora de la comunidad virtual. Los

elementos del componente Metabase de Datos son el Diccionario de Términos, el Depósito de Metadatos y el Depósito Temporal de Metadatos.

- **Diccionario de Términos.** Es el conjunto de nombres de recursos almacenados en el nodo de la comunidad. Esto es, los nombres de los recursos representan palabras clave. Estas palabras clave son comparadas con la solicitud de búsqueda para la identificación de estaciones y recursos que se desean consultar.
- **Depósito de Metadatos.** Es una estructura para el almacenamiento de los metadatos de los recursos del nodo de la comunidad.
- **Depósito Temporal de Metadatos.** Es una estructura para el almacenamiento temporal de los metadatos de los recursos del nodo de la comunidad. Esto es, los resultados producidos por el proceso de búsqueda son integrados en esta estructura temporal del nodo solicitante.

La tabla 3.1 muestra un ejemplo de la relación de los metadatos propuestos para la descripción de los recursos de los nodos de la comunidad virtual que se almacenan en el depósito de metadatos. La lista de metadatos almacenará una descripción de los recursos que desea compartir un nodo. Estos metadatos permitirán localizar, consultar y administrar los recursos de los nodos.

Tabla 3.1: Ejemplo de una relación de metadatos para la descripción de los recursos almacenados en un nodo de la comunidad.

Atributo	Descripción
Fecha de publicación	Fecha de publicación del recurso
Edición	Edición del recurso
Publicador	Quien propone el recurso a compartir
Sigue ...	

Continuación...	
Lugar de publicación	Donde se realizó la publicación
Clave de estación	Identificador de la estación
Clave de recurso	Identificador del recurso
Título	Título del recurso
Nombre	Nombre del recurso
Descripción	Breve descripción que detalle la importancia del recurso
Dirección en red	Dirección electrónica del recurso compartido
Nombre del recurso en red	El nombre del archivo o servicio que permitirá obtener el conjunto de datos
Relación	Nombre de la relación o tabla contenedora de datos
Usuario	Nombre de usuario
Password	Clave de acceso
Fin de la tabla 3.1	

3.2 Búsqueda Distribuida

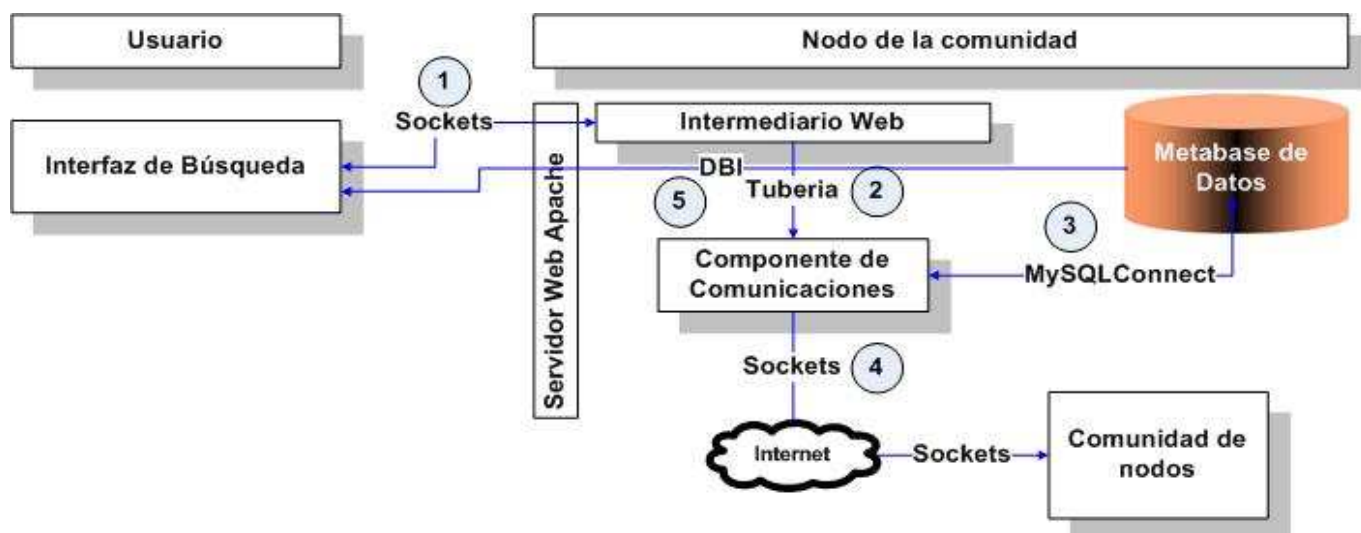


Figura 3.2: Componentes de la Búsqueda Distribuida.

La figura 3.2 presenta el Proceso de Búsqueda Distribuida que tiene por objeto encontrar recursos dispersos en la comunidad. Los componentes del Proceso de Búsqueda Distribuida son la Interfaz de Búsqueda, el Intermediario Web, el Componente Comunicaciones y la Metabase de Datos.

- Interfaz de Búsqueda. Es una aplicación Web que permite al usuario del sistema formular consultas que son procesadas por el mecanismo de Búsqueda.
- Intermediario Web. Es un programa que sirve de interfaz entre el Componente de Comunicaciones y la Interfaz de Búsqueda.
- Componente de Comunicaciones. Realiza la búsqueda de los metadatos en el **nodo local** y en los **nodos remotos** que forman parte de la comunidad.
- Metabase de Datos. Este componente contiene la base de datos que almacena el esquema de los metadatos de los recursos disponibles en el nodo local, así como el Diccionario de Términos, el Depósito de Metadatos y el Depósito Temporal de Metadatos, sección 3.1.

En primer lugar, el usuario accede a un nodo de la comunidad a través de su explorador Web (HTTP), a este nodo se le conoce como *nodo solicitante*. En seguida, el usuario accede a la Interfaz de Búsqueda

y formula una solicitud de búsqueda de recursos, ver figura 3.2. El componente Interfaz de Búsqueda envía la solicitud al componente Intermediario Web por medio de Sockets (1). A su vez, el componente Intermediario Web envía la solicitud de búsqueda al Componente Comunicaciones a través de Tuberías (2). En seguida, el Componente Comunicaciones extrae los recursos locales del componente Metabase de Datos a través de MySQLConnect (3). La búsqueda se propaga a otros nodos de la comunidad utilizando Sockets (4).

3.2.1 Interfaz de Búsqueda

La Interfaz de Búsqueda incluye las siguientes funciones: Diccionario de Estaciones, Diccionario de Recursos y Compartir Recursos.

- Diccionario de Estaciones. Este diccionario permite al administrador del sistema organizar los datos de los atributos de la relación *stations* del nodo de la comunidad, tales como el identificador de la estación, nombre y descripción de la estación.
- Diccionario de Recursos. Este diccionario permite al administrador del sistema organizar los datos de la relación *Resource* del nodo de la comunidad, tales como el identificador del recurso, nombre y descripción del recurso.
- Compartir Recursos. Esta función permite al usuario del sistema ingresar y compartir recursos oceanográficos del nodo de la comunidad.

3.2.2 Intermediario Web

El Intermediario Web es un programa que sirve de interfaz entre el Componente de Comunicaciones y la Interfaz de Búsqueda. Las comunicaciones entre la Interfaz Web y el Intermediario Web se llevan a cabo mediante sockets.

3.2.3 Componente de Comunicaciones

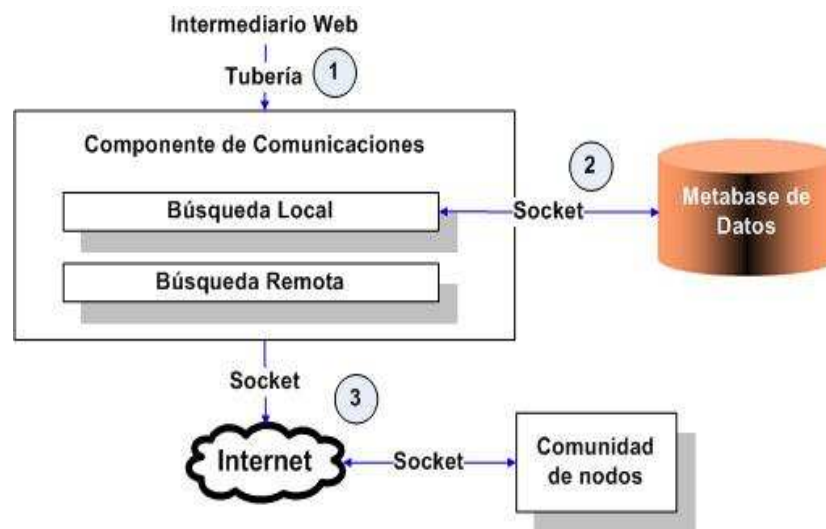


Figura 3.3: Componente de Comunicaciones.

En la figura 3.3 se presenta el Componente de Comunicaciones, el cual tiene por objeto buscar los recursos en los nodos de la comunidad. Los subcomponentes del Componente de Comunicaciones son: Búsqueda Local y Búsqueda Remota.

- Búsqueda Local. Es el conjunto de actividades que tienen por objeto extraer los metadatos de los recursos almacenados en el depósito de metadatos del nodo solicitante.
- Búsqueda Remota. La meta de estas actividades es propagar la solicitud de búsqueda de recursos hacia otros nodos de la comunidad. En cada uno de los nodos remotos de la comunidad, también se activa la Búsqueda Local.

Búsqueda Local

La Búsqueda Local, tal como se muestra en la figura 3.4, incluye las siguientes funciones: Analizador Léxico, Construcción de Instrucciones SQL y Ejecución de Consulta.

- Analizador Léxico. Separa las palabras clave de la solicitud de búsqueda en función del diccionario de términos que se encuentra almacenado en el componente de Metabase de Datos.

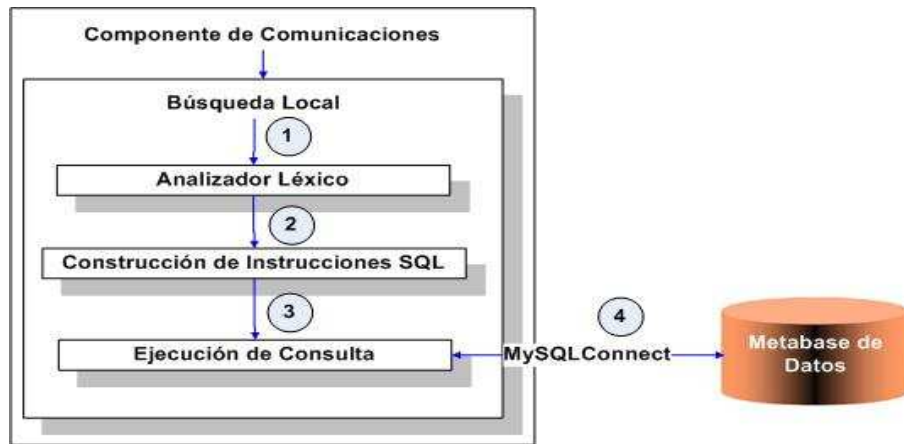


Figura 3.4: Subcomponente de Búsqueda Local.

- **Construcción de Instrucciones SQL.** Relaciona las palabras claves identificadas por el componente Analizador Léxico con los términos almacenados en el diccionario de términos y se construye una consulta en lenguaje SQL.
- **Ejecución de Consulta.** Se encarga de ejecutar la consulta SQL formulada en el punto anterior, extrayendo los metadatos de los recursos del componente de Metabase de Datos y almacenarlos en el *Depósito Temporal de Metadatos*. Enseguida los recursos localizados son mostrados al usuario a través del componente de Interfaz de Búsqueda.

La figura 3.4 muestra como el proceso Búsqueda Local envía la solicitud de búsqueda a la función Analizador Léxico (1). De igual forma, la función Analizador Léxico envía las palabras claves identificadas, a la función Construcción de Instrucciones SQL (2). La función Construcción de Instrucciones SQL envía la consulta formulada en lenguaje SQL a la función Ejecución de Consulta (3). Por último la función Ejecución de Consulta extrae los recursos almacenados en el Depósito de Metadatos y los almacena en el Depósito Temporal de Metadatos que forma parte del nodo solicitante.

La figura 3.5 muestra el diagrama de flujo de la función Analizador Léxico. Sea B , el conjunto de palabras de la solicitud de búsqueda, V el conjunto de nombres de las estaciones, W el conjunto de nombres de recursos oceanográficos y S el conjunto resultante de palabras de estaciones y recursos identificados. Ahora bien, si el conjunto de nombres de estaciones es diferente a vacío, entonces, se toma el elemento v del

conjunto de nombre de estaciones V , si el nombre de la estación v , forma parte del conjunto de palabras contenidas en la solicitud de búsqueda entonces, v se agrega al conjunto de estaciones y recursos identificados S . En seguida el elemento v se elimina del conjunto de nombre de estaciones V . Por otra parte, si el elemento v no forma parte del conjunto de nombres de estaciones entonces, el elemento v es eliminado del conjunto V y no se agrega al conjunto S . El objetivo de estos pasos es identificar los nombres de las estaciones que forman parte del conjunto de palabras de la solicitud de búsqueda del usuario. Ahora bien, si el conjunto de nombres de estaciones es igual a vacío entonces, se inicia el proceso de identificación de palabras asociadas a recursos, si el conjunto de nombres de recursos es diferente a vacío entonces, w es un elemento del conjunto de nombres de recursos W , si el nombre del recurso w , forma parte de las palabras contenidas en la solicitud de búsqueda de recursos entonces, el elemento w formará parte del conjunto S , En seguida, el elemento w se elimina del conjunto de nombres de recursos W . Por otra parte, si el elemento w no forma parte del conjunto de nombres de recursos entonces, el elemento w es eliminado del conjunto W . Finalmente, si el conjunto de nombres de recursos es igual a vacío entonces, termina la ejecución. El objetivo de estos pasos es identificar los nombres de estaciones y recursos que forman parte del conjunto de palabras de la solicitud de búsqueda del usuario. Como ejemplo, considere $B = \text{"primer", "nivel", "del", "mar", "en", "tampico", "y", "veracruz"}$, $V = \text{Mezquital, La Pesca, Tampico, Madero, Tuxpan, Veracruz}$ y $W = \text{Primer nivel del mar, Temperatura del aire, Temperatura del mar, Velocidad del viento, Dirección del viento, Presión barométrica, Marea alta, Marea baja, Salinidad}$. Ahora bien, de acuerdo al procedimiento Analizador Léxico, ver figura 3.5. El conjunto S resultante es "128,700,pw1" . Esto se debe a que las claves "128" y "700" corresponden a las estaciones de "Tampico" y "Veracruz" respectivamente, que contienen el recurso $\text{"primer nivel del mar"}$ y la clave "pw1" que corresponde al recurso $\text{"Primer nivel del mar"}$.

La figura 3.6 muestra el diagrama de flujo de la función Construcción de la Instrucción SQL. Sea S el conjunto de palabras de estaciones y recursos identificados y C el conjunto de nombres de estaciones y recursos para la formulación de la solicitud de búsqueda en lenguaje SQL. Si el conjunto de palabras de estaciones y recursos identificados es diferente a vacío entonces, s es un elemento que forma parte del conjunto de palabras de estaciones y recursos identificados, en seguida, el elemento s formará parte del

conjunto de caracteres para la formulación de la búsqueda C , finalmente, el elemento s se elimina del conjunto de palabras de estaciones y recursos identificados. Finalmente, si el conjunto de estaciones y recursos es vacío, entonces termina la función. El objetivo de estos pasos es formular la consulta de recursos en lenguaje SQL y almacenarla en el conjunto S . Como ejemplo. Para la entrada $S = "128,700,pw1"$. El resultado del algoritmo 3.6 es $C = "Select * from DepositoDeMetadatos Where ClaveEstación = 128 y ClaveEstación = 700 y ClaveRecurso = pw1"$, donde el campo ClaveEstación almacena el identificador de la estación y el campo ClaveRecurso almacena el identificador del recurso. Estos dos campos pertenecen al depósito de metadatos, ver tabla 3.1.

La función Construcción de la Instrucción SQL consiste de un conjunto de instrucciones que tiene la finalidad de establecer conexión con la Metabase de Datos del nodo de la comunidad y extraer los metadatos que cumplan con la solicitud de búsqueda de recursos del usuario, ver figura 3.7. Por ejemplo, se toma la cadena de formato de búsqueda $C = "Select * from DepositoDeMetadatos Where ClaveEstación = 128 y ClaveEstación = 700 y ClaveRecurso = pw1"$ y se procede a establecer conexión con la base de datos del nodo ROPVO. Enseguida se ejecuta la cadena de formato de búsqueda C y se extraen los datos del Depósito de Metadatos. Finalmente se almacenan los metadatos extraídos en el Depósito Temporal de Metadatos.

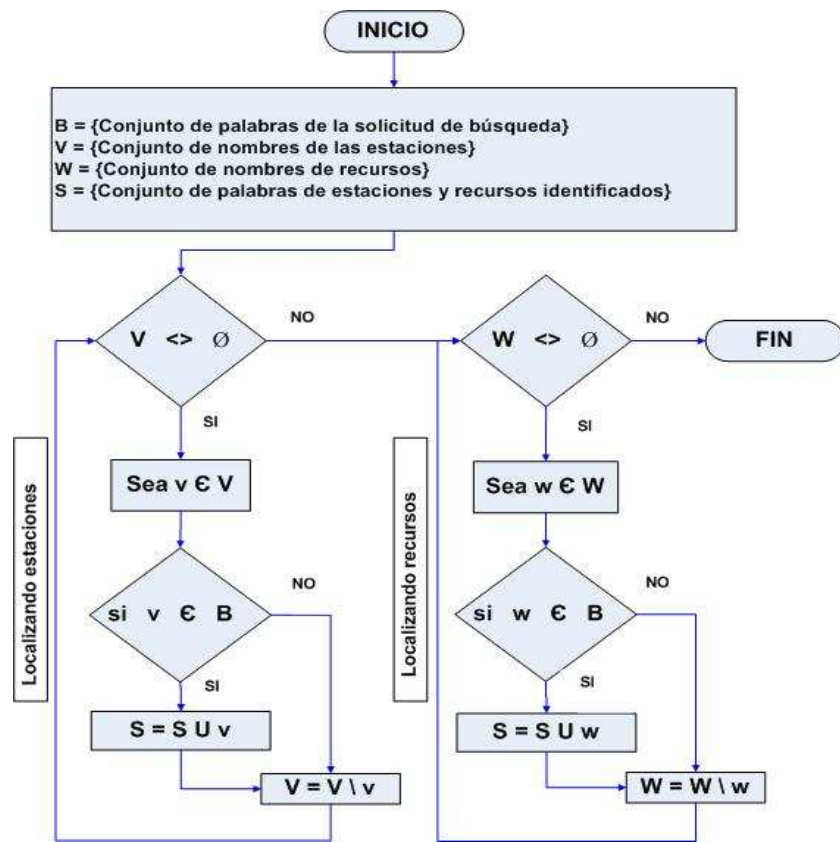


Figura 3.5: Diagrama de flujo de la función Analizador Léxico.

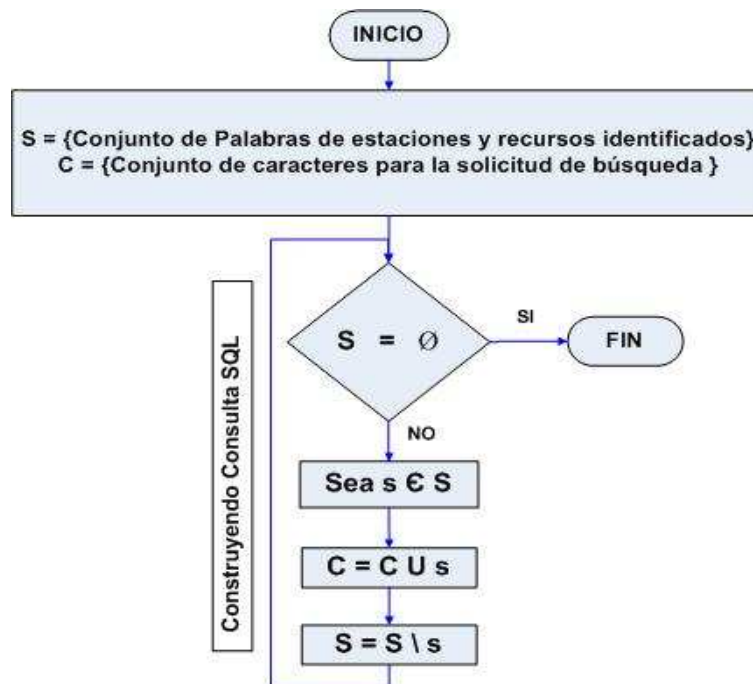


Figura 3.6: Diagrama de flujo de la función Construcción de la Instrucción SQL.

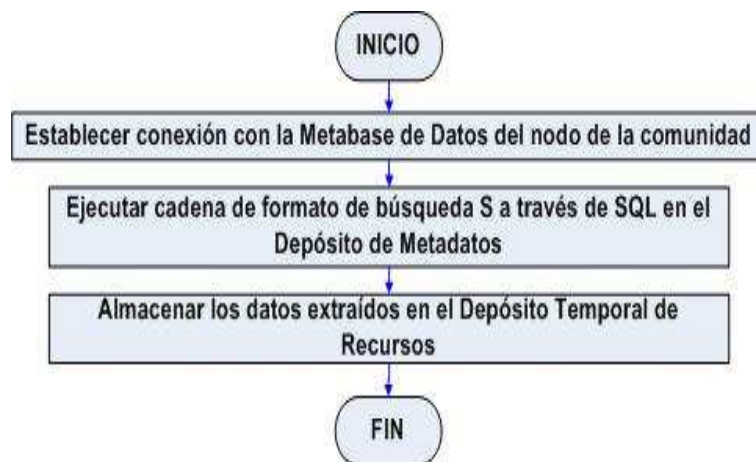


Figura 3.7: Diagrama de flujo que muestra el proceso para la extracción de los metadatos.

Búsqueda Remota

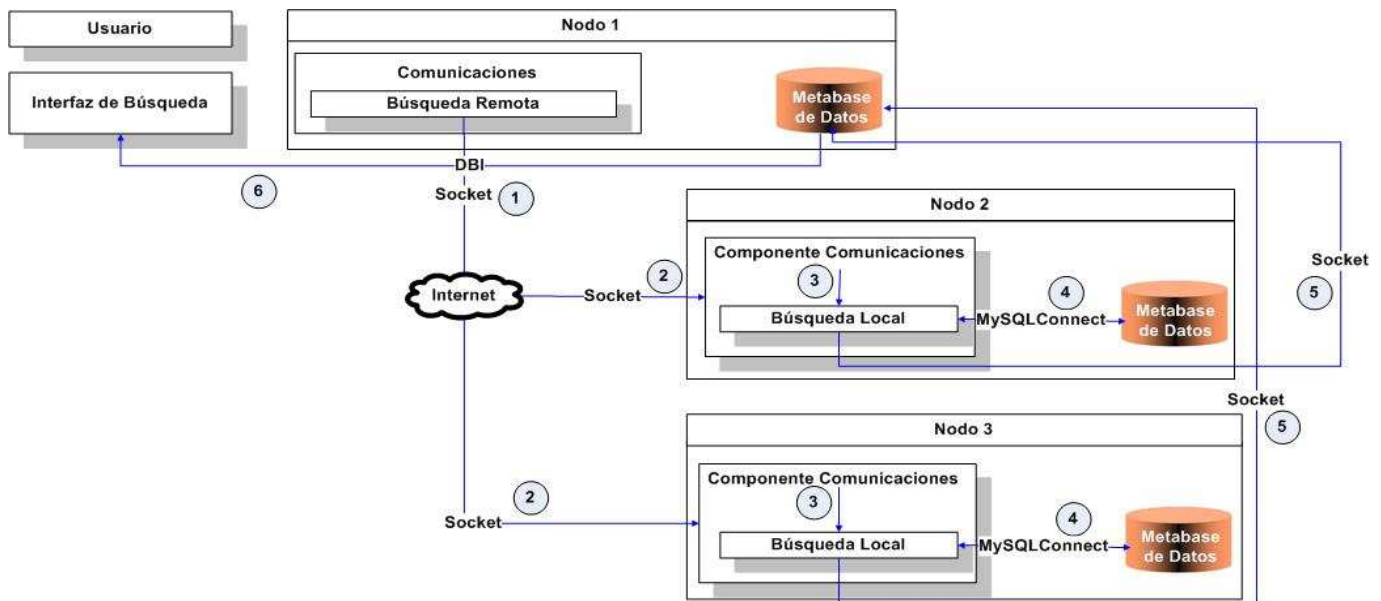


Figura 3.8: Subcomponente de Búsqueda Remota.

En la figura 3.8 se observa que el subcomponente Búsqueda Remota propaga la solicitud de búsqueda en los nodos de la comunidad a través de Internet usando Sockets (1); cada nodo que recibe la solicitud de búsqueda desde un nodo solicitante se conoce como *nodo receptor*. Enseguida, el Componente de Comunicaciones del nodo receptor, recibe la solicitud de búsqueda (2) y la envía hacia su subcomponente Búsqueda Local (3). El subcomponente Búsqueda Local extrae los metadatos de los recursos almacenados en el componente Metabase de Datos (4). En seguida, el subcomponente Búsqueda Local del nodo receptor envía los metadatos localizados hacia el componente Metabase de Datos del nodo solicitante (5). Finalmente el componente Interfaz de Búsqueda extrae los metadatos almacenados en el depósito de metadatos del nodo solicitante y se muestran al usuario a través de una aplicación Web (6).

3.2.4 Presentación de Resultados

Los metadatos obtenidos por el Proceso de Búsqueda, se muestran al usuario en forma de lista a través del componente Interfaz de Búsqueda. La lista de metadatos visible para el usuario está formada por

los metadatos de título, descripción del recurso y dirección del recurso en red. La lista de metadatos no visibles para el usuario está formada por los metadatos de fecha de publicación, edición, publicador, lugar de publicación, identificador del recurso, identificador de la estación, nombre del recurso, nombre de la relación, nombre usuario, password y abreviación del recurso. La razón del ocultamiento de estos recursos es por seguridad. Ahora bien, el proceso de Búsqueda permite acceder a recursos a través de enlaces Web y almacenados en bases de datos de los nodos de la comunidad.

Para los recursos accesibles a través del Web, el Proceso de Búsqueda muestra al usuario los metadatos de título, descripción del recurso y dirección del recursos en red. El texto del metadato de título aparece en forma de enlace y permite el acceso a la página Web del recurso que describe. Con respecto al almacenamiento de recursos en bases de datos, es importante aclarar que los nodos de la comunidad cuentan con dos bases de datos, separados entre sí, con objetivos completamente diferentes. La metabase de datos tiene la finalidad de almacenar una breve descripción que permita la compartición de recursos entre los nodos de la comunidad y la base de datos del sistema tiene como objetivo almacenar datos de diferentes recursos. La base de datos de los nodos estan en cada uno de los nodos de la comunidad. Por ejemplo, la base de datos de cada nodo contiene datos de recursos de: nivel del mar, salinidad, temperatura del mar y aire, nivel de marea baja y alta, velocidad y dirección del viento, presión barométrica, velocidad del mar, entre otros. Para los recursos almacenados en bases de datos el Proceso de Búsqueda muestra al usuario los metadatos de título, descripción del recurso y dirección del recurso en red. El texto del metadato de título aparece en forma de enlace y permite el acceso al Proceso de Consulta Distribuida del recurso que describe. Además, el metadato de título esta precedido de una caja de verificación que tiene la funcionalidad de un interruptor, apagado ó encendido. Cuando éste interruptor es encendido por el usuario, indica al Proceso de Búsqueda que éste recurso se ha seleccionado para su consulta, teniéndose entonces acceso a los datos de los recursos almacenados en las base de datos de los nodos de la comunidad. Finalmente, a través de un botón se lanza el Proceso de Consulta. En este punto de transición del Proceso de Búsqueda de Recursos Distribuidos hacia el Proceso de Consulta Distribuida, los recursos marcados por el usuario en la caja de verificación, pasarán a ser almacenados en una matriz de memoria M del nodo

solicitante. La figura 3.9 muestra un ejemplo de la matriz de metadatos M , que almacena los metadatos de los recursos que el usuario desea consultar, la matriz contiene r renglones por c columnas. Igualmente la tabla 3.2 ejemplifica una descripción del significado de las claves que aparecen en la columna Clave Recurso de la figura 3.9. Por último, los metadatos se envían al Proceso de Consulta Distribuida.

Tabla 3.2: Ejemplo de lista de claves y descripción de recursos.

Abreviación	Descripción
pwl	Nivel primario del mar
atp	Temperatura del aire
wtp	Temperatura del mar
wsd	Velocidad del viento
wdr	Dirección del viento
bpr	Preción barométrica
hw	Marea alta
lw	Marea baja
sal	Salinidad

Matriz M

Clave	ChBox	# Estación	Título	Descripción	Dirección red	Base de Datos	Relación	Usuario	Contraseña	Clave Recurso	Abreviación Estación
1	ChBox_1	700	Estación de Mezquitil	Datos de la estación mareográfica Mezquitil	148.204.119.114	Pharosdb	sz700	pharosdb	*****	pwl	MezqMX
2	ChBox_2	128	Estación de Tampico	Datos de la estación mareográfica Tampico	148.204.119.114	Pharosdb	sz128	pharosdb	*****	pwl	TampMX
3	ChBox_3	136	Estación de la Pesca	Datos de la estación mareográfica la Pesca	148.204.119.114	Pharosdb	sz136	pharosdb	*****	pwl	PescMX
4	ChBox_4	730	Estación de Veracruz	Datos de la estación mareográfica Veracruz	148.204.119.55	Pharosdb	sz730	pharosdb	*****	pwl	VeramX
5	ChBox_5	700	Estación de Mezquitil	Datos de la estación mareográfica Mezquitil	148.204.119.114	Pharosdb	sz700	pharosdb	*****	wtp	MezqMX
6	ChBox_6	128	Estación de Tampico	Datos de la estación mareográfica Tampico	148.204.119.114	Pharosdb	sz128	pharosdb	*****	wtp	TampMX
7	ChBox_7	720	Estación de Tuxpan	Datos de la estación mareográfica Tuxpan	148.204.119.56	Pharosdb	sz720	pharosdb	*****	pwl	TuxpMX
....

i

j

p

q

Figura 3.9: Ejemplo de la matriz de memoria que almacena los metadatos de los recursos que el usuario del sistema desea consultar.

3.3 Consulta Distribuida

La Consulta Distribuida es un mecanismo que permite formular una consulta y recuperar los datos oceanográficos almacenados en las bases de datos de diferentes nodos de la comunidad. En la figura 3.10 se presenta la *Consulta Distribuida*. Los componentes de la Consulta Distribuida son: la Interfaz de Consulta, el Intermediario de Consulta y el Componente de Extracción de Datos.

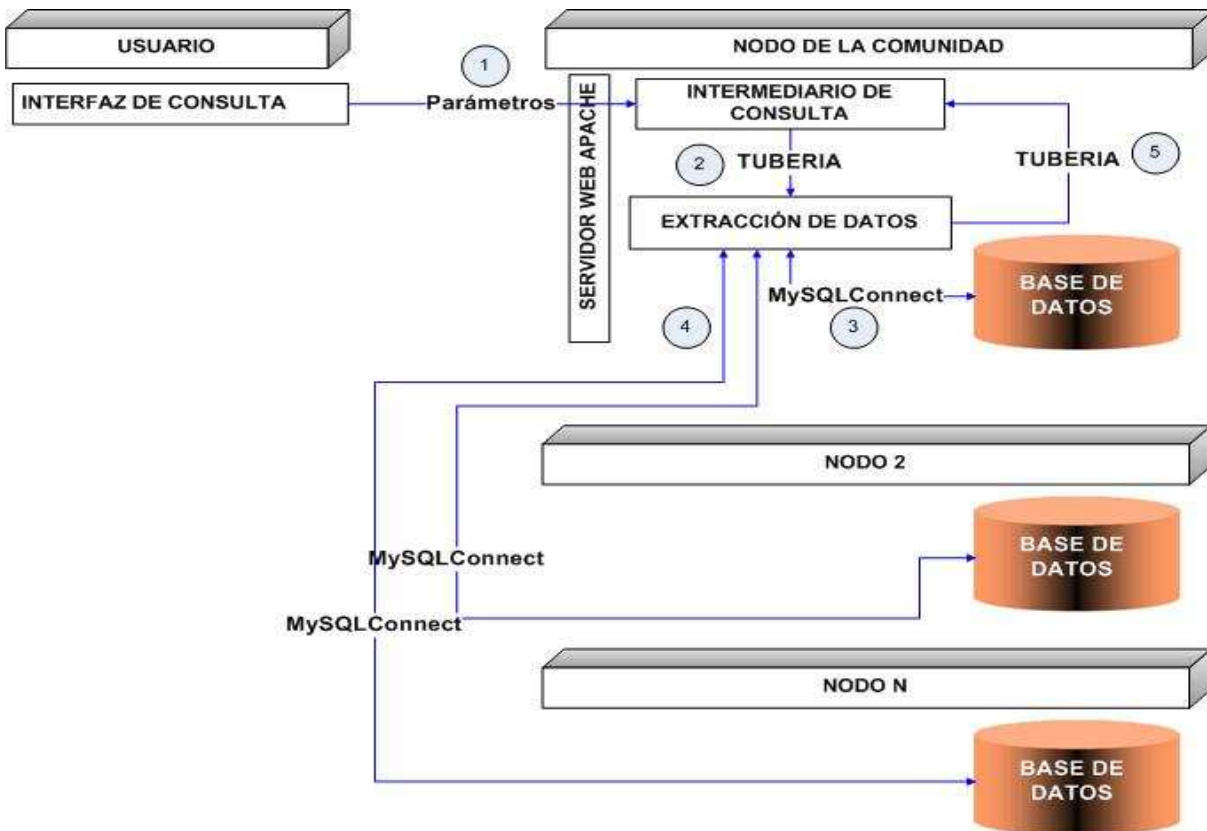


Figura 3.10: Componentes del Proceso de Consulta Distribuida.

- **Interfaz de Consulta.** Es una aplicación Web que permite al usuario del sistema formular consultas distribuidas sobre los recursos identificados por el Proceso de Búsqueda.
- **Intermediario de Consulta.** Se encarga de coordinar la extracción de recursos y de la integración de los mismos para ser presentados en la interfaz de consulta.
- **Componente de Extracción de Datos.** Se encarga de establecer la conexión con la Base de Datos y extraer los datos de los recursos de diferentes nodos de la comunidad.

Como se muestra en la figura 3.10, la Consulta Distribuida se realiza en cuatro etapas: Acceso a la Interfaz de Consulta, Formulación de la Consulta, Procesamiento de la Consulta y la Presentación de Resultados de la Consulta Distribuida, mismas que son descritas a continuación:

3.3.1 Interfaz de Consulta

Primeramente el usuario accede al Componente Interfaz de Consulta. Enseguida el Componente Interfaz de Consulta lee la matriz M de metadatos que fue generada por el proceso de Búsqueda de Recursos (ver figura 3.9). La figura 3.11 muestra el diagrama de flujo que usa la matriz M de metadatos para la agrupación de las estaciones, donde M es una matriz de orden (pq) , que contiene la relación descrita en la tabla 3.3.

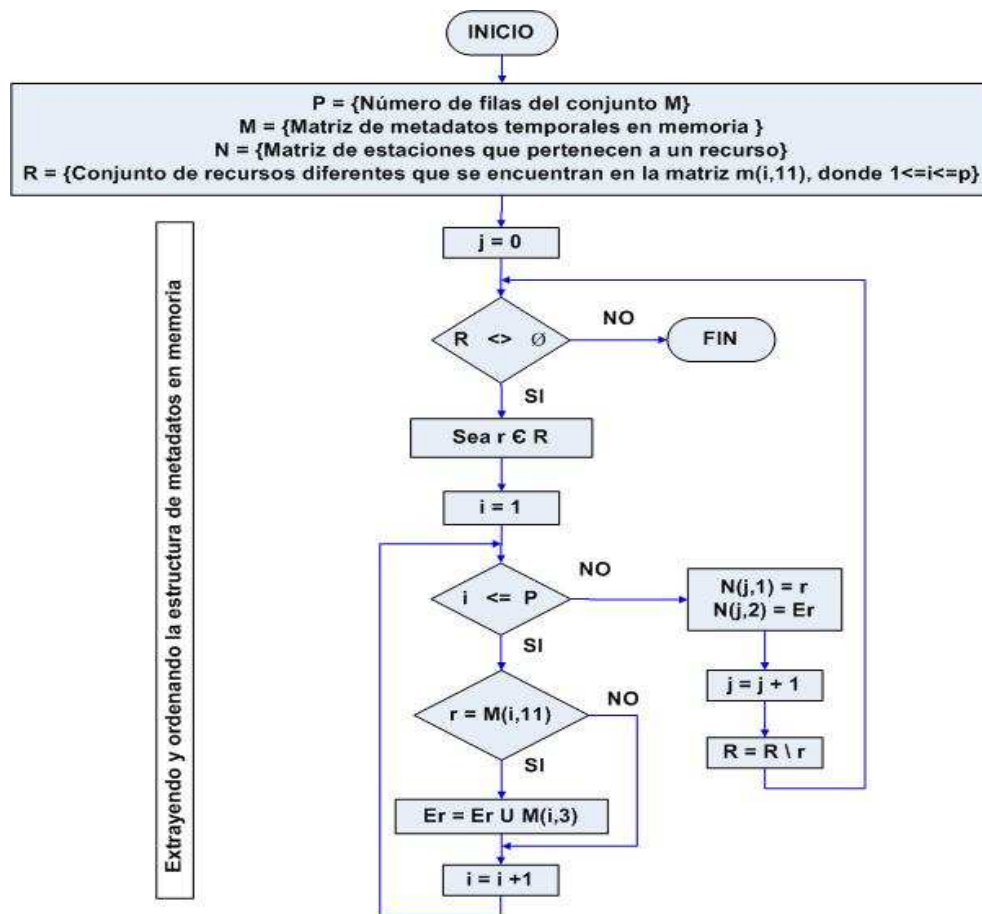


Figura 3.11: Diagrama de flujo que muestra el proceso para la creación de un conjunto de estaciones que tienen un recurso en común.

Sea E_r el conjunto de estaciones que contienen un recurso dado. Ahora bien, se verifica que existan estaciones en el conjunto R , si existen estaciones en R , entonces se toma un elemento de R y se almacena en r . Enseguida, se comparan todos los elementos de la matriz $M(i, 11)$ con el recurso almacenado en r . Para los casos en que la comparación sea verdadera, el número de estación $M(i, 3)$ se almacena en el conjunto E_r . Así, cuando ya no existan más elementos por analizar en la matriz $M(i, 11)$, se toma el recurso r y se almacena en la matriz $N(j, 1)$ y se toma el conjunto de estaciones E_r y se almacena en la matriz $N(j, 2)$. En caso de que ya no existan recursos en el conjunto R por analizar se termina el programa. El objetivo de estos pasos es obtener el conjunto de estaciones que tienen un recurso en común y almacenarlos en la matriz N , ver figura 3.12. La matriz N será usada más adelante por el componente Intermediario de Consulta.

Ahora bien, el usuario procede a formular la solicitud de consulta, para lograr esto, el componente Interfaz de Consulta contiene cajas de texto, listas de opciones múltiples y botones que permiten al usuario definir los parámetros para la búsqueda de recursos. La definición de los parámetros se realiza en tres secciones: Parámetros de Búsqueda Básica, Opciones de Graficación y Opciones ASCII, ver figura 3.13. Finalmente a través de un botón se lanza el Procesamiento de la Consulta.

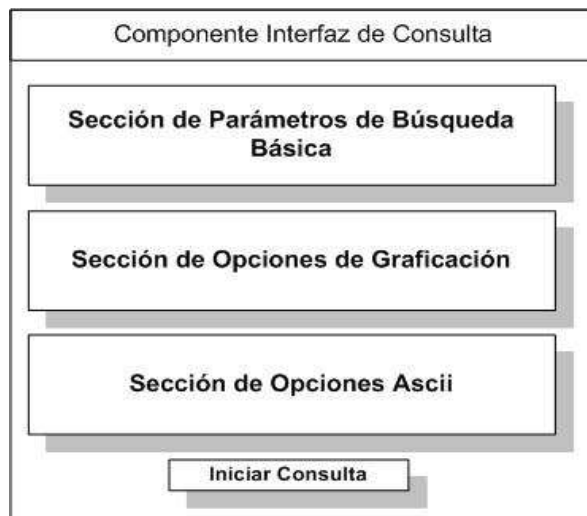


Figura 3.13: Secciones de la Interfaz de Consulta.

- **Parámetros de Búsqueda Básica.** En esta sección se define el intervalo de fechas de extracción de datos. También se define el formato para la presentación de resultados al usuario, siendo estos,

archivos de texto ó archivo gráfico.

- **Opciones de Graficación.** Permite definir el título, tamaño, borde, y las coordenadas en x,y de la gráfica. También permite definir varias etiquetas para la gráfica.
- **Opciones ASCII.** Esta sección permite activar o desactivar los comentarios. También permite definir un símbolo para sustituir los datos inexistentes.

3.3.2 Intermediario de Consulta

La formulación de la consulta del usuario, es enviada al componente Intermediario de Consulta(1), ver figura 3.10. El componente Intermediario de Consulta recibe la solicitud, enseguida toma la matriz N , (ver figura 3.12) y genera una lista de estaciones a consultar. La lista de estaciones debe tener el siguiente formato: número de la estación, seguido de un guión y posteriormente la clave del recurso. En caso de que existan más estaciones, el formato de la lista de las estaciones es separada por comas. Por ejemplo "128-pwl,700-pwl,128-wtp". Esta lista de estaciones es utilizada para la extracción de datos (recursos) de las bases de datos de los nodos ROPVO.

La figura 3.14 muestra el diagrama de flujo para la creación de la lista de estaciones L_i . Considere $E_i = N(i, 2)$ como un conjunto, donde i es el número de fila de la matriz N . Esto es, E_i contiene el conjunto de estaciones para el recurso almacenado en $N(i, 1)$. Sea L_i la cadena que contiene el formato de lista mostrado en el párrafo anterior. Primeramente, verificamos que existan estaciones en $N(i, 2)$, si existen estaciones en $N(i, 2)$, entonces se almacena el recurso contenido en $N(i, 1)$ en r_i y se almacena la lista de estaciones contenida en $N(i, 2)$ en E_i . Para cada estación e almacenada en E_i , se agrega el recurso r_i y la estación e a la lista L_i . En caso de que ya no existan estaciones almacenadas en E_i , se procede a tomar la siguiente lista de estaciones $N(i, 2)$ y se almacena en E_i y el programa termina. Todos estos pasos tienen como objetivo crear la cadena que contendrá el formato de lista de estaciones con su recurso, respectivamente. Por ejemplo "128-pwl,700-pwl,128-wtp".

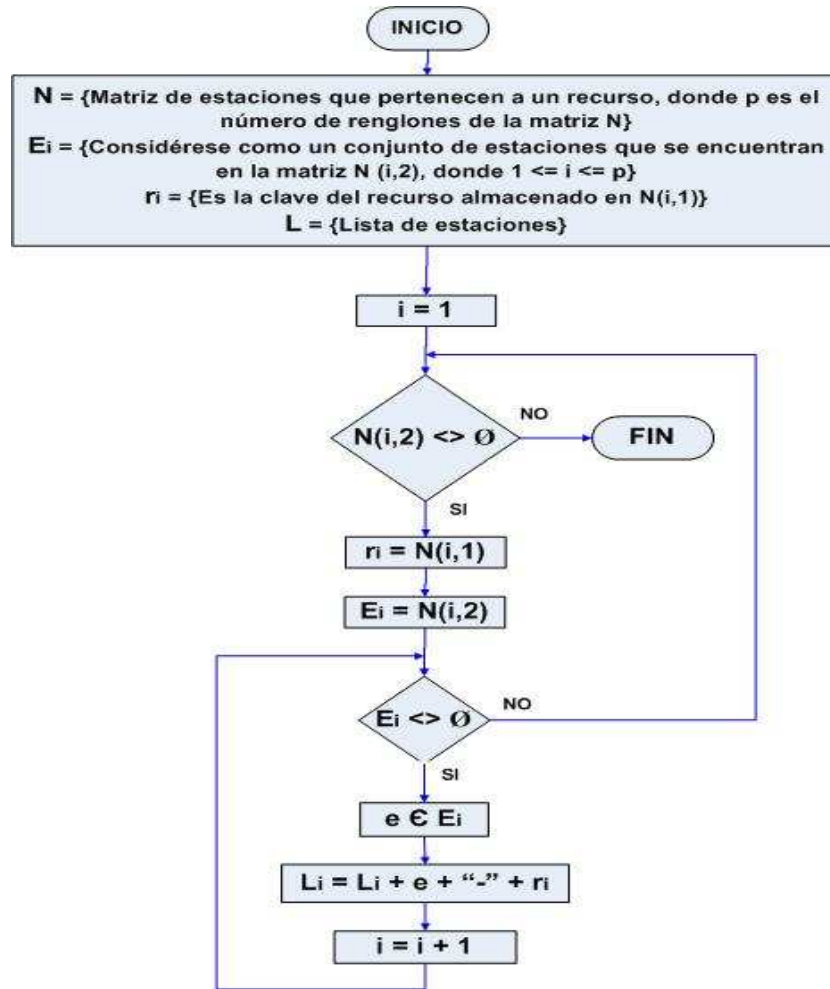


Figura 3.14: Diagrama de flujo que muestra el proceso de creación de la lista de estaciones.

La cadena que contiene el formato de lista de estaciones L_i y los parámetros de la solicitud de búsqueda (fecha de inicio de la consulta, fecha final de la consulta, tipo de archivo de salida, título de la gráfica, tamaño de la gráfica, borde, y las coordenadas en x,y de la gráfica) son enviados al Componente Extracción de Datos a través de Tuberías(2), ver figura 3.10. Enseguida el Componente Extracción de Datos extrae los datos de las Base de Datos de los nodos ROPVO. La extracción de los datos se realiza a través de MySQLConnect (3)(4).

3.3.3 Extracción de Datos

La extracción de datos consiste en un conjunto de instrucciones que tienen la finalidad de establecer conexión con las bases de datos de los nodos de la comunidad y extraer los recursos que cumplan con la solicitud de búsqueda, ver figura 3.15.

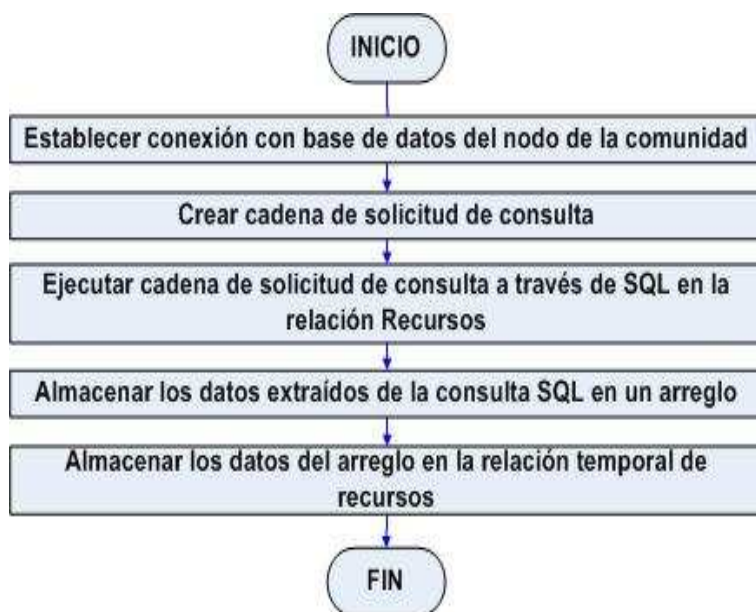


Figura 3.15: Diagrama de flujo que muestra el proceso para la extracción de datos.

Para hacer esto, primeramente se verifica que existan elementos en la cadena de formato de lista de estaciones L_i . Si existen elementos en la cadena L_i , se procede a tomar un elemento de la cadena de formato de lista de estaciones, por ejemplo: "128-pwl". Enseguida se procede a extraer la estación y el recurso del elemento cadena de formato de lista de estaciones. El siguiente paso es extraer la fecha de inicio y la fecha final de la solicitud de consulta del usuario. Después, se procede a establecer conexión con la base de datos del nodo de la comunidad, enseguida se crea la cadena de solicitud de consulta en lenguaje SQL, usando como parámetros de la consulta, el número de estación, la clave del recurso, la fecha de inicio y la fecha final de la consulta. Se ejecuta la cadena de consulta y se extraen los datos de la relación recursos. Finalmente se almacenan los datos del arreglo en la relación temporal de recursos. Los datos almacenados en la relación temporal de recursos se envían al componente Intermediario de Consulta a través de una Tubería (5), ver figura 3.10. Finalmente, la Interfaz de Consulta presenta los resultados del Proceso de Consulta Distribuida al usuario.

3.3.4 Presentación de Resultados

La presentación de los resultados al usuario se lleva a cabo de dos formas. La primera es presentar los resultados de la consulta al usuario en forma de archivo de texto. La segunda es graficar los resultados de la consulta y presentarla al usuario a través de la Interfaz Web.

La figura 3.16 ejemplifica el diseño del archivo de texto para la presentación de los resultados del proceso de Consulta Distribuida al usuario. El campo "estación y clave del recursos" contiene las estaciones que fueron consultadas. El campo "recurso" contiene los nombres de los recursos consultados. El campo "fecha inicial y fecha final" almacena los rangos de fechas de la consulta. El campo "número de registros" contiene el total de registros consultados. El campo "base de datos" contiene el nombre de la base de datos consultada por el Proceso de Consulta Distribuida. El campo "clave estación" almacena el número de estación consultada. El campo "fecha" almacena la fecha en que fue registrado este recurso. El campo "clave recurso" almacena la clave del recurso consultado. Ahora bien, desde la columna *M1* hasta la columna *M10* se almacenan mediciones (considerando mediciones cada 6 minutos), del recurso que contiene el campo "clave recurso". Esto es, desde *M1* a *M10* corresponde a 1 hora de mediciones por intervalos de cada 6 minutos.

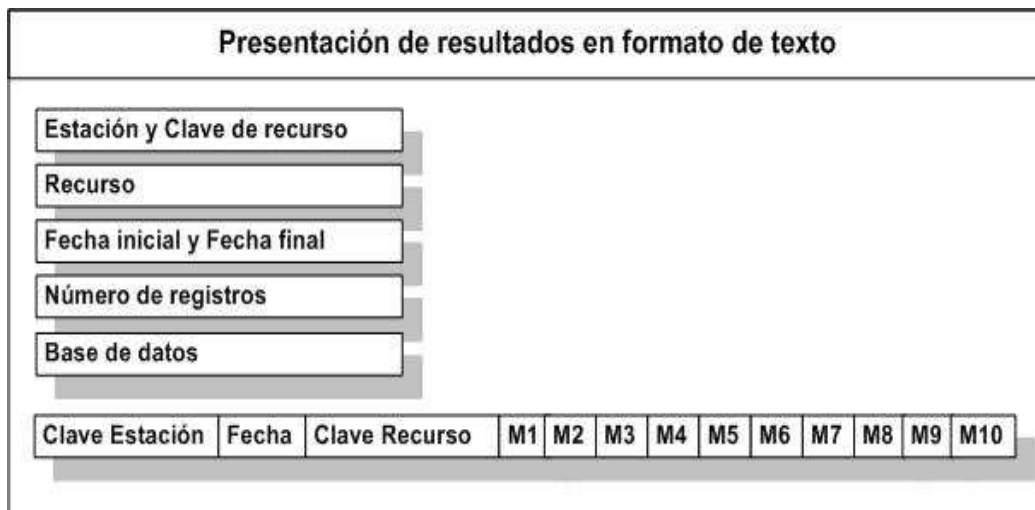


Figura 3.16: Ejemplo del diseño del archivo de texto para la presentación de resultados.

La figura 3.17 ejemplifica el diseño de la gráfica para la presentación de los resultados del proceso de Consulta Distribuida. El área "Valor de la medición del recurso" contiene los rangos posibles, desde el valor mínimo de cero hasta el valor máximo de los datos de los recursos consultados. El área "Intervalo de fechas de la consulta" contiene el intervalo de fecha inicial y final para la presentación de los resultados de la consulta. El "Area de la gráfica", contiene los datos de las estaciones consultadas en forma de gráfica y el área de "Estaciones y recursos" contiene una lista de las estaciones consultadas. Ahora bien, es importante mencionar que la gráfica de resultados cuenta con el mecanismo para agrupar los datos de las estaciones que pertenecen a un recurso. En caso de que el usuario seleccione consultar más de un recurso, entonces la Interfaz de Resultados mostrará, por cada recurso, una gráfica de resultados.

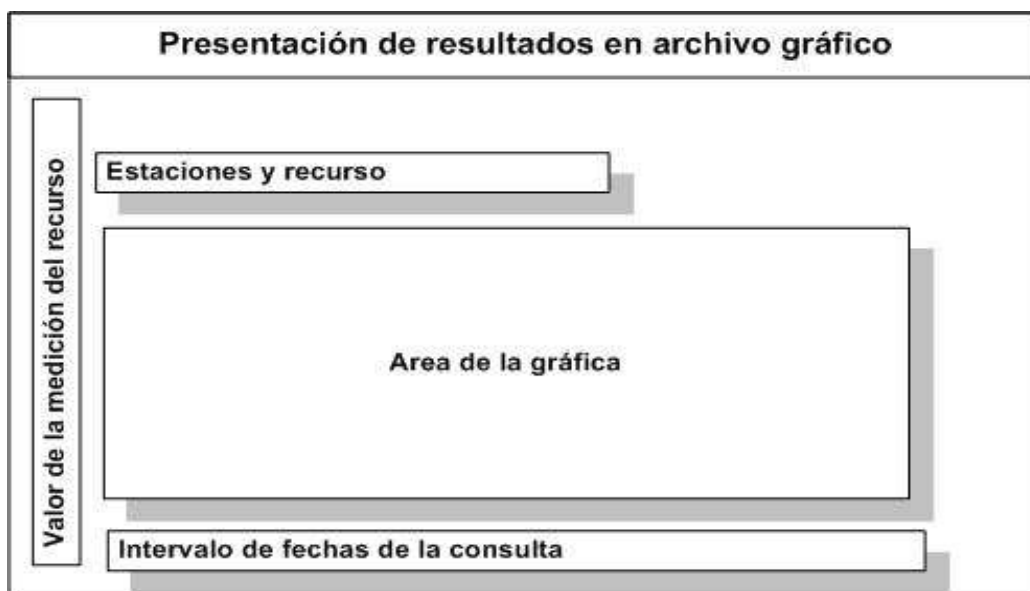


Figura 3.17: Ejemplo del diseño del archivo gráfico para la presentación de resultados.

Capítulo 4

Resultados

Este trabajo de tesis propone el diseño y desarrollo de una arquitectura de software para la búsqueda y consulta de recursos distribuidos sobre Internet. La arquitectura permitió a la comunidad virtual ROPVO compartir recursos oceanográficos. Para su implementación se llevo acabo la creación y codificación de una *Metabase de Datos* y de los procesos de *Búsqueda y Consulta de Recursos Distribuidos*.

4.1 Metabase de Datos

El diseño de la metabase de datos incluye cuatro factores que son: la definición del esquema de metadatos, la creación de la metabase de datos, la creación del depósito de metadatos y la creación del depósito temporal de metadatos, mismos que son descritos a continuación.

4.1.1 Definición del Esquema de Metadatos

El esquema de metadatos se utiliza para describir los recursos que permiten la localización y administración de los recursos oceanográficos de la red ROPVO. Este esquema está basado en las especificaciones del Comité de Datos Geográfico Federal (FGDC, por sus siglas en inglés) y es el resultado del análisis para identificar los metadatos necesarios para la localización de los recursos oceanográficos en los nodos ROPVO. Con base en la tabla 3.1 se definen los metadatos para la creación del esquema de metadatos, utilizado en este trabajo. (ver tabla 4.1).

En la siguiente sección se describe la creación de la Metabase de Datos.

Tabla 4.1: Esquema de metadatos tomados del estandar FGDC, los cuales fueron seleccionados para la relación de metadatos.

Metadato	Descripción del metadato
publication_Date	Fecha de publicación
edition	Edición
publisher	Quien propone la metabase de datos
publication_place	donde se realizó la publicación
idstation	Identificador de la estación
idresource	Identificador del recurso
title	Título del esquema de metadatos
station_name	Nombre de la estación
resource_description	Descripción del recurso
Network_Address	Dirección electrónica del recurso compartido
Network_Resource_Name	El nombre del archivo o servicio que permitirá obtener el conjunto de datos
Relation	Nombre de la relación
User	Nombre de usuario
Password	Clave de acceso

4.1.2 Creación de la Metabase de Datos

La metabase de datos se creó en el SMBD MySQL y su nombre es DBmetadatos. DBmetadatos tiene la finalidad de almacenar los metadatos de los recursos compartidos por la organización de una manera clara, sencilla y ordenada. La figura 4.1 muestra la línea de comando para la creación de la base de datos DBmetadatos.

```
mysql> CREATE DATABASE DBmetadatos;
```

Figura 4.1: Línea de comando utilizada para la creación de la base de datos DBmetadatos.

4.1.3 Creación del Diccionario de Términos

El diccionario de términos se creó en la base de datos DBmetadatos y está formado por dos relaciones: la relación *stations* y la relación *resource*. La tabla 4.2 muestra la estructura de la relación *stations*, mientras que la figura 4.2 muestra la línea de comando utilizada para la creación de la relación *stations*.

Tabla 4.2: Estructura de la relación *stations*.

Campo	Tipo de dato
id	int(11)
idStation	int(11)
Station_Name	varchar(200)
Resource_Description	text
latitude	varchar(20)
longitude	varchar(20)
stnabbr	varchar(15)

```
mysql> Create table stations (id int(11), idStation int(11), station_name
varchar(200), resource_description text, latitude varchar(20), longitude
varchar(20), stnabbr (15));
```

Figura 4.2: Línea de comando utilizada en la creación de la relación stations.

La tabla 4.3 muestra la estructura de la relación resource, y en la figura 4.3 se muestra la línea de código para la creación de la relación resource.

Tabla 4.3: Estructura de la relación resource.

Campo	Tipo de dato
id	int(11)
idResource	varchar(11)
resource_name	varchar(200)
resource_description	text

```
mysql> create table resource (id int(11), idResource varchar(11),
resource_name varchar(200), resource_description text);
```

Figura 4.3: Línea de comando utilizada para la creación de la relación resource.

4.1.4 Creación del Depósito de Metadatos

El Depósito de Metadatos se creó en la base de datos DBmetadatos y su nombre es DataResource. En la tabla 4.4 se muestra la estructura de la relación DataResource mientras que en la figura 4.4 se muestra la línea de comando para la creación la relación DataResource.

Tabla 4.4: Estructura de la relación DataResource.

Campo	Tipo de dato
id	int(11)
publication_date	date
edition	varchar(100)
publisher	varchar(100)
publication_place	varchar(100)
idDataResource	varchar(11)
idStation	int(11)
idResource	int(11)
title	int(11)
resource_description	text
network_address	varchar(240)
network_resource_name	varchar(150)
relation	varchar(150)
User	varchar(150)
password	varchar(150)
stnabbr	varchar(15)

```
mysql> create table DataResource (id int(11), publication_date date, edition varchar(100),
publisher varchar(100), publication_place varchar(100), idDataResource varchar(100),
idStation int(11), idResource varchar(11), title varchar(150), resource_description text,
network_address varchar(240), network_resource_name varchar(150), relation varchar(150),
user varchar(150), password varchar(150), stnabbr varchar(15));
```

Figura 4.4: Línea de comando utilizada en la creación de la relación DataResource.

4.1.5 Creación del Depósito Temporal de Metadatos

El Depósito Temporal de Metadatos se creó en la base de datos DBmetadatos y su nombre es TempDataResource. En la tabla 4.5 se muestra la estructura de la relación TempDataResource mientras que en la figura 4.5 se muestra la línea de comando para la creación de la relación TempDataResource.

Tabla 4.5: Estructura de la relación TempDataResource.

Campo	Tipo de dato
id	int(11)
iduser	varchar(20)
IPServer	varchar(20)
publication_date	date
edition	varchar(100)
publisher	varchar(100)
publication_place	varchar(100)
idDataResource	varchar(11)
idStation	int(11)
idResource	int(11)
title	int(11)
resource_description	text
network_address	varchar(240)
network_resource_name	varchar(150)
relation	varchar(150)
user	varchar(150)
password	varchar(150)
Sigue ...	

Continuación...	
stnabbr	varchar(15)

```
mysql> create table TempDataResource (id int(11), iduser varchar(20), IPServer varchar(20),
publication_date date, edition varchar(100), publisher varchar(100), publication_place
varchar(100), idDataResource varchar(100), idStation int(11), idResource varchar(11), title
varchar(150), resource_description text, network_address varchar(240),
network_resource_name varchar(150), relation varchar(150), user varchar(150), password
varchar(150), stnabbr varchar(15));
```

Figura 4.5: Línea de comando utilizada en la creación de la relación TempDataResource.

4.2 Proceso de Búsqueda Distribuida

A continuación se describe el diseño y desarrollo de los componentes del proceso de búsqueda de recursos distribuidos: la Interfaz de Búsqueda, el Intermediario Web y el Componente Comunicaciones.

4.2.1 Acceso a la Interfaz de Búsqueda

Figura 4.6: Interfaz de Búsqueda de Recursos Distribuidos sobre Internet.

La figura 4.6 muestra el diseño de la Interfaz de Búsqueda de Recursos Distribuidos. La Interfaz de

Búsqueda es un programa creado con el lenguaje de programación Perl y su nombre es MBBusquedaDistribuida.cgi. La Interfaz de Búsqueda incluye las funciones: Diccionario de Estaciones, Diccionario de Recursos y Compartir Recursos Oceanográficos. Si el administrador del sistema elige dar un click en el vínculo Diccionario de Estaciones, entonces se inicia la ejecución del programa MBRegEstacion.cgi, que permite dar de alta estaciones mareográficas de las cuales el nodo ROPVO tiene datos. Esta página solicita al administrador que introduzca el identificador de la estación, nombre y la descripción de la estación, ver figura 4.7.

CICATA-IPN
Altamira

Red de Observaciones ROPVO-GM > ROPVO > Diccionario de Estaciones

Email CICATA
Español

Diccionario de Estaciones

28 May 2008 00:49 UTC

ROPVO-GM
TCOON
ROPVO
Datos
Predicción
Olas
Recursos

idEstacion

Nombre

Descripcion

Latitud

Longitud

Participantes

Figura 4.7: Interfaz para el registro de Estaciones Mareográficas.

Si el administrador del sistema elige dar un click en el vínculo Diccionario de Recursos entonces, se inicia la ejecución del programa MBRegTipoRecurso.cgi, que permite dar de alta recursos de los cuales el nodo ROPVO tiene datos. Esta página solicita al usuario que introduzca el identificador del recurso, nombre y la descripción del recurso, ver figura 4.8.

Si el administrador del sistema elige dar un click en el vínculo Compartición de Recursos Oceanográficos, entonces se inicia la ejecución del programa MBRegRecurso.cgi, el cual permite registrar los metadatos de los recursos de los cuales el nodo ROPVO tiene datos. Esta página solicita al usuario que introduzca el identificador de la estación, el identificador del recurso, la fecha de publicación, la edición, el publicador,

Figura 4.8: Interfaz para el registro de Diccionario de Recursos.

el lugar de publicación, el título, la descripción del recurso, la dirección en la red, el nombre del recurso en la red, el nombre de la relación, el nombre del usuario y la contraseña, (figura 4.9).

Figura 4.9: Interfaz para el registro de Compartición de Recursos Oceanográficos.

Ahora bien, para iniciar el proceso de Búsqueda Distribuida, primeramente el usuario escribe en la caja de texto, Descripción del recurso, la solicitud de Búsqueda, Figura 4.6. Enseguida procede a dar un click en el botón *Buscar*; esta acción enviará la solicitud de búsqueda al componente *Intermediario Web*.

4.2.2 Intermediario Web

El Intermediario Web es un programa creado con el lenguaje de programación Perl y su nombre es MBIntermediarioW.cgi. La figura 4.10 muestra las funciones que integran al Intermediario Web.

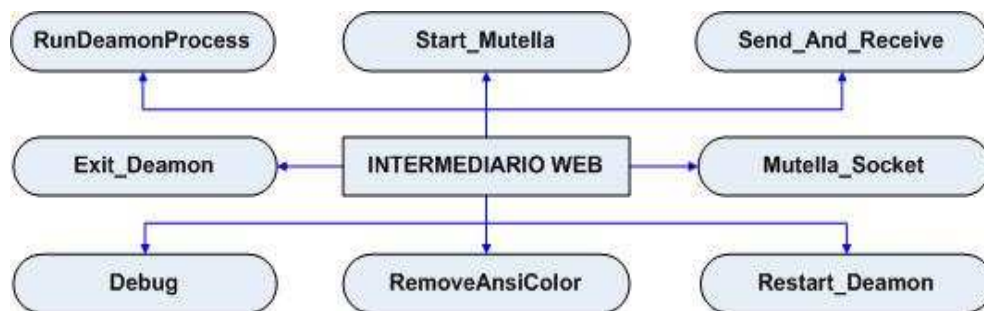


Figura 4.10: Funciones del Intermediario Web.

A continuación se describen cada uno de los elementos que integran al Intermediario Web.

- **Mutella_Socket.** Este procedimiento crea un socket para la comunicación entre la Interfaz Web de Búsqueda y el Intermediario Web.
- **RunDeamonProcess.** Este procedimiento contiene el mecanismo para recibir las peticiones de la Interfaz Web de Búsqueda y hacerlas llegar al Mecanismo de Búsqueda Distribuida.
- **Start_Mutella.** Inicializar el proceso de comunicación con el Intermediario Web por medio de una tubería.
- **Send_And_Receive.** Este procedimiento contiene las instrucciones necesarias para hacer llegar las peticiones de la Interfaz Web de Búsqueda al Intermediario Web.

- **Restart_Deamon.** Se encarga de reiniciar el proceso del Intermediario Web en caso de falla.
- **Exit_Deamon.** Este procedimiento se encarga de finalizar la ejecución de la aplicación Intermediario Web.
- **RemoveAnsiColor.** Este procedimiento se encarga de eliminar los caracteres de color producto de las respuestas del Mecanismo de Búsqueda de Recursos Distribuidos.
- **Debug.** Este procedimiento se encarga de mostrar los resultados del proceso de ejecución del Intermediario Web.

El Componente Comunicaciones es un programa P2P creado con el lenguaje de programación C++ y su nombre es *mutella.c*. El programa *mutella* contiene el proceso de búsqueda distribuida que a su vez contiene dos procesos que son la Búsqueda Local y la Búsqueda Remota.

4.2.3 Proceso de Búsqueda Local

El Proceso de Búsqueda Local (PBL) pertenece al Componente de Comunicaciones, (ver figura 3.3). El PBL es una función que se realizó con el lenguaje de programación C++ y su nombre es GnuSearch.cpp. La Implementación del PBL se realizó mediante la función void MgnuSearch::SendQuery() que pertenece a la clase gnuSearch, por lo tanto, esta función contiene el código que realiza el PBL.

```
Void MgnuSearch::SendQuery()
{
.
.
.
1  char* Dato = "/root/mutella-0.4.5/mutella/./AnalizadorLexico '%s'";
2  char DatoR[1000];
3  sprintf(DatoR,Dato, Squery.c_str());
4  system (DatoR);
.
.
.
}
```

Figura 4.11: Fragmento de código que ejemplifica el paso de parámetros de solicitud de búsqueda local al componente Analizador Léxico.

La figura 4.11 ejemplifica el fragmento de código que se encargó de realizar el PBL de recursos. Note que sQuery.c_str() es la variable que contiene la solicitud de búsqueda del usuario. La línea 4 muestra como la instrucción System() invoca al programa AnalizadorLexico y le envía el parámetro de solicitud de búsqueda almacenado en sQuery.c_str(). De esta manera se puede resumir que el PBL incluye las funciones Analizador Léxico, Construcción de la Instrucción SQL, Ejecución de la Consulta y Almacenamiento Temporal de Metadatos.

Función Analizador Léxico

El Analizador Léxico es un programa diseñado con el lenguaje de programación C++, ver sección 3.2.3. Su nombre es AnalizadorLexico.cpp que incluye dos programas que son *FBibliotecaEstaciones* y *FBibliotecaTipoRecurso*. *FBibliotecaEstaciones* es un programa que identifica los nombres de estaciones que forman

parte de la solicitud de búsqueda del usuario. La figura 4.12 ejemplifica las líneas de código del programa *FBibliotecaEstaciones*.

```
int FBibliotecaEstaciones()
{
.
.
.
1 char sqlcmdE[1000] = "Select * from Stations";
2 smhandle = mysql_connect(smhandle, "localhost", user, pass);
3 mysql_select_db(smhandle,database);

4 if (mysql_query(smhandle,sqlcmdE) == 0) {
5 crresult = mysql_store_result(smhandle);
6 corcount = mysql_num_rows(crresult);
}

7 if (corcount < 1) return 0;
8 for (i=0;i<corcount;i++) {
9     crfields = mysql_fetch_row(crresult);
10    r = RegExpReg (0, crfields[1], nombre);
11    if (r){
12        sprintf(numCad,"%d",i);
13        strcpy (MatEstaciones[i][0], numCad);
14        strcpy (MatEstaciones[i][1], "idStation");
15        strcpy (MatEstaciones[i][2], crfields[1]);
16        strcpy (MatEstaciones[i][3], crfields[0]);
17        i++;
    }
}
.
.
}
```

Figura 4.12: Fragmento de código que ejemplifica la detección de estaciones.

De la línea 1 a la línea 6 se obtienen los nombres de las estaciones de la relación *stations* que pertenecen a la metabase de datos *DBmetadatos*. De la línea 7 a la línea 17 se comparan los nombres de las estaciones con la solicitud de búsqueda del usuario. Al terminar la función, la matriz *MatEstaciones* almacenó el identificador de la matriz, la cadena "idStation", el nombre de la estación y la clave de la estación que pertenecen a la estación buscada.

FBibliotecaTipoRecurso es un programa que identifica los nombres de los recursos que forman parte de la solicitud de búsqueda del usuario. La figura 4.13 ejemplifica las líneas de código del programa FBibliotecaTipoRecurso.

```

int FBibliotecaTipoRecurso()
{
.
.
.
1 char sqlcmdT[1000] = "Select * from Resource";
2 smhandle = mysql_connect(smhandle, "localhost", user, pass);
3 mysql_select_db(smhandle,database);

4 if (mysql_query(smhandle,sqlcmdT) == 0) {
5 crresultT = mysql_store_result(smhandle);
6 corcount = mysql_num_rows(crresultT);
7 }

7 if (corcount < 1) return 0;

8 for (i=0;i<corcount;i++) {
9   crfieldsT = mysql_fetch_row(crresultT);
10   r = RegExpReg (0, crfieldsT[1], nombre);
11   if (r){
12     sprintf(numCad,"%d",i);
13     strcpy (MatEstaciones[i][0], numCad);
14     strcpy (MatEstaciones[i][1], "idResource");
15     strcpy (MatEstaciones[i][2], crfieldsT[1]);
16     sprintf(VidResource,"%s", crfieldsT[0]);
17     strcpy (MatEstaciones[i][3], VidResource);
18     i++;
19   }
20 }
.
.
.
}

```

Figura 4.13: Fragmento de código que ejemplifica la detección de recursos.

De la línea 1 a la línea 6 se obtienen los nombres de los recursos de la relación *Resource* que pertenecen a la metabase de datos *DBmetadatos*. De la línea 7 a la línea 18 se comparan los nombres de los recursos con la solicitud de búsqueda del usuario. Al terminar la función, la matriz *MatEstaciones* almacenó el identificador de la matriz, la cadena "idResource", el nombre del recurso y la clave del recurso que pertenecen al recurso buscado.

Construcción de la Instrucción SQL

La Construcción de la Instrucción SQL es un programa que se diseñó con el lenguaje de programación C++ el cual es llamado CadenaSQL (ver sección 3.2.3). La figura 4.14 muestra las líneas de código de la función *CadenaSQL*. De la línea 2 a la línea 15 se recorre la matriz *MatEstaciones* que contiene los nombres de las estaciones y recursos a consultar. Al terminar la función, la cadena *condición* contendrá la instrucción SQL para el acceso a los metadatos de los recursos.

```
int CadenaSQL()
{
.
.
.
1  char sqlcmdCR[1000] = "Select * from DataResource where ";
2  for (x=0; x<l; x++)
3  {
4  sprintf(Cond, "%s = %s", MatEstaciones[x][1], MatEstaciones[x][3]);
5  strcat(Condicion, Cond);
6  if ((x+1) < l)
7  {
8      if (strcmp(MatEstaciones[x][1], MatEstaciones[x+1][1] == 0)
9      {
10         strcat(Condicion, "or");
11     }else{
12         strcat(Condicion, ") and (");
13     }
14 }
15 }
16 cout << Condicion << endl;
.
.
.
}
```

Figura 4.14: Fragmento de código que ejemplifica la construcción de la cadena SQL.

Ejecución de la Consulta

La Ejecución de la Consulta es un programa que se diseñó con el lenguaje de programación C++, el cual es llamado EjecucionSQL (ver sección 3.2.3). La figura 4.15 muestra las líneas de código de la función.

De la línea 1 a la línea 8 se ejecuta la consulta SQL almacenada en la variable *Condición* y se extrajeron los metadatos que cumplieron con la solicitud de consulta de la relación *DataResource* que pertenece a la metabase de datos *DBmetadatos*. De la línea 9 a la línea 13, se encuentran los comandos que envía los metadatos obtenidos por la consulta a la función *Almacenamiento Temporal de Metadatos*.

```
int EjecucionSQL()
{
.
.
.
1  char sqlcmdCR[1000] = "Select * from DataResource where ";
2  smhandle = mysql_connect(smhandle, "localhost", user, pass);
3  mysql_select_db(smhandle, database);
4  strcat(sqlcmdCR, Condicion);

5  if (mysql_query(smhandle, sqlcmdCR) == 0) {
6      crresultCR = mysql_store_result(smhandle);
7      corcount = mysql_num_rows(crresultCR);
8  }

9  if (corcount < 1) return 0;
10 for (i=0; i<corcount; i++) {
11     crfieldsCR = mysql_fetch_row(crresultCR);
12     r = FInsertarFilasEnTempRecursos(crfieldsCR);
13 }

.
.
.
}
```

Figura 4.15: Fragmento de código que ejemplifica la ejecución de la cadena SQL.

Almacenamiento Temporal de Metadatos

El Almacenamiento Temporal de Metadatos es un programa que se diseñó con el lenguaje de programación C++, el cual es llamado *FInsertarFilasEnTempRecursos* (ver sección 3.2.3). La figura 4.16 ejemplifica las

líneas de código de la función. De la línea 1 a la línea 7 se establece conexión con la base de datos *DB-metadatos*. La línea 8 almacena temporalmente los metadatos recibidos por la función *EjecucionSQL* en la relación *TempDataResource* perteneciente a la metabase de datos *DBmetadatos*.

```
int FInsertarFilasEnTempRecursos(MYSQL_ROW crfieldsTR)
{
.
.
.
1  char sqlcmdTR[1000] = "insert into TempDataResource
(idUser,IPServer,publiation_date,edition,publisher,publiation_place,idDataResource,idStatio
n,idResource,title,resource_description,network_address,network_resource_name,relation,
User,password,latitude,longitude,snabbr) values
('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s','%s')";
.
.
2  smhandle = mysql_connect(smhandle, host, user, pass);
3  mysql_select_db(smhandle,database);
.
.
4  sprintf(sqlcmd1, sqlcmdTR, idUser, IPEndPoint, crfieldsTR[0], crfieldsTR[1], crfieldsTR[2],
crfieldsTR[3], crfieldsTR[4], crfieldsTR[5], crfieldsTR[6], crfieldsTR[7], crfieldsTR[8],
crfieldsTR[9], crfieldsTR[10], crfieldsTR[11], crfieldsTR[12], crfieldsTR[13], crfieldsTR[14],
crfieldsTR[15], crfieldsTR[16]);
.
.
5  if (mysql_query(smhandle,sqlcmd1) == 0) {
6      cout << "Datos Registrados...." << endl;
7  }
.
.
}
```

Figura 4.16: Fragmento de código que ejemplifica el almacenamiento temporal de metadatos.

4.2.4 Proceso de Búsqueda Remota

El proceso de Búsqueda Remota también pertenece al Componente de Comunicaciones, ver figura 3.3. El Proceso de Búsqueda Remota (PBR) es una función que se realizó con el lenguaje de programación C++ el cual es llamado *GnuSearch.cpp* y se encargó de la propagación la solicitud de búsqueda del usuario en los nodos ROPVO. Ahora bien, la función encargada de recibir la solicitud de búsqueda que se propagó por un nodo ROPVO es: `void MgnuNode::Receive_Query(packet_Query* Query, int nLength)`, que pertenece

a la clase `gnuNode`, por lo tanto, esta función contiene el código que realiza el PBL para la extracción de los metadatos remotos al nodo solicitante.

```
void MGNUNode::Receive_Query(packet_Query* Query, int nLength)
{
.
.
.
1   char* Dato = "/root/mutella-0.4.5/mutella/AnalizadorLexico '%s'";
2   char DatoR[1000];
3   sprintf(DatoR,Dato,pSR->m_query.Text);
4   system (DatoR);
.
.
}
```

Figura 4.17: Fragmento de código que ejemplifica el paso de parámetros de solicitud de búsqueda remota al componente `AnalizadorLexico`.

La figura 4.17 ejemplifica el fragmento de código que se encarga de recibir la solicitud de búsqueda que se propagó desde un nodo remoto ROPVO. Note que `pSR->m_query.Text` es la variable que contiene la solicitud de búsqueda del usuario. Ahora bien, en la línea 3 se observa que la variable `DatoR` almacena la ruta y nombre del programa `Analizador Lexico` más el parámetro de solicitud de búsqueda. La línea 4 muestra como la instrucción `System()` invoca al programa `AnalizadorLexico.cpp` y le envía el parámetro de solicitud de búsqueda almacenado en `pSR->m_query.Text`.

4.2.5 Resultados de la Búsqueda Distribuida

Los resultados de la Búsqueda Distribuida no es más que la representación de los resultados del PBL y PBR en una Interfaz Web. La función `MBResultados` forma parte del programa `MBBusquedaGeneral.cgi` y es la encargada de mostrar los resultados de la solicitud de búsqueda en la Interfaz Web. La figura 4.18 muestra la Interfaz Web de Resultados de la Búsqueda de Recursos Distribuidos. La Interfaz Web de Resultados presenta al usuario los metadatos de título del recurso, de descripción y la dirección física en red. Si el usuario desea consultar recursos almacenados en páginas Web entonces, procede a dar un click

en el título del recurso que desea consultar, ver figura 4.19. Si el usuario desea consultar los datos de los recursos almacenados en base de datos entonces, el usuario procede a dar un click en una de las cajas de verificación que pertenecen al recurso que se desea consultar. En seguida se procede a dar un click en el botón Consultar, ésta acción inicia el *proceso de Consulta de Recursos Distribuidos*.



Figura 4.18: Interfaz Web que muestra los metadatos que se obtuvieron del proceso de Búsqueda de Recursos Distribuidos.

4.3 Proceso de Consulta Distribuida

A continuación se describen los elementos que incluyeron al proceso de Consulta Distribuida: el Acceso a la Interfaz de Consulta, Formulación de la Consulta, el Proceso de Consulta, el proceso de Extracción de Datos y los Resultados de la Consulta.

4.3.1 Acceso a la Interfaz de Consulta

La figura 4.20 ejemplifica las líneas de código para la creación de un conjunto de estaciones que tienen un recurso en común.

```

1  foreach $fila (@Filas){
  .
  .
  .
2  $elementos = @ClasRecursos;
3  if ($elementos == 0) {
4    $ClasRecursos[0][0] = $MEstaSelec[$J][10]; #idrecurso
5    $ClasRecursos[0][1] = $MEstaSelec[$J][2]; #Num Station
6  }else{
7    $Localizado = 0;
8    for ($ski = 0; $ski < $elementos; ++$ski){
9      if ($ClasRecursos[$ski][0] eq $MEstaSelec[$J][10]) {
10         $ClasRecursos[$ski][1] = $ClasRecursos[$ski][1] . "," . $MEstaSelec[$J][2];
11         $Localizado = 1;
12      }
13    }
14    if ($Localizado == 0) {
15      $ClasRecursos[$ski][0] = $MEstaSelec[$J][10];
16      $ClasRecursos[$ski][1] = $MEstaSelec[$J][2];
17    }
18  }
19  $J++;
20 }
  .
  .
  .

```

Figura 4.20: Fragmento de código que ejemplifica la creación de un conjunto de estaciones que tienen un recurso en común.

Primeramente se hace notar que la matriz *@MEstaSelec* representa la matriz *M* y la matriz *@ClasRe-*

cursons representa la matriz N , ver sección 3.3.1. Ahora bien, de la línea 1 a la línea 20 se recorre la matriz $@MEstaSelec$ para obtener las estaciones que tienen un recurso en común. De la línea 3 a la línea 6 se registra la primera estación que tiene un recurso en común en la matriz $@ClasRecursos$. Enseguida, de la línea 7 a la línea 17 se verifica si el recurso almacenado en $@MEstaSelec[j][10]$ pertenece a uno de los recursos almacenados en la matriz $@ClasRecursos$. Para los casos en que la comparación sea verdadera, la estación $MEstaSelec[j][2]$ formó parte del conjunto de estaciones del recurso $ClasRecursos[ki][1]$. En caso de que el recurso $MEstaSelec[j][10]$ no formara parte del conjunto de recursos $ClasRecursos[ki][0]$ entonces, de la línea 14 a la línea 17 se creó un nuevo conjunto de estaciones con un recurso en común llamado $MEstaSelec[j][10]$ que almacenó la estación $MEstaSelec[j][2]$.

4.3.2 Formulación de la Consulta

La formulación de la consulta se llevó a cabo a través de la Interfaz de Consulta que es un programa que se creó con el lenguaje de programación Perl, cuyo nombre es *MBpquery.cgi*. La figura 4.21 muestra el diseño de la Interfaz de Consulta que incluye las secciones Parámetros de Búsqueda Básica, Opciones de Graficación y Opciones Ascii.

The screenshot shows a web browser window displaying the 'Basic Query Parameters' form. The form is organized into several colored sections with various input fields and radio buttons. On the left side of the browser window, there is a vertical column of logos, including the University of the Pacific (UP) logo and logos for NOAA, SNN, and other organizations.

Basic Query Parameters

- Dates:** Input field containing '01/23/2004-01/30/2008'. A text box on the right explains the date format: 'Enter the range of dates (UTC) for which you want data. In general you can enter dates in the form mm/dd/yyyy-mm/dd/yyyy, but other specifications such as yesterday, now, 7d, and mm/yyyy also work.'
- Format:** Radio buttons for 'Graph' (selected) and 'ASCII'. Under 'Graph', there are radio buttons for 'GIF' (selected) and 'PostScript'. Under 'ASCII', there are radio buttons for 'Text Rows' and 'Columns'. A text box on the right says: 'Select your desired output format.'
- Units:** Radio buttons for 'Metric' (selected) and 'English'. Under 'Metric', there are radio buttons for 'DNR'. A text box on the right says: 'Select your desired output units. CBI units indicates the default unit of measurement as stored in our database (always an integer value).'
- Elevation:** A dropdown menu set to 'Station Datum'. A text box on the right explains: 'What do you want vertical elevations (e.g., water level) referenced to? The default is station datum, which is an arbitrary zero established at each station. Other elevations may not be available for the station(s) you've requested.'
- Date Format:** A dropdown menu set to 'Entered below (default if blank)'. A text box on the right explains: 'How do you want dates to be displayed? The default gives reasonable output for most requests, but if you need more control you can either select a predefined format from the top selection box or enter your own strftime specification in the bottom box.'
- Time Zone:** A dropdown menu set to 'Coordinated Universal Time (UTC)'. A text box on the right says: 'Select the time zone for output dates/times.'
- SSLList:** An empty input field. A text box on the right explains: 'This is an "advanced" method to select specific combinations of stations and series. You can safely leave this field alone.'

At the bottom of the form, there is a button labeled 'Click here to retrieve data'.

Graph Options

- Labels:** An input field labeled 'Title' containing the text 'Recursos'.

Figura 4.21: Interfaz Web para la Consulta de Recursos Distribuidos.

4.3.3 Proceso de Consulta

El Procesamiento de la Consulta lo realizó el Intermediario de Consulta que es un programa que se creó con el lenguaje de programación Perl, cuyo nombre es *MBpmdata.cgi* y se encarga de procesar la solicitud de consulta del usuario. La figura 4.22 ejemplifica las líneas de código para la creación del formato para la lista de estaciones. Primeramente se hace notar que la matriz *@VarClasRecursosFila* representa la matriz *N*, ver sección 3.3.1, y la variable *Varsslist* representan la cadena de formato para la lista de estaciones. De la línea 1 a la línea 19 se recorre la matriz *VarClasRecursosFila* que contiene los conjuntos de estaciones que contiene un recurso en común. De la línea 7 a la línea 10 se obtiene el conjunto de estaciones que pertenecen a un recurso y se almacena en la matriz *VarEstaciones*. Enseguida, de la línea 11 a la línea 19 se creó la cadena de formato de lista de estaciones y se almacenó en la variable *Varsslist*. Finalmente el formato para la lista de estaciones es enviado al Proceso de Extracción de Datos.

```

.
.
.
1  $elementos = @VarClasRecursos;
2  for ($k = 0; $k < $elementos; ++$k) {
4      if ($k > 0){
5          $Varsslist = $Varsslist . ",";
6      }

7  @VarClasRecursosFila = &FcadenaAmatriz2(@VarClasRecursos[$k]);
8  #$VarClasRecursosFila[0]; # recurso
9  #$VarClasRecursosFila[1]; # estaciones
10 @VarEstaciones = split /\,/ , $VarClasRecursosFila[1];

11 $elementosEst = @VarEstaciones;
12 for ($k5 = 0; $k5 < $elementosEst; ++$k5)
13 {
14     if ($k5 > 0){
15         $Varsslist = $Varsslist . ",";
16     }

17     $Varsslist = $Varsslist . $VarEstaciones[$k5] . "-" . $VarClasRecursosFila[0];
18 }
19 }
.
.
.

```

Figura 4.22: Fragmento de código que ejemplifica el proceso para la creación del formato para la lista de estaciones.

4.3.4 Proceso de Extracción de Datos

El proceso de Extracción de Datos es un programa que se creó con el lenguaje de programación C++ cuyo nombre es *sxext.c*. La figura 4.23 ejemplifica las líneas de código para la extracción de los recursos de la base de datos del nodo ROPVO.

```

1  const char* smselect =
   "SELECT jul,src,smv0,smv1,smv2,smv3,smv4,smv5,smv6,smv7,smv8,smv9 FROM sx%s "
   "WHERE ser = '%s' and jul >= %s and jul <= %s ORDER BY jul";

2  smhandle = mysql_connect(smhandle, host, user, pass);
3  if (!smhandle) {
4      printf("#%s: ?unable to connect to MySQL server (%s)...abort!\n",
5          arg0,host);
6      return -1;
7  }
8  if (mysql_select_db(smhandle,database) != 0) {
9      printf("#%s: ?unable to connect to database (%s)...abort!\n",arg0,database);
10     return -1;
11 }

12 sprintf(sqlcmd,smselect,id,ser,sjul0,sjul1);
13 if ((mysql_query(smhandle,sqlcmd)==0) && (smresult = mysql_store_result(smhandle))) {
14     smrow = mysql_fetch_row(smresult);
15 }

16 if (!fillflag || showempty) {
17     OutputRec(id,gmt,ser,outv);
18     if (showsrc) {
19         printf("#src:");
20         for(i=0;i<10;i++) { printf(" %s",srctag[outp[i]]); }
21         printf("\n");
22     }
23     filled+=fillflag;
24     data++;
25 }
26 mysql_free_result(smresult);

```

Figura 4.23: Fragmento de código que ejemplifica el Proceso de Extracción de Datos.

De la línea 1 a la línea 11 se estableció conexión con la base de datos *DBmetadatos*. La línea 12 almacena en la variable *sqlcmd* la solicitud de consulta y el formato de cadena de lista de estaciones. De la línea 13 a la línea 15 se ejecuta la solicitud de consulta almacenada en la variable *sqlcmd* y se extrajeron

los recursos de la base de datos del nodo ROPVO. De la línea 16 a la línea 27 se envió los resultados obtenidos de la ejecución de la solicitud de consulta hacia la Interfaz Web del usuario.

4.3.5 Resultados de la Consulta Distribuida

La figura 4.24 muestra en forma de gráfica los datos oceanográficos recolectados de la estación mareográfica de Tampico Tamaulipas. El intervalo de fechas de los datos mostrados por la gráfica van de 01/01/2006 a 01/09/2006 con una altura máxima del nivel del mar de aproximadamente 4m.

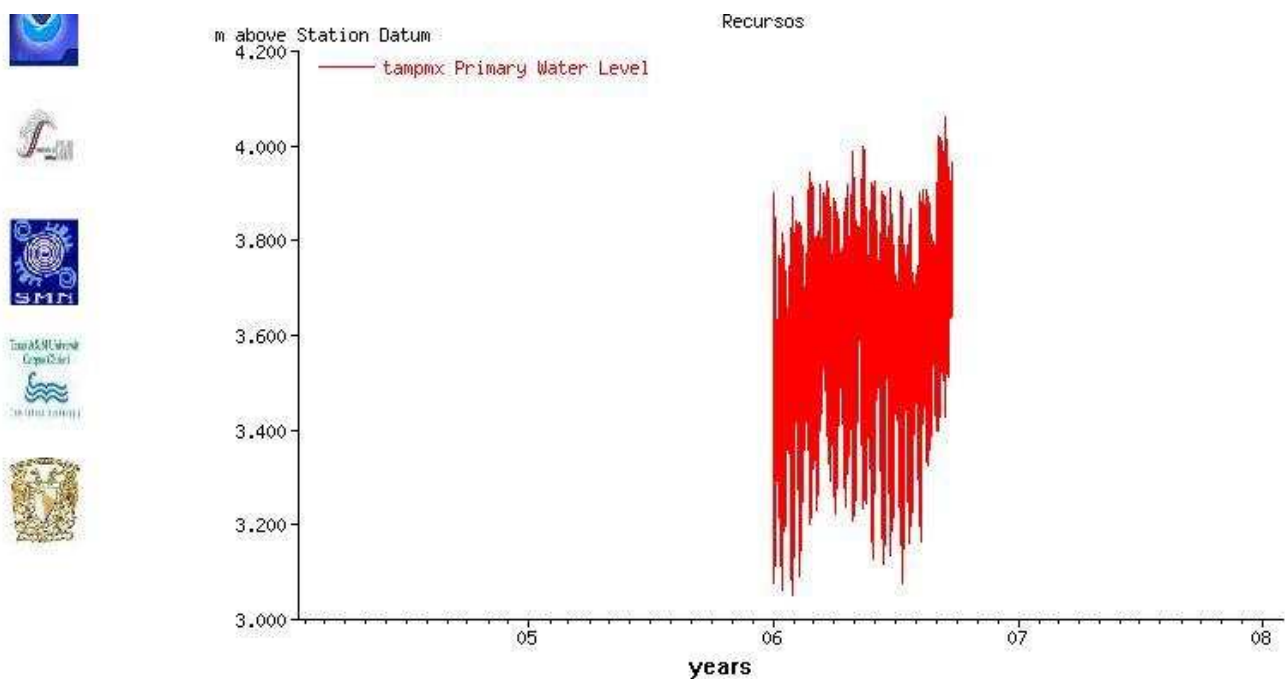


Figura 4.24: Interfaz Web que muestra en forma de gráfica los datos extraídos de la estación mareográfica de Tampico Tamaulipas.

La figura 4.25 muestra los resultados obtenidos del proceso de Consulta Distribuida en forma de texto en la Interfaz Web.

```
#pmdata: data: started 2008069+1702/UTC
#pmdata: DISCLAIMER: The data described below have been collected
#pmdata: from automated equipment and are furnished "as is". PDB makes
#pmdata: no warranties (including no warranties as to merchantability
#pmdata: or fitness) either expressed or implied with respect to the
#pmdata: data or its fitness for any specific application.
#pmdata: 128-pwl: Tampico Mexico (25.8500,97.2700)
#pmdata: 128-pwl: Primary Water Level (m)
#pmdata: 128-pwl: Elevations relative to Station Datum (stnd)
#pmdata: 128-pwl: dates=2003061+0000,2008070+0000/UTC tz=UTC0
#sxext: started 2008069+1702/UTC
#sxext: database=pharosdb
#sxext: 128-pwl: dates=2003061+0000,2008070+0000/UTC
#sxext: 128-pwl: srclist=*,nes,nws
#sxext: 128-pwl: tz=UTC0 datefmt=%Y%j+%H%M fmt=%.3f mult=0.001 offs=0
#sxext: 128-pwl: na=NA rm=RM vperrow=10 gnuplot=0 pdl=0

128 2006001+0000 pwl 3.557 3.575 3.589 3.614 3.619 3.616 3.627 3.635 3.654 3.686
128 2006001+0100 pwl 3.715 3.729 3.737 3.733 3.738 3.733 3.734 3.747 3.745 3.754
128 2006001+0200 pwl 3.764 3.764 3.779 3.794 3.805 3.807 3.801 3.796 3.801 3.802
128 2006001+0300 pwl 3.820 3.829 3.835 3.849 3.858 3.878 3.894 3.894 3.882 3.877
128 2006001+0400 pwl 3.863 3.866 3.857 3.856 3.849 3.851 3.848 3.851 3.856 3.867
128 2006001+0500 pwl 3.881 3.894 3.882 3.872 3.864 3.861 3.870 3.875 3.865 3.850
128 2006001+0600 pwl 3.857 3.852 3.850 3.846 3.866 3.874 3.874 3.869 3.864 3.853
128 2006001+0700 pwl 3.855 3.850 3.836 3.836 3.828 3.827 3.827 3.836 3.837 3.833
128 2006001+0800 pwl 3.839 3.849 3.853 3.848 3.827 3.836 3.824 3.844 3.849 3.860
128 2006001+0900 pwl 3.860 3.862 3.856 3.844 3.832 3.834 3.844 3.861 3.870 3.860
128 2006001+1000 pwl 3.849 3.828 3.831 3.840 3.849 3.864 3.882 3.880 3.884 3.886
128 2006001+1100 pwl 3.885 3.875 3.866 3.880 3.895 3.900 3.908 3.910 3.909 3.906
```

Figura 4.25: Interfaz Web que muestra en forma de texto los datos extraídos de la estación mareográfica de Tampico Tamaulipas.

Capítulo 5

Pruebas

Este capítulo contiene la descripción de las pruebas, las cuales incluyen la descripción del ambiente y condiciones de prueba realizadas a la implementación de la arquitectura.

5.1 Objetivos de las pruebas

El objetivo principal de las pruebas fue comprobar la funcionalidad de la herramienta en los siguientes aspectos.

- La capacidad de la herramienta de software para acceder a los recursos oceanográficos distribuidos en los nodos ROPVO a través de Internet.
- Verificar que la herramienta de software permite la búsqueda remota de los metadatos distribuidos en diferentes nodos ROPVO.
- Verificar que la herramienta de software permite la consulta remota de los recursos de las estaciones mareográficas de los nodos ROPVO.

5.2 Descripción del Ambientes de Pruebas

Las siguientes pruebas principalmente se realizaron en la Intranet del CICATA Unidad Altamira Ubicado en Altamira, Tamaulipas Mexico y otras se realizaron en el Internet. Ambos ambientes de pruebas se mencionan a continuación.

La figura 5.1 muestra el ambiente de pruebas Uno, que incluye la computadora A y el Nodo ROPVO A. En el Nodo A se encuentran los datos de las estaciones mareográficas de Mezquital, La Pesca y Tampico. Las comunicaciones entre las computadoras se realiza en la Intranet instalada en el CICATA. La computadora A es una HP Compaq y posee un procesador Intel Core 2 Duo E4400 a 2.0 Ghz, memoria de 1 GB, disco duro de 160 GB, tarjeta de red 10/100, monitor LCD de 17 pulgadas y sistema operativo Windows XP. La computadora A descarga la Interfaz de Búsqueda y Consulta Distribuida del Nodo ROPVO A. El nodo ROPVO A es una computadora Intel y posee un procesador Intel Pentium 4 a 2.26 Ghz, memoria Ram de 256 MB, tarjeta de red 10/100, disco duro de 40 GB, tarjeta madre Intel, monitor de 17 pulgadas y sistema operativo Red Hat Linux 9.0. El nodo ROPVO A, contiene la implementación de ASBUCORDI y el sistema de información Pharos. Para conocer la instalación del sistema Pharos checar [35], que almacena los recursos oceanográficos que desea compartir el nodo ROPVO A.

La figura 5.2 muestra el ambiente de pruebas Dos, que incluye la computadora A y los nodos ROPVO A y B. En el Nodo B se encuentran los datos de la estación mareográfica de Veracruz. Las comunicaciones entre las computadoras se realiza en la Intranet instalada en el CICATA. El ambiente de pruebas Dos es similar al ambiente de pruebas mostrado en la figura 5.1, la diferencia radica en que el ambiente de pruebas Dos contiene un Nodo adicional. El nodo ROPVO B es una computadora Compaq Presario con procesador Pentium 4 a 1.8 Ghz, memoria de 512MB en Ram, tarjeta de red 10/100, disco duro de 80 GB, tarjeta madre Intel, monitor de 17 pulgadas y sistema operativo Red Hat Linux 9.0. El Nodo ROPVO B, contiene la implementación de ASBUCORDI y el sistema de información Pharos, que almacena los recursos oceanográficos que desea compartir el Nodo B.

La figura 5.3 muestra el ambiente de pruebas Tres, que incluye la computadora A y los nodos ROPVO A,B y C. En el Nodo C se encuentran los datos de la estación mareográfica de Tuxpan. Las comunicaciones entre los nodos se realiza en la Intranet instalada en el CICATA y la computadora A se comunica con los nodos ROPVO a través de Internet. El ambiente de pruebas Tres es similar al ambiente de pruebas mostrado en la figura 5.2, la diferencia radica en que el ambiente de pruebas Tres contiene un nodo adicional, y que la computadora A no se encuentra en la red del CICATA. El nodo C es una computadora Intel con procesador Pentium 4 a 2.6 Ghz, memoria de 512MB en Ram, tarjeta de red 10/100, disco duro de 200 GB, tarjeta madre Intel, monitor de 17 pulgadas y sistema operativo Red Hat Linux 9.0. El Nodo ROPVO C, contiene la implementación de ASBUCORDI y el sistema de información Pharos, que almacena los recursos oceanográficos que desea compartir el nodo ROPVO C.

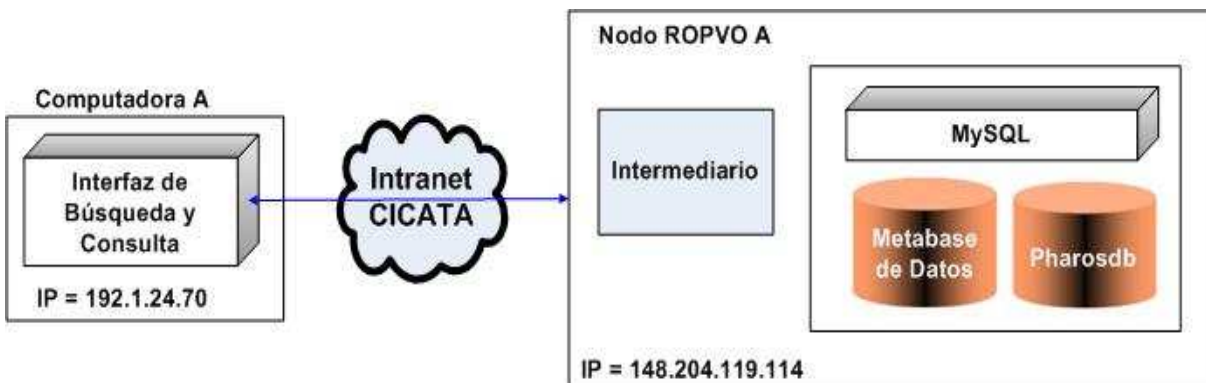


Figura 5.1: Ambiente de pruebas número Uno.

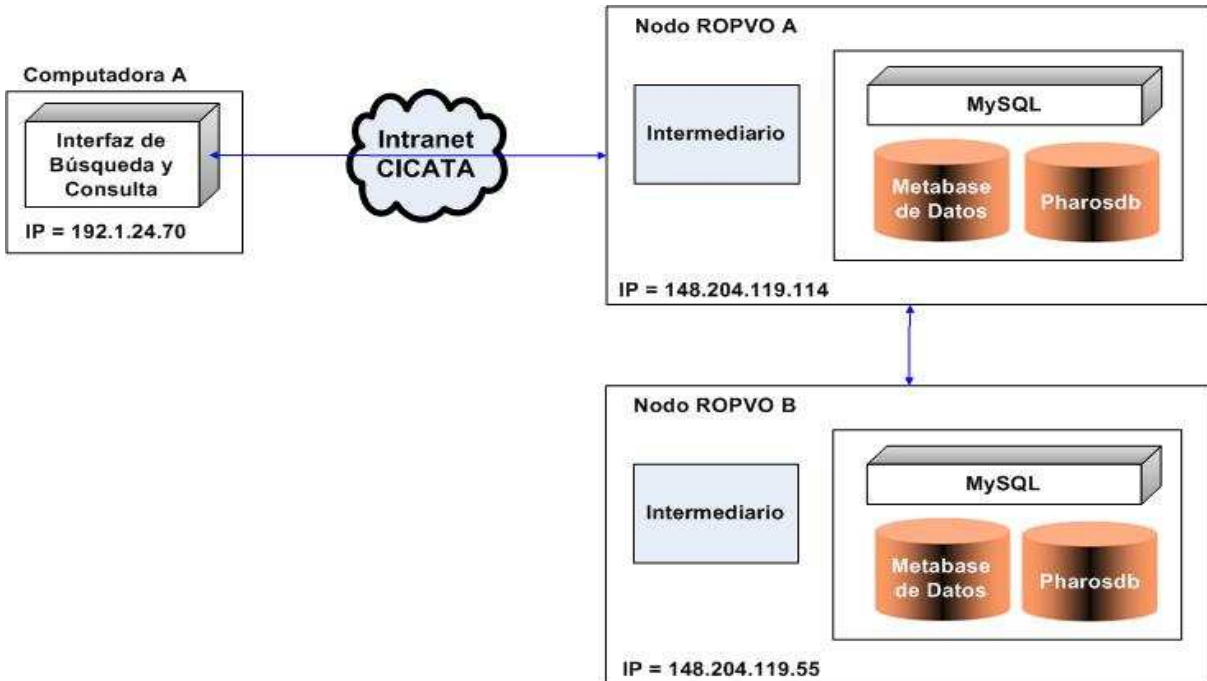


Figura 5.2: Ambiente de pruebas número Dos.

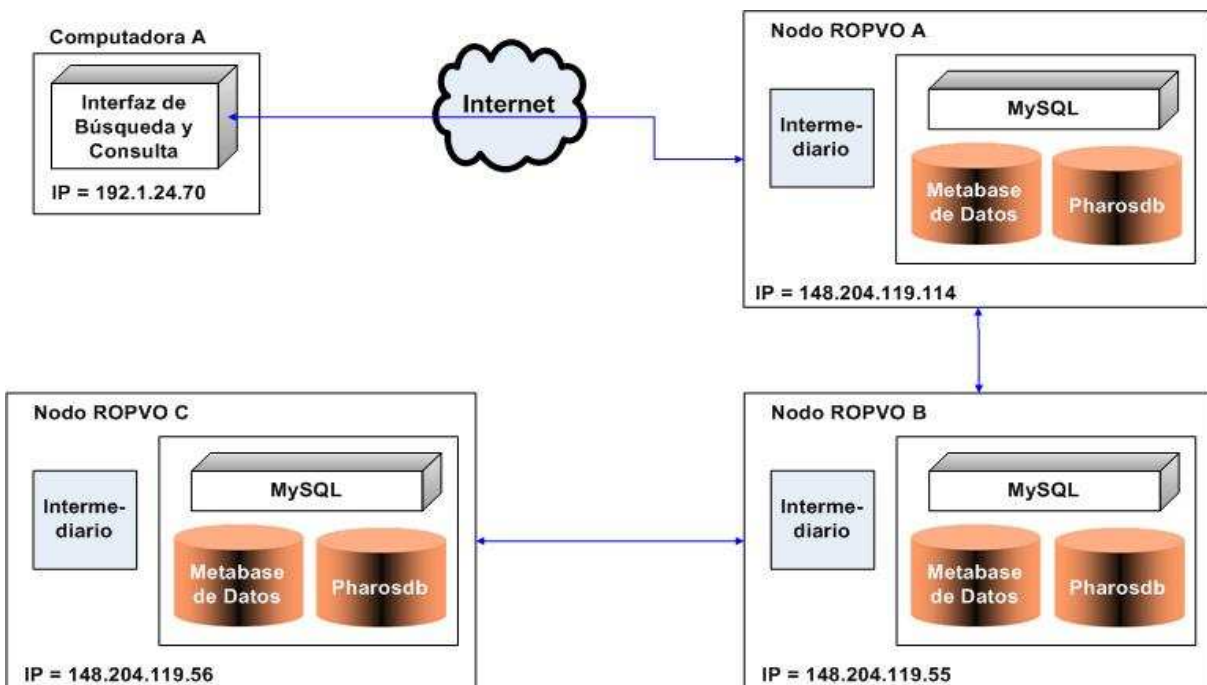


Figura 5.3: Ambiente de pruebas número Tres.

5.3 Casos de Prueba.

Como se describe en el paso anterior se presenta en total tres casos de prueba, con los que se comprueba la funcionalidad de la herramienta ASBUCORDI.

5.3.1 Prueba No. 1.

Probar que la herramienta de software efectúa Búsquedas y Consultas de recursos sobre un nodo ROPVO dentro de la red del CICATA.

Ejecución de la prueba. Para esta prueba se utilizó el ambiente de pruebas Uno, en donde tenemos un usuario conectado al Nodo ROPVO A, figura 5.1. En el Nodo A se encuentran los datos de las estaciones de Mezquital, La Pesca y Tampico. Primeramente se accede a un nodo ROPVO a través de un navegador Web, escribiendo en el URL del explorador "http://148.204.119.55/MBBusquedaDistribuida". Enseguida se procede a escribir en la caja de texto titulado descripción del recurso "primer nivel del mar", figura 5.4. Después se procede a presionar el botón Buscar, esta acción mostrará la Interfaz de Resultados de Búsqueda.

CICATA-IPN Altamira Red de Observaciones ROPVO-GM > ROPVO > Busqueda Distribuida de Recursos Email CICATA Español

Busqueda Distribuida de Recursos

28 May 2008 00:43 UTC

ROPVO-GM
TCOON
ROPVO
Datos
Predicción
Olas
Recursos

Diccionario de Estaciones
Diccionario de Recursos
Comparticion de Recursos Oceanograficos

Descripción del recurso

primer nivel del mar

Localizar Buscar Resultado

Figura 5.4: Interfaz de Búsqueda de Recursos Distribuidos sobre Internet.

La Interfaz Web de Resultados de la Búsqueda muestra al usuario los recursos extraídos del nodo A, figura 5.5. Enseguida el usuario procede a marcar con un click las cajas de verificación de las estaciones mareográficas de Mezquital, Tampico y Pesca. Enseguida presiona el botón Consultar Recursos y la herramienta muestra al usuario la Interfaz Web de Consulta de Recursos Distribuidos.



Figura 5.5: Interfaz que muestra los resultados de la Búsqueda de Recursos Distribuidos sobre Internet.

La Interfaz Web de Consulta de Recursos Distribuidos permite al usuario definir los atributos de la solicitud de Consulta que se aplicarán en el proceso de Extracción de los recursos de las estaciones de Mezquital, Pesca y Tampico. Ahora bien, el usuario procede a formular la solicitud de consulta en la Interfaz de Consulta. En el campo "Dates" escribe "01/23/2004-01/30/2008", en el campo "Format" selecciona la opción "Gif", en el campo "Elevation" selecciona la opción "Station Datum", en el campo "Time Zone" selecciona la opción "Coordinated Universal Time(UTC)". Enseguida presiona el botón titulado "Click here to retrieve data" quien mostrará la Interfaz Web de Resultados de la Consulta, figura 5.6. En la figura 5.7 se muestra la Interfaz Web de Resultados de la consulta distribuida en forma de gráfica, gif.

Basic Query Parameters

Dates: 01/23/2004-01/30/2008

Format: GIF Text Rows PostScript Columns

Units: Metric DNR English

Elevation: Station Datum

Date Format: Entered below (default if blank)

Time Zone: Coordinated Universal Time (UTC)

SSList:

[Click here to retrieve data.](#)

Graph Options

Labels: Title: Recursos

Figura 5.6: Interfaz Web para la Consulta de Recursos Distribuidos.

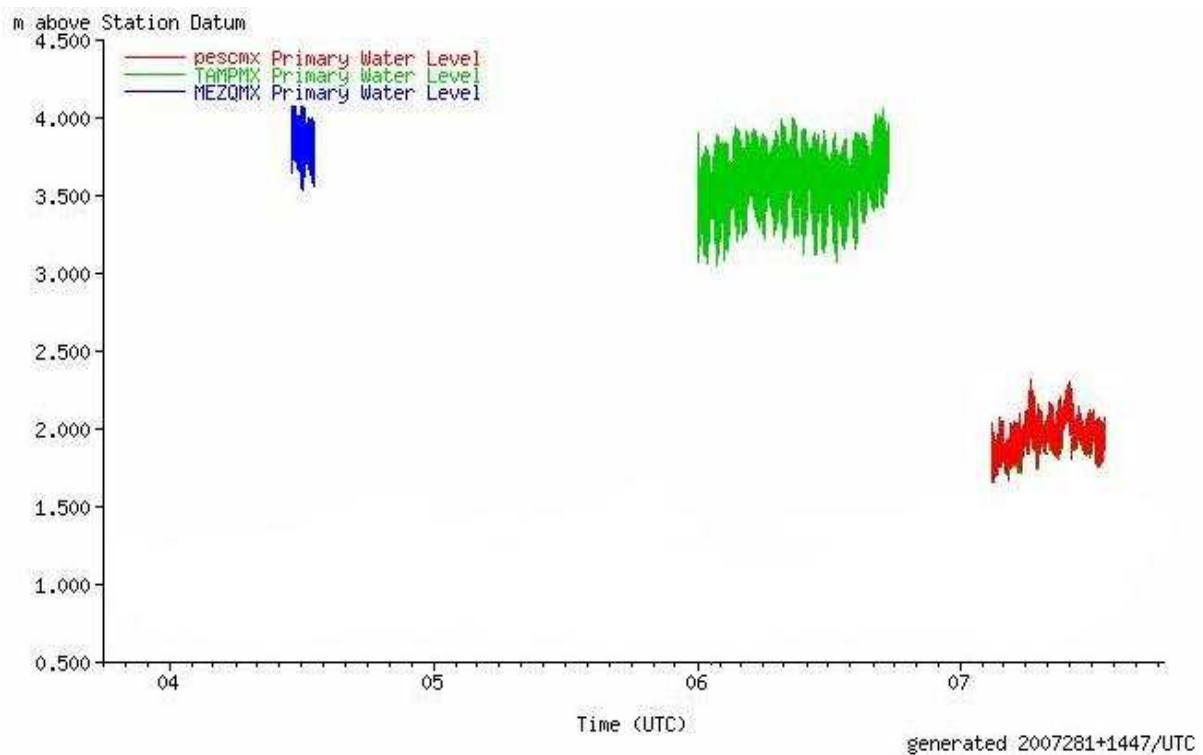


Figura 5.7: Gráfica que muestra los datos extraídos de las estaciones de Mezquital, la Pesca y Tampico.

5.3.2 Prueba No. 2.

Probar que la herramienta de software efectúa Búsquedas y Consultas de recursos sobre dos nodos ROPVO dentro de la red del CICATA.

Ejecución de la prueba. Para esta prueba se utilizó el ambiente de pruebas Dos, en donde tenemos dos Nodos ROPVO conectados entre sí, figura 5.2. En el Nodo A se encuentran los datos de las estaciones de Mezquital, La Pesca y Tampico, En el Nodo B se encuentran los datos de las estaciones de Veracruz. El ejemplo para la búsqueda de recursos fue "primer nivel del mar", figura 5.4. La Interfaz Web de Resultados de la búsqueda muestra al usuario los metadatos de las estaciones de Mezquital, La Pesca, Tampico y Veracruz extraídos de los nodos ROPVO A y B, figura 5.8.



Página Web de la estación de Veracruz
Página Web de la estación mareográfica de Veracruz
<http://148.204.119.55/overview/730>

Primer nivel del mar en la pesca
Datos sobre el primer nivel del mar en la pesca
148.204.119.114

Página Web de la Pesca
Página Web de la Pesca
<http://148.204.119.114/overview/139>

Datos de la estación de Veracruz
Datos de la estación mareográfica de Veracruz
148.204.119.55

Datos de Mezquital
Datos del primer nivel del mar en la estación mareográfica del Mezquital
148.204.119.114

nivel del mar en tampico
Datos sobre el nivel del mar en tampico tamaulipas
148.204.119.114

Página Web de Tampico
Página Web de la estación mareográfica de Tampico, tamaulipas
<http://148.204.119.114/overview/128>

Consultar recursos

Figura 5.8: Interfaz que muestra los resultados de la Búsqueda de Recursos Distribuidos sobre Internet.

El usuario procede a marcar con un click las cajas de verificación de las estaciones mareográficas de

Mezquital, La Pesca, Tampico y Veracruz. Continuando con el procedimiento presiona el botón Consultar Recursos que muestra la Interfaz Web de Consulta de Recursos Distribuidos al usuario. Ahora bien, el usuario del sistema formula la solicitud de búsqueda de recursos en base a los parámetros mostrados en la figura 5.6. En este punto se realiza la consulta a múltiples Bases de Datos y se integran los resultados para presentarlos en la Interfaz de Resultados de la Consulta como una gráfica gif, figura 5.9.

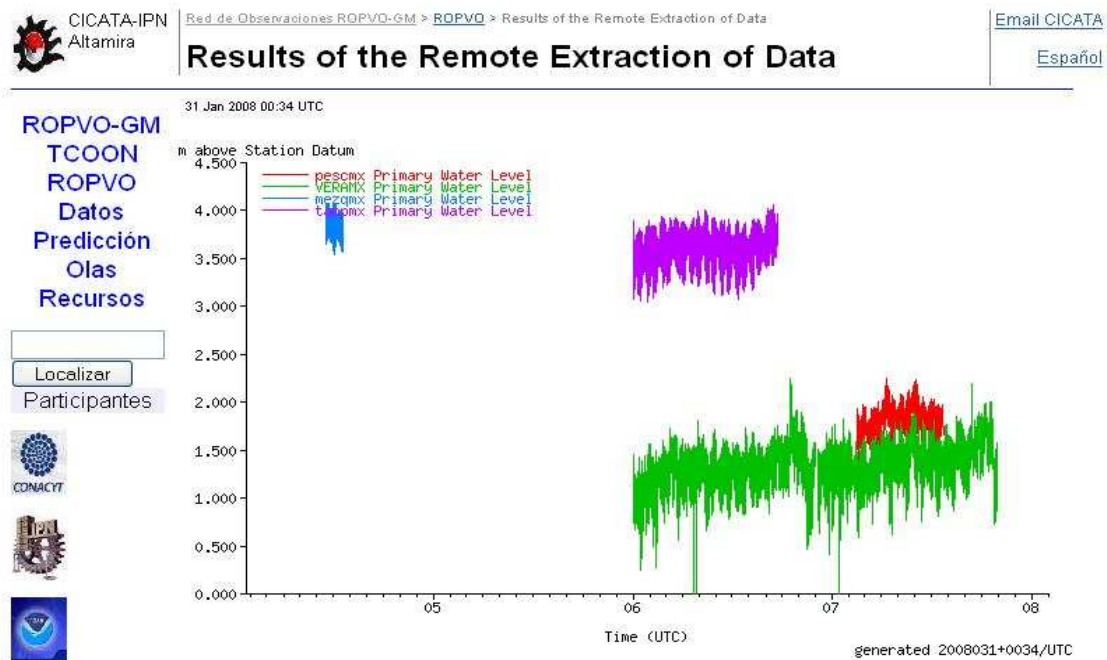


Figura 5.9: Gráfica que muestra los datos extraídos de las estaciones de Mezquital, la Pesca, Tampico y Veracruz.

5.3.3 Prueba No. 3.

Probar que la herramienta de software efectúa Búsquedas y Consultas de recursos utilizando tres nodos ROPVO a través de Internet.

Ejecución de la prueba. Para esta prueba se utilizó el ambiente de pruebas número Tres, en donde tenemos tres nodos ROPVO conectados entre sí, figura 5.3. En el nodo A se encuentran los datos de las estaciones de Mezquital, La Pesca y Tampico, En el nodo B se encuentran los datos de las estaciones de Veracruz y en el nodo C se encuentran los datos de las estaciones de Tuxpan. El ejemplo para la búsqueda de recursos fue "primer nivel del mar", figura 5.4. Enseguida la Interfaz Web de Resultados de la búsqueda muestra al usuario los metadatos de las estaciones de Mezquital, La Pesca, Tampico, Tuxpan y Veracruz extraídos de los nodos ROPVO A,B y C, figura 5.10.

Enseguida el usuario procede a marcar con un click las cajas de verificación de las estaciones mareográficas de Mezquital, La Pesca, Tampico, Tuxpan y Veracruz. Posteriormente presiona el botón Consultar Recursos y se muestra al usuario la Interfaz Web de Consulta de Recursos Distribuidos. Ahora bien, el usuario del sistema formula la solicitud de búsqueda de recursos en base a los parámetros mostrados en la figura 5.6. Finalmente se realiza la Consulta a múltiples Bases de Datos y se integran los resultados para presentarlos en la Interfaz de Resultados de la Consulta como una gráfica gif, figura 5.11.

5.4 Otras Pruebas.

Se realizaron las tres mismas pruebas con salida a archivos. Sin embargo los listados no se incluyen para evitar muchas figuras.



Pagina Web de la estacion de Tuxpan
Pagina Web de la estacion mareografica de Tuxpan
<http://148.204.119.56/overview/720>

Datos de Tuxpan
Datos de la estacion mareografica de tampico, tamaulipas
148.204.119.56

Pagina Web de la estacion de Veracruz
Pagina Web de la estacion mareografica de Veracruz
<http://148.204.119.55/overview/730>

Primer nivel del mar en la pesca
Datos sobre el primer nivel del mar en la pesca
148.204.119.114

Pagina Web de la Pesca
Pagina Web de la Pesca
<http://148.204.119.114/overview/139>

Datos de la estacion de Veracruz
Datos de la estacion mareografica de Veracruz
148.204.119.55

Datos de Mezquital
Datos del primer nivel del mar en la estacion mareográfica del Mezquital
148.204.119.114

nivel del mar en tampico
Datos sobre el nivel del mar en tampico tamaulipas
148.204.119.114

Pagina Web de Tampico
Pagina Web de la estacion mareografica de Tampico, tamaulipas
<http://148.204.119.114/overview/128>

[Consultar recursos](#)

Figura 5.10: Interfaz que muestra los resultados de la Búsqueda de Recursos Distribuidos sobre Internet.

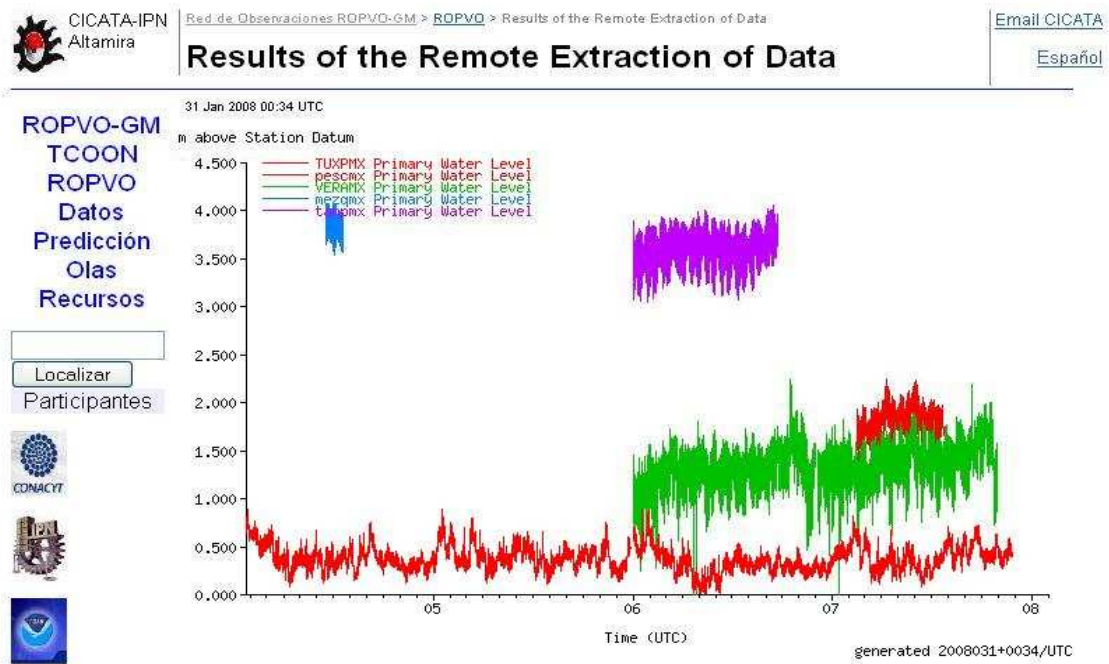


Figura 5.11: Gráfica que muestra los datos extraídos de las estaciones de Mezquital, la Pesca, Tampico, Tuxpan y Veracruz.

Capítulo 6

Conclusión y discusión

6.1 Conclusión

La planeación y desarrollo del proyecto de investigación demostró la importancia del esquema de metadatos estandar FGDC, como método que permite la organización e identificación de los recursos oceanográficos compartidos en los nodos ROPVO mediante un mecanismo de búsqueda automatizado, que al ser implementado en una metabase de datos hace posible la compartición y extracción de recursos oceanográficos distribuidos a través de Internet. Para la adaptación correcta del estandar FGDC a nuestro esquema de metadatos propuesto en la sección 4.1.1, fue necesario agregar algunos metadatos particulares como *idstation*, *idresource*, *station_name* y *relation*.

La comunidad ROPVO ahora cuenta con un mecanismo que le permite compartir recursos oceanográficos a través de Internet. Estos recursos se encuentran almacenados en diferentes Bases de Datos. Los recursos de las instituciones que participan en la red ROPVO pueden almacenar y compartir recursos que se encuentran almacenados en texto, imágenes, video, sonido, páginas Web o bases de datos. Los científicos del área de ingeniería oceánica y costera ahora cuentan con una herramienta poderosa para la búsqueda y consulta de recursos oceanográficos distribuidos sobre Internet.

La herramienta de búsqueda y consulta de recursos distribuidos es un medio de consulta de datos

oceanográficos útil para la sociedad en general, ya que apoya a las instituciones encargadas de la seguridad en los puertos y protección civil al informarles del clima oceánico del Golfo de México en tiempo casi real. Gracias a esta nueva tecnología, las Instituciones interesadas en compartir recursos oceanográficos tienen la posibilidad de compartir su información a través de Internet.

Los resultados exitosos obtenidos de nuestra herramienta en el proceso de búsqueda y consulta de recursos oceanográficos distribuidos en la comunidad ROPVO, ver sección 5.1. demostró que es factible el desarrollo de aplicaciones distribuidas mediante el uso de software libre.

La implementación de ésta herramienta diseñada bajo una arquitectura peer-to-peer, demostró ser una herramienta factible para el intercambio de recursos oceanográficos distribuidos en la comunidad ROPVO, además esta arquitectura permitió reducir los costos monetarios en equipo de cómputo, aproximadamente a \$1,200.00 por cada nodo conectado a la comunidad. Debido a su naturaleza, nuestra herramienta demostró ser un sistema altamente tolerante a fallos a pesar de la actividad intermitente de los nodos de la comunidad ROPVO, evitando así la caída total del sistema.

El empleo de *cookies* fue de gran importancia en el desarrollo de este proyecto ya que fue el medio utilizado para el paso de parámetros entre páginas Web. La utilización de la interfaz de comunicación *socket* fue la clave para la comunicación entre aplicaciones de software realizadas en diferentes lenguajes de programación.

El SMBD MySQL demostró ser una herramienta eficaz para el almacenamiento de los metadatos y recursos de la comunidad ROPVO. Sin embargo, por defecto las bases de datos creadas por MySQL no cuentan con el permiso de acceso remoto desde otras computadoras, por lo que fue necesario establecer estos permisos de acceso mediante los comandos *setpasswordfor*, quien permite establecer la contraseña de acceso a base de datos y *grantallprivilegeson*, quien permite el acceso al usuario desde una computadora remota.

Finalmente, el sistema Pharos desarrollado por DNR[3], es ahora ampliado con un mecanismo que le permite realizar extracciones y consultas de datos oceanográficos distribuidos sobre Internet.

6.2 Alcances Logrados

La implementación de la Arquitectura de Software actualmente puede crecer hasta soportar 10 nodos simultáneamente por cada nodo conectado a la red ROPVO. Sin embargo el administrador del sistema puede incrementar o decrementar el número de nodos participantes en la red ROPVO sin que esto afecte el desempeño del sistema. Esto se logra cambiando el valor del parámetro "MaxConnection = 10", quien contiene el número máximo de nodos permitidos para conectarse a un nodo ROPVO. Este parámetro se localiza en el archivo de configuración *.mutella/mutellarc*.

Actualmente la metabase de datos soporta cinco recursos oceanográficos que son: Primer Nivel del Mar, Temperatura del Mar, Salinidad y Velocidad Promedio, sin embargo es posible incrementar los tipos de recursos que soporte el sistema. Para esto, la Interfaz de Búsqueda de recursos cuenta con el módulo Diccionario de recursos que permite al administrador del sistema agregar, modificar o borrar los nombres de los recursos que soporte el sistema.

En estos momentos la implementación de la arquitectura de software opera solamente bajo el sistema operativo Red Hat Linux 9.0. El equipo de hardware empleado requiere, para un óptimo funcionamiento como mínimo, un procesador pentium 4 a 2.5. Ghz, memoria de 1 GB en Ram, tarjeta de red 10/100, tarjeta madre Intel y monitor de 17 pds.

Por el momento las conexiones entre los nodos ROPVO se realizan de un modo manual. Esto quiere decir, que el administrador del sistema necesita ingresar al Componente de Comunicaciones Mutella y ejecutar la línea de comando open "Dirección IP:Puerto"; donde el parámetro "Dirección IP:Puerto" corresponde a la dirección IP y puerto del Componente de Comunicaciones Mutella perteneciente al nodo

ROPVO con quien se desea establecer comunicación, ver sección 3.2. Por tal motivo se requiere desarrollar un mecanismo automático que realice las actividades mencionadas anteriormente y que permita al sistema establecer conexión con cada uno de los nodos ROPVO.

6.3 Trabajos Futuros

Desarrollar un software tipo Web cache, que contenga un mecanismo que permita localizar automáticamente nuevos nodos en la red ROPVO. El software deberá contener un archivo llamado Host, que almacene las direcciones IP de los nodos ROPVO. Cuando el administrador del sistema ejecute la herramienta, esta establecerá comunicación con el software Web cache que devolverá las direcciones IP de los nodos ROPVO disponibles para el intercambio de recursos. Enseguida con la herramienta de software presentada en este trabajo se procederá a establecer conexión con cada uno de los nodos devueltos por el sistema Web cache iniciándose con el proceso de intercambio de recursos.

Actualmente la instalación de la herramienta ASBUCORDI en computadoras con Red Hat Linux 9.0 se realiza manualmente por lo que es necesario diseñar e implementar un software para la instalación automática de la herramienta. El software se encargará de instalar los Procesos de Búsqueda y Consulta distribuida, además de instalar el sistema peer to peer mutella, encargado de la comunicación entre los nodos ROPVO.

En estos momentos la herramienta de software ASBUCORDI opera bajo el sistema operativo Red Hat Linux 9.0, por lo que otro trabajo interesante sería emigrar la herramienta de software al Sistema Operativo Windows (SOW). Para lograr esto, es primeramente necesario instalar en la máquina con SOW el ambiente de desarrollo de la herramienta, que son: el servidor Web Apache, la herramienta para el desarrollo de páginas Web dinámicas WIKI y el lenguaje de programación para Internet PHP y PERL. Enseguida se tendría que verificar la compatibilidad del código de la herramienta con el ambiente de desarrollo de sitios web para Windows. Finalmente se tendrían que realizar las modificaciones necesarias al código de

la herramienta para lograr una migración exitosa al SOW.

Una vez emigrado la herramienta de software ASBUCORDI al SOW, otro trabajo a futuro consistiría en implementar el software de instalación automática de la herramienta para el SOW.

Bibliografía

- [1] Comisión Oceanográfica Intergubernamental, 2002, "El desarrollo sostenible y la Comisión Oceanográfica Intergubernamental de la UNESCO", 30 pp.
- [2] <http://lighthouse.tamucc.edu/TCOON>, Texas Coastal Ocean Observation Network (TCOON).
- [3] <http://dnr.cbi.tamucc.edu/TCOON/HomePage>, Texas Coastal Ocean Observation Network (TCOON).
- [4] Meza E., Salles P., Figueroa J., Silva Rodolfo, 2003 "Red de Observaciones y Predicciones de Variables Oceánicas en las Costas y Puertos del Golfo de México (ROPVO-GM)".
- [5] Fabiola Rasgado Celaya, 1999, "Herramienta para Consultas Basadas en Ejemplos (QBE) para una Base de Datos en Internet", Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca Morelos.
- [6] May Arrijoja, Alfonso de los Angeles, 2000, "Herramienta para Consultas Basadas en Ejemplos (QBE) para Multibases de datos en Internet". Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca Morelos.
- [7] George Coulouris, Jean Dollimore, Tim Kindberg, 2005, "Sistemas Distribuidos Conceptos y Diseño", Editorial Addison Wesley, Tercera Edición, 287 pp.
- [8] http://www9.limewire.com/developer/gnutella-protocol_0.4.pdf, 2001, "Gnutella: The gnutella protocol specification v.0.4"

- [9] Carmen de Pablos Heredero, José Joaquín López-Hermoso Aguius, "Dirección y gestión de los sistemas de información en la empresa una visión Integrada". Segunda Edición, Universidad Rey Juan Carlos, 342 pp.
- [10] Ramón Jesús Millán Tejedor, 2007, "Domine las Redes (P2P) "Peer-to-Peer", Editorial Alfaomega, Segunda Edición, 330 pp.
- [11] Ming-Hsiang Tsou, 2002, "An operacional Metadata Framework for Searching, Indexing, and Retrieving Distributed Geographic Information Services on the Internet" Departament of Geography, San Diego State University, mtsou@mail.sdsu.edu.
- [12] Paul Clements. "A Survey of Architecture Description Languages". Proceedings of the International Workshop on Software Specification and Design, Alemania, 1996.
- [13] David Garlan y Mary Shaw., 1994, "An introduction to software architecture. CMU Software Engineering Institute Technical Report", CMU/SEI-94-TR-21, ESC-TR-94-21.
- [14] Mary Shaw y David Garlan.,1996, "Software Architecture: Perspectives on an emerging discipline", Upper Saddle River, Prentice Hall.
- [15] Andrew S. Tanenbaum, Roberto Escalona García, 2004 "Sistemas Operativos Modernos", Editorial Prentice Hall, Segunda edición, 951 pp.
- [16] Peter Rob, Carlos Coronel, 2002 "Sistemas de Bases de Datos, Diseño, Implementación y Administración", Editorial Prentice Hall, Quinta edición, 838 pp.
- [17] Edelstein, Herb., 1994 "Unraveling Client/Server Architecture." 34 pp.
- [18] <http://www.openp2p.com/>, Clay Shirky, 2000, "What is P2P ... And What is not."
- [19] José Antonio Mañas, Rafael Fernández Calvo, 2003, "Mundo IP, Introducción a los Secretos de Internet y las Redes de Datos", Editorial nowtilus, 360 pp.

- [20] Manuel G. Velasquez, 2006, "Ética en los Negocios, Conceptos y Casos", Sexta edición, Editorial Prentice Hall, 448 pp.
- [21] <http://www.gnutella.com/>, "Gnutella"
- [22] Wally Wang, 2004 "Steal this File Sharing Book: What They Won't Tell You about File", 273 pp.
- [23] Gonzalo Asensio, 2006, "Seguridad en Internet. Una guía práctica y eficaz para proteger su PC con software gratuito", editorial nowtilus, 318 pp.
- [24] <http://setiathome.ssl.berkeley.edu/download.com/>, SETI@Home, "Búsqueda de Inteligencia Extraterrestre"
- [25] <http://www.stanford.edu/group/pandegroup/folding/>, Folding@Home, "Comp. distribuida"
- [26] <http://www.stanford.edu/group/pandegroup/gnome/>, Genome@Home, "The Human Genome Project"
- [27] Ahmar Abbas, 2005, "Grid Computing: A Practical Guide to Technology and Applications", 408 pp.
- [28] Belázquez, F., Cabero, J. y Loscertales, 1994, "Nuevas tecnologías de la Información y la Comunicación para la Educación." Ediciones Alfar, págs. 114-121.
- [29] Gunter Born, 2001, "Compendium HTML: Con XHTML, DHTML, CSS, XML, XSL y WML", Editorial marcombo, 974 pp.
- [30] http://www9.limewire.com/developer/gnutella_protocol.0.4.pdf, LimeWire Site, "The Gnutella Protocol Specification v0.4, Document Revision 1.2"
- [31] <http://mutella.sourceforge.net/>, "Sistema peer to peer Mutella"
- [32] Ingo Lackerbauer, 2000, "Todo sobre Internet: Completo, Claro y Conciso", Editorial marcombo, 215 pp.
- [33] Amit Sheth, Miltiadis D. Lytras, 2007 "Semantic Web-based Information Systems: State-of-the-art Applications", 317 pp.

- [34] <http://www.niso.org>, 2004, "Understanding Metadata, National Information Standards Organization"
- [35] Manuel Borrego, Eustorgio Meza, Rogelio Ortega, 2005 "Diseño y Desarrollo de un Portal Web para un Sistema de Observación Oceanográfico en el Golfo de México", CICATA-IPN. 77 pp.
- [36] F.J. Ceballos, 2006 "Java 2: Lenguaje y aplicaciones", Editorial Rama, 215 pp.