



INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

**BIBLIOTECA DIGITAL
BASADA EN
SERVICIOS WEB**

T E S I S

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA
COMPUTACIÓN

PRESENTA
LIC. OSCAR FABELA CANO

DIRECTOR:
M. EN C. RUBÉN PEREDO VALDERRAMA



México D.F. Diciembre de 2006



INSTITUTO POLITECNICO NACIONAL SECRETARIA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 17:00 horas del día 30 del mes de Junio de 2006 se reunieron los miembros de la Comisión Revisora de Tesis designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis de grado titulada:

“BIBLIOTECA DIGITAL BASADA EN SERVICIOS WEB”

FABELA

Apellido paterno

CANO

materno

OSCAR

nombre(s)

Con registro:

B	0	1	1	4	1	0
---	---	---	---	---	---	---

aspirante al grado de: **MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

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

LA COMISIÓN REVISORA

Presidente

Dr. Felipe Rolando Menchaca García
Primer vocal
(Director)

Secretario

M. en C. Gilberto Lorenzo Martínez Luna
Segundo vocal

M. en C. Ruben Peredo Valderrama

M. en C. Sandra Dinora Orantes Jiménez

Tercer Vocal

M. en C. Alejandro Botello Castillo

EL PRESIDENTE DEL COLEGIO

Dr. Hugo César Coyote Estrada
DIRECCION





INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA LICENCIAMIENTO DE DERECHOS

En la Ciudad de México, Distrito Federal, el día 17 del mes de Noviembre del año 2006 el que suscribe Lic. Oscar Fabela Cano, alumno del Programa de Maestría en Ciencias de la Computación, adscrito a la Sección de Estudios de Posgrado e Investigación del CIC, manifiesta que es autor intelectual del presente Trabajo de Tesis bajo la dirección del M. en C. Rubén Peredo Valderrama y autoriza el uso no comercial del trabajo titulado: **Biblioteca Digital Basada en Servicios Web**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, o datos del trabajo sin el permiso expreso del autor. Este puede ser obtenido escribiendo a la siguiente dirección: ofabela@cic.ipn.mx si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Lic. Oscar Fabela Cano

Agradecimientos

*A mí compañero y amigo Ing. Jorge Martínez,
por su ayuda incondicional.*

*Al M. en C. Miguel Contreras Montoya,
por el tiempo dedicado a esas grandes sugerencias y por su valiosa orientación*

*A mi director de tesis, M. en C. Rubén Peredo,
Por su apoyo en la dirección de esta tesis.*

*Al Dr. Hugo César Coyote Estrada,
por todo su apoyo.*

*A los miembros de la Comisión Revisora,
por sus valiosas sugerencias.*

*Al Instituto Politécnico Nacional y al CIC,
por las grandes enseñanzas.*

A todos los que contribuyeron en la realización de este trabajo

Resumen

Actualmente los Servicios Web (en inglés Web Services) están considerados como una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones, estos a su vez pueden ser empleados por distintas aplicaciones de software para intercambiar información a través de la red.

Una de las ventajas más importantes del uso de esta tecnología es la aportación de interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar abiertos como XML, SOAP, WSDL y UDDI.

El propósito general de este trabajo fue desarrollar un conjunto de Servicios Web, que sirven de modelo y proporcionan funcionalidad a una biblioteca digital, la cuál, también fue implementada como parte fundamental de la realización de la tesis.

La biblioteca digital implementada ofrece servicios de administración y acceso remoto a un repositorio centralizado de recursos digitales en especial de publicaciones (artículos de revista, publicaciones en memorias de congresos, reportes técnicos, etc.) a través de una interfaz Web mediante el uso de un navegador convencional (Internet Explorer, Netscape, Mozilla Firefox, etc.). La interfaz permite a los usuarios realizar consultas (por autor, tema, tipo de publicación, año, etc.) y visualizar en el navegador los documentos deseados.

Los Servicios Web pueden aportar gran independencia entre la aplicación que usa el Servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro. Esta flexibilidad será cada vez más importante dado que la tendencia a construir grandes aplicaciones a partir de componentes distribuidos más pequeños es cada día más solicitada.

La propuesta y solución tecnológica desarrollada puede ser reutilizada para la puesta en marcha de otras biblioteca digitales.

Abstract

Currently, Web Services are regarded as a collection of protocols and standards for data interchange among heterogeneous software applications. This interchange is accomplished through a connected data network.

One of the most important advantages of the Web Service technology, is the characteristic interoperability among platforms from different providers, which is realized by the used of open standard protocols such as: XML, SOAP, WSDL and UDDI.

The main purpose of the present work is the development of a set of Web Services that work as a model and provide functionality to a Digital Library, which is also implemented as a core result of this thesis.

The Digital Library offers management services and remote access to a centralized repository of digital resources, with emphasis on publications (journal papers, conference proceedings, technical reports and so on), by using a Web-based interface and conventional browsers (Internet Explorer, Netscape Navigator, Mozilla, Firefox). The interface allows users to request document queries by keywords, author, topic, type of publication and year; the document visualization is accomplished directly on the web browser.

Web services represent a great independence between the application consuming the service and the service itself. In this way, changes on one of them during the course of time are not meant to affect each other. This type of flexibility becomes more and more important considering the trend to build larger applications from distributed and small components.

The developed solution in the present work can be reused for setting up and on other digital libraries.

Tabla de Contenido

RESUMEN	I
ABSTRACT	VI
GLOSARIO.....	1
CAPÍTULO I INTRODUCCIÓN	1
1.1 ANTECEDENTES	6
1.2 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	7
1.3 OBJETIVOS.....	8
1.3.1 <i>General</i>	8
1.3.2 <i>Específicos</i>	9
1.4 LÍMITES Y ALCANCES	9
1.5 BENEFICIOS ESPERADOS	10
1.6 ORGANIZACIÓN DE LA TESIS	10
CAPÍTULO II MARCO TEÓRICO.	12
2.1 BIBLIOTECAS DIGITALES	12
2.1.1 <i>Conceptos Generales</i>	16
2.1.2 <i>Arquitectura General de una Biblioteca Digital</i>	17
2.1.3 <i>Sistemas de Bibliotecas Digitales</i>	18
2.2 SERVICIOS WEB.....	21
2.2.1 <i>Conceptos Generales</i>	21
2.2.2 <i>Utilidad y beneficios de los servicios Web</i>	22
2.2.3 <i>Arquitectura de los Servicios Web</i>	23
2.2.4 <i>Simple Object Access Protocol (SOAP)</i>	24
2.2.5 <i>Web Service Description Lenguaje (WSDL)</i>	26
2.2.6 <i>Universal Description, Discovery and Integration (UDDI)</i>	27
2.2.7 <i>Servicios Web y bibliotecas digitales</i>	27
2.2.8 <i>Tecnologías para desarrollo de Servicios Web</i>	28
CAPÍTULO III ANÁLISIS Y DISEÑO	30
3.1 ANÁLISIS DEL SISTEMA	30
3.1.1 <i>Objetivos del sistema</i>	31
3.1.2 <i>Requerimientos</i>	31
3.1.2.1 <i>Módulo de administración</i>	31
3.1.2.2 <i>Módulo de usuario principal</i>	32
3.1.3 <i>Arquitectura del sistema</i>	33
3.1.3.1 <i>Casos de Uso</i>	34
3.1.3.2 <i>Diagrama de caso de uso general del sistema</i>	34
3.1.3.3 <i>Diagrama de secuencia general del sistema</i>	35
3.1.3.4 <i>Diagrama de caso de uso del usuario principal</i>	36
3.1.3.5 <i>Diagrama de secuencia de usuario principal</i>	38
3.1.3.6 <i>Diagrama de casos de uso del administrador</i>	39
3.1.3.7 <i>Diagrama de secuencia del administrador</i>	40

3.1.5 Diagrama relacional	42
3.1.6 Paquetes	44
3.1.7 Diagramas de clases.....	45
3.1.7.1 Diagrama de clases del modulo de administración	45
3.1.7.2 Diagrama de clases modulo usuario principal	47
CAPÍTULO IV IMPLEMENTACIÓN, PRUEBAS Y RESULTADOS	49
4.1 DESCRIPCIÓN DEL SISTEMA	50
4.1.1 Módulos de administración y usuario principal	50
4.1.1.1 Web Forms (aspx). Modulo de administrador.	51
4.1.1.2 Servicios Web (asmx). Modulo de admistrador	51
4.1.2.1 Web Forms (aspx). Modulo de usuario principal.....	52
4.1.2.2 Servicios Web (asmx). Modulo usuario principal	53
4.1.3 Localización y consumo dinámico de Servicios Web	56
4.1.4 Pruebas y resultados	57
4.1.4.1 Tiempo de recuperación de materiales.	57
4.1.4.2 Consumo dinámico de Servicios Web.....	59
4.1.4.2 Comparativo de funcionalidades con otras bibliotecas.	60
CAPÍTULO V CASO DE USO	61
5.1 MODULO ADMINISTRADOR.....	61
5.1.1 Alta de Recursos	62
5.1.2 Alta de Usuarios	63
5.1.2 Edición, actualización , borrado de usuarios y recursos	64
5.2 MODULO USUARIO PRINCIPAL	65
5.2.1 Buscador Principal.....	66
5.2.2 Mi Buscador y Mi Librero	68
5.2.3 Mis Contactos	70
CAPÍTULO VI CONCLUSIONES Y TRABAJOS FUTUROS	71
6.1 CONCLUSIONES	71
6.2 RESULTADOS FINALES	72
6.3 TRABAJOS FUTUROS	73
BIBLIOGRAFÍA	74
ANEXO A.....	76
AUTENTIFICACIÓN.....	76
ANEXO B	77
FORMA WEB (ASPX).....	77
ANEXO C.....	82
SERVICIO WEB (ASMIX).....	82
ANEXO D.....	84
REQUERIMIENTOS.....	84

Índice de Figuras

Figura 2.1 Componentes funcionales de una biblioteca digital.....	18
Figura 2.2 Arquitectura básica de los servicios Web.	24
Figura 2.3 Estructura de Mensaje SOAP.....	25
Figura 3.1 Arquitectura del Sistema.....	33
Figura 3.4 Diagrama de caso de uso del usuario principal.....	36
Figura 3.6 Diagrama de casos de uso del administrador.....	39
Figura 3.8 Diagrama Relacional.....	42
Figura 3.9 Diagrama de paquetes.....	44
Figura 3.10 Diagrama de clases para "Ambiente Administración".....	45
Figura 3.11 Diagrama de clases para "Ambiente Usuario Principal".....	47
Figura 4.1 Ubicación y consumo dinámico de Servicios Web.....	57
Figura 4.2 Buscador principal del ambiente de usuario principal.....	58
Figura 4.3 Ubicación y consumo dinámico de Servicios Web.....	59
Figura 4.4 Ubicación y consumo dinámico de Servicios Web.....	59
Figura 5.1 Inicio de sesión de administrador.....	61
Figura 5.2 Modulo de Administración.....	62
Figura 5.3 Formulario de registro de recursos o materiales.....	62
Figura 5.4 Formulario de registro (opción Ubicación).....	63
Figura 5.5 Formulario de registro del usuario principal.....	63
Figura 5.6 Formulario "BCUsuarios" Bajas-Cambios (usuarios).....	64
Figura 5.7 Formulario "BCUsuarios" opción actualizar activada.....	64
Figura 5.8 Modulo "Mi Cuenta".....	65
Figura 5.9 Modulo "Buscador Principal".....	66
Figura 5.10 Opción "Mostrar resultados por:".....	67
Figura 5.11 Opción "En Metadato:".....	67
Figura 5.12 Modulo "Mi Buscador".....	68
Figura 5.13 Despliegue de un recurso seleccionado.....	69
Figura 5.14 Modulo "Mi Librero" Un usuario visualiza los registros anteriormente seleccionados.....	69
Figura 5.15 Contactos de un usuario.....	70

Glosario

ADO.NET: *ActiveX Data Objects.NET*. Es una evolución del modelo de acceso a datos de ADO que controla directamente los requisitos del usuario para programar aplicaciones escalables. Se diseñó específicamente para el [Web](#), teniendo en cuenta la escalabilidad, la independencia y el estándar [XML](#).

API: *Application Programming Interface*. Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos API comunes a los que deben ajustarse todos los desarrolladores de aplicaciones.

Aplicación: software que realiza una función particular para el usuario.

Archivo WSDL: *Web Services Description Language*. Archivo utilizado para describir Servicios *Web* y que permite a las aplicaciones el describirle a otras aplicaciones, las reglas para interactuar entre si.

Argumento: parte de una función que identifica los datos sobre los cuales se debe operar.

Autenticación: Capacidad de probar que una entidad, por ejemplo un usuario o una computadora es quien dice ser.

Biblioteca Digital: Extensión electrónica de funciones que los usuarios típicamente realizan y los recursos que ellos acceden en una biblioteca tradicional. (Sun Microsystems)

Base de Datos: es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior.

Browser: Visor o examinador. Programa cliente y herramienta básica de navegación para buscar los diferentes recursos de Internet. Los más usados son Netscape Navigator, Microsoft Internet Explorer, y Mosaic de la NCSA.

Clase: en programación orientada a objetos, un tipo de datos definido por el usuario que especifica un conjunto de objetos que comparten las mismas características. Un miembro de la clase (objeto) es un "ejemplo" o caso de la clase. Las clases concretas están diseñadas para citar como ejemplos, mientras que las clases abstractas, para pasar las características por herencia.

Cliente/Servidor: modelo lógico de una forma de proceso cooperativo, independiente de plataformas hardware y sistemas operativos. El concepto se refiere más a una filosofía que a un conjunto determinado de productos. Generalmente, el modelo se refiere a un puesto de trabajo o cliente que accede mediante una combinación de hardware y software a los recursos situados en una computadora denominado servidor.

Código de Acceso: combinación de letras, números y signos que debe introducirse para obtener acceso a un programa o partes de un programa determinado, una terminal ó computadora personal, un punto en la red, etc.

Código Fuente: programa en su forma original, tal y como fue escrito por el programador, el código fuente no es ejecutable directamente por el computador, debe convertirse en lenguaje de maquina mediante compiladores, ensambladores o interpretes.

Comando: instrucción dirigida a una computadora que invoca la ejecución de una secuencia de instrucciones programada previamente.

Componente: una unidad de composición de aplicaciones que posee un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes, de forma independiente en tiempo y espacio.

Conectividad: referente al grado en que los dispositivos de hardware, el software, y las bases de datos se pueden relacionar funcionalmente entre sí.

Controlador: software o pequeños programas necesarios para aprovechar al máximo los dispositivos, accesorios o periféricos distintos que se le pueden conectar a una computadora. Por otro lado también es utilizado para poder lograr conectividad con diversas aplicaciones, principalmente base de datos.

Conexión: circuito virtual de transporte que se establece entre dos programas de aplicación con fines comunicativos.

Contraseña: combinación de letras, números y signos que debe introducirse para obtener acceso a un programa o partes de un programa determinado, una terminal ó computadora personal, un punto en la red, etc.

Dato: Se le considera la unidad mínima y básica de toda información; es un elemento susceptible de observación directa ya que no dice algo por si mismo, es decir, los datos no son significativos como tales si no son procesados y convertidos para integrarse a toda la información.

DataSet: Conjunto de registros de una tabla.

Dirección IP: *Internet Protocol*. Protocolo de comunicaciones para Internet. Representación o notación de la dirección de un equipo conectado a Internet o a una red con el protocolo TCP/IP.

Dirección URL: *Uniform Resource Locator*. Abreviatura de localizador uniforme de recursos. Es la dirección que especifica la ubicación electrónica de un sitio (*site*) o un recurso (un archivo) específico de Internet. Una dirección URL consta normalmente de cuatro partes: protocolo, servidor (o dominio), ruta de acceso y nombre de archivo.

Firewall: Equipo de hardware o software utilizado en las redes de computadoras para prevenir algunos tipos de comunicaciones prohibidos por la política de red.

HTML: *HyperText Markup Language*. Lenguaje de programación de páginas *Web*. Se escribe utilizando identificadores. Estos son órdenes que se dan al navegador para que muestre una página *Web*.

HTTP: *Hypertext Transfer Protocol*. Protocolo de transferencia de HiperTexto. Es el protocolo en que se basa la tecnología de *World Wide Web*, consistente en un conjunto de reglas que gobiernan el software que transporta los documentos HTML a través de Internet.

Información: es la emisión o procesamiento de los datos, lo que puede proporcionar un conocimiento o la comprensión de ciertos factores.

Interface: en programación orientada a objetos, permiten acceder a métodos con el mismo nombre en diferentes clases.

Interfaz de usuario: Engloba la forma en la que el operador interactúa con la computadora, los mensajes que éste recibe en pantalla, las respuestas de la computadora a la utilización de periféricos de entrada de datos, etc.

Internet: es una red de redes a escala mundial de millones de computadoras interconectadas con el conjunto de protocolos TCP/IP. También se usa este nombre como sustantivo común y por tanto en minúsculas para designar a cualquier red de redes que

use las mismas tecnologías que la Internet, independientemente de su extensión o de que sea pública o privada.

Interoperabilidad: Característica de las computadoras que permite su interconexión y funcionamiento conjunto de manera compatible. Esto no siempre es posible, debido a los diferentes sistemas operativos y arquitecturas de cada sistema, pero los esfuerzos de estandarización están permitiendo que cada vez sean más las computadoras capaces de interoperar entre sí.

Intranet: red privada dentro de una organización. Las Intranet suelen utilizar protocolos de Internet para entregar contenido. A menudo se protegen contra el acceso desde Internet mediante servidores de seguridad.

Java: Lenguaje de programación desarrollada por Sun Microsystems, en conjunto con varios *frameworks* conforma una plataforma de software. Esta plataforma ha sido desarrollada de tal manera que los programas desarrollados para ella puedan ejecutarse de la misma forma en diferentes tipos de arquitecturas y dispositivos computacionales.

Juddi: Implementación en Java de código libre, de *Universal Description, Discovery, and Integration* (UDDI). Servidor de Servicios web.

Lenguaje de Programación: técnica estándar de comunicación para ordenarle instrucciones a la computadora. Un lenguaje le da la capacidad al programador de especificarle a la computadora, qué tipo de datos actúan y que acciones tomar bajo una variada gama de circunstancias, utilizando un lenguaje relativamente próximo al lenguaje humano.

Middleware: término común que se aplica al software que intercambia información en forma transparente entre aplicaciones y bases de datos. El Middleware es un mecanismo de conexión abstracta entre el software de aplicación y la base de datos, y oculta al programador de aplicaciones los elementos específicos que dependen de la implementación.

Multiplataforma: término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

ODBC: *Open Database Connectivity*. Estándar de acceso a bases de datos desarrollado por Microsoft Corporation.

OLE DB: *Object Linking and Embedding Data Base*. Parte central de la tecnología de conectividad de base de datos de Microsoft.

Página Web: documento el cual esta basado en el lenguaje HTML, la cual permite obtener información de cualquier tipo.

Paquete: uno o más programas diseñados para realizar una tarea de procesamiento particular.

PDF: *Portable Document Format*. Formato de Documento Portátil es una forma de almacenamiento de documentos, desarrollado por la empresa Adobe, que es independiente de la plataforma y puede ser enviado a través de un medio de comunicación como es Internet sin perder su configuración original.

Plataforma: se define como el funcionamiento, estructura y diseño de computadoras. Dentro de este término se engloban aspectos como formato de instrucción, modo de direccionamiento y conjunto de instrucciones.

Protocolo: sistema de reglas o estándares para comunicarse a través de una red, en especial a través de Internet. Los equipos y las redes interactúan de acuerdo con los protocolos que determinan el comportamiento que cada lado espera del otro en la transferencia de información.

Red de Área Local (LAN): Conexión entre computadoras mediante un medio de transmisión dentro de una distancia inferior a los 10 kilómetros. Son utilizadas principalmente en edificios donde se encuentran oficinas, universidades, hospitales y centros de investigación (entre otros).

Red de Área Amplia (WAN): interconexión de computadoras en áreas geográficas distantes; dicha interconexión necesita contar con mecanismos de acceso especiales, los cuales se enmarcan en lo que se conoce como intercomunicación remota, requiriendo para ello el empleo de microondas, fibra óptica, cable submarino y satélites.

Router (enrutador o encaminador): es un dispositivo hardware o software de interconexión de redes de computadoras que opera en la capa tres (nivel de red) del modelo OSI. Este dispositivo interconecta segmentos de red o redes enteras. Hace pasar paquetes de datos entre redes tomando como base la información de la capa de red.

SABD: Sistema Administrador de base de datos.

Servicios Web: componentes de software que permiten a los usuarios usar aplicaciones de negocio que comparten datos con otros programas modulares, vía Internet. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos *Web* estándar, como XML, SOAP, UDDI o WSDL.

Servidor Web: programa que implementa el protocolo HTTP. Se encarga de mantenerse a la espera de peticiones HTTP llevada a cabo por un cliente HTTP y de acuerdo a las mismas enviar un resultado.

Sistema Operativo: conjunto de programas que permiten al usuario, interactuar con un computadora. L

Sitio Web: colección de páginas *Web* relacionadas, que residen en el mismo servidor y están interconectadas entre sí mediante vínculos de hipertextos que facilitan el desplazarse ordenadamente a través de las páginas que lo forman.

Sitios Dinámicos: un sitio *Web* que se genera de forma dinámica en base a las solicitudes enviadas por el usuario.

SOAP: *Simple Object Access Protocol*. Estándar creado por *World Wide Web Consortium* que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

Software: parte lógica de la computadora, esto es, el conjunto de instrucciones (programas) que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Es por tanto un campo de estudio de la informática.

SQL: *Structured Query Language*. Lenguaje declarativo de acceso a Bases de Datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas.

Switches: llevan acabo la conectividad de una Red Local (LAN "Local Area Network").

UML: *Unified Modeling Language*. Lenguaje para el modelado, es un lenguaje para la especificación, visualización, construcción y documentación de los componentes de un Sistema.

Usuario: Individuo que interactúa con la computadora, o bien con la aplicación

WWW: *World Wide Web*. La *Web* o *WWW*, es un sistema de hipertexto que funciona sobre Internet. Para ver la información se utiliza una aplicación llamada navegador *Web* para extraer elementos de información (llamados "documentos" o "páginas *Web*") de los servidores *Web* (o "sitios") y presentar los resultados en la pantalla del usuario.

XML: *eXtensible Markup Language*. *Lenguaje Extensible de Marcas* desarrollado por el World Wide Web Consortium.

Capítulo I

Introducción

Actualmente las bibliotecas digitales, día a día toman relevancia dentro del área de las tecnologías de la información, dados los sorprendentes avances tecnológicos conforman actualmente una interesante combinación entre aplicaciones, sistemas y desde luego viejas teorías sobre el manejo de información ordenada y estructurada. Hoy en día representan una amplia oportunidad de adquisición de conocimiento, especialmente en áreas de ciencia y tecnología.

Las bibliotecas digitales trabajan bajo los conceptos generales de recopilación, organización y almacenamiento de información de manera digital, con el propósito de que los usuarios tengan opciones de búsqueda, recuperación y procesamiento de información, utilizando las actuales tecnologías como son: las redes de computadoras e Internet.

Este trabajo de investigación tiene como resultado la creación de una biblioteca digital, basada en la actual tecnología de Servicios Web [1] y representa una alternativa para el almacenamiento, procesamiento y recuperación de información de diferentes fuentes de información, como son por ejemplo los materiales científicos del área de computación y sistemas que han sido publicados en diferentes congresos nacionales e internacionales.

1.1 Antecedentes

El acelerado crecimiento en los volúmenes de información, la limitación de recursos humanos para su procesamiento y el surgimiento de nuevas tecnologías informáticas tales como Internet, las bases de datos y las aplicaciones para desarrollo Web, hicieron posible que investigadores, especialistas y técnicos del área bibliotecaria y de sistemas, se reunieran y decidieran apoyarse, para presentar en medios electrónicos esas grandes cantidades de volúmenes de información.

Como consecuencia de lo anterior surgen las bibliotecas digitales, que tiene como uno de los objetivos principales, impulsar el progreso tecnológico mediante la recopilación, procesamiento y recuperación de información, además de tener el propósito de hacer llegar cualquier acervo cultural a la mayoría de las personas.

La tendencia hacia el desarrollo de aplicaciones basadas en la Web, han hecho que la mayoría de las bibliotecas digitales busquen publicar su información a través de este medio, para así consultarla y poder trabajar con ella. A la fecha se han desarrollado tecnologías como XML (Extensible Markup Lenguaje, Lenguaje Extensible de Marcas) que permite intercambiar y publicar información de una manera segura, confiable y sencilla; además de reducir algunas tareas tediosas para el programador, como la validación de los

datos o el recorrido de las estructuras, las cuales están descritas en la especificación del estándar XML [2].

Otra tecnología existente son los Servicios Web, basados en XML, que permiten el intercambio, de datos y la llamada remota de lógica de aplicaciones. Estos pueden ser accedidos desde casi cualquier tipo de aplicación, incluidos otros Servicios *Web* XML, aplicaciones Web, de Windows y de Consola, donde el único requisito es que el cliente debe poder enviar, recibir y procesar mensajes del servicio Web XML [3].

1.2 Planteamiento del Problema y Justificación

Funcionalmente, una biblioteca digital es una colección organizada de documentos almacenados en formato digital que a su vez ofrece servicios de búsqueda y recuperación de información. Los documentos que se encuentran en una biblioteca digital pueden ser texto, imágenes, sonido, video o combinaciones de cualquiera de estas. Idealmente se debe almacenar y recuperar documentos completos y las búsquedas, se realizan sobre el contenido completo de los documentos, es decir, si el documento es texto, es posible realizar búsquedas sobre cada palabra incluida en el documento completo y una vez localizado es posible obtenerlo de manera inmediata. En el caso de otros formatos como audio y video, la búsqueda se realiza sobre los metadatos que definen las características de cada material.

La mayoría de los recursos digitalizados están almacenados en bases de datos de manera centralizada en caso de colecciones pequeñas y aunque lo común, es contar con información distribuida (más de dos bases de datos), en este caso, las búsquedas y las consultas de un sistema a otro deberán ser transparentes para el usuario.

La interfaz por excelencia de cualquier biblioteca digital hasta el momento es la Web, ya que ofrece una plataforma abierta, principalmente para proporcionar el acceso a los usuarios, aunque también es utilizada por los administradores de la biblioteca y por los grupos encargados de generar contenidos.

Las bibliotecas digitales proporcionan sus servicios a través de alguna red de cómputo, ya sea una red sólo de uso local o tan abierta como Internet, esto dependerá de los servicios y las restricciones que deban hacerse para el acceso a la información.

En el mercado existen tecnologías como ODBC [4], JDBC [5], OLEDB, y ADO [6], etc., que facilitan la interoperabilidad con diversas fuentes de información. A pesar de ello, se deben contemplar usuarios, códigos de acceso, entre otros, con los que normalmente el usuario no cuenta. Otra cosa que también debe siempre estar contemplada al utilizar este tipo de tecnologías, es que siempre habrá dentro la infraestructura a utilizar, un gestor de base de datos que permita organizar los contenidos, deberán de estar disponibles una gran diversidad de servidores Web, con hardware que pueda balancear la carga de peticiones, además de switches y routers.

Cuando se ingresa a la página principal de alguna biblioteca digital, se trata de una aplicación o serie de aplicaciones Web que pueden ser complejas, además de interactuar entre sí para darnos la impresión de estar tratando con alguna aplicación conformada de una sola pieza, robusta y fiable.

La presente biblioteca digital desarrollada, está basada en la utilización de Servicios Web, por lo que se logran algunas ventajas frente al desarrollo de portales de bibliotecas digitales tradicionales:

- Los Servicios Web son aplicaciones *sencillas* en donde se puede exponer parte del funcionamiento interno, son baratas de construir y de mantener.
- Las interfaces que se utilizan no son complejas para los usuarios.
- Permiten al desarrollador adaptar a la medida del cliente el funcionamiento de un portal completo.
- Al escribirse aplicaciones basadas en los Servicios Web desarrollados, es posible aportar valor a nuestro trabajo, desde luego con una cantidad reducida.

Sin embargo, para que los Servicios Web puedan ser publicados, descritos y/o descubiertos en Internet, es necesario contar con un servidor Web que permita realizar este tipo de operaciones mediante la comunicación en una Intranet y/o Internet, además de contar con las API (*Application Programming Interface*, Interfaz de Programación de Aplicación), necesarias para el manejo de mensajes SOAP (*Simple Object Access Protocol*, Protocolo Simple de Acceso a Objetos) [7], utilizados para comunicar al cliente con los Servicios Web y un archivo WSDL (*Web Service Description Language*, Lenguaje de Descripción de Servicio *Web*), que describe los servicios [8].

La presente tesis, concluye con el desarrollo de un conjunto de Servicios Web (SW), que proporcionan la funcionalidad básica de una biblioteca digital. Estos SW sirvieron como modelo para la implementación de la aplicación utilizando la tecnología de .NET.

1.3 Objetivos

El desarrollo de este trabajo de tesis tiene los siguientes objetivos:

1.3.1 General

Desarrollar, documentar y publicar un conjunto de servicios *Web*, que proporcionen la funcionalidad a una biblioteca digital (previa implementación), con el propósito de habilitar, la interoperabilidad de información a través de Internet y/o Intranet.

1.3.2 Específicos

- Diseñar e implementar, un ambiente de biblioteca digital que utilice los Servicios Web implementados, los cuáles proporcionarán la funcionalidad del sistema.
- Utilizar un servidor Web público (gratuito) de Servicios Web, para el registro, descubrimiento e integración de los Servicios Web realizados.

1.4 Limites y Alcances

En el presente trabajo de tesis, se describe el desarrollo de un conjunto de Servicios Web, que dan funcionalidad a una biblioteca digital, con los siguientes límites y alcances:

- En primera instancia los Servicios Web están orientados a apoyar la operación de la biblioteca digital institucional a pequeña y mediana escala, por lo que no se contempla su uso y acceso masivo por parte de usuarios, como podría serlo la biblioteca digital de la Association for Computer Machinery.
- Los criterios de búsquedas disponibles para el usuario están restringidas a los metadatos: título, autor, palabras claves y descripción. Asimismo, no se cuenta con un lenguaje para la definición de búsquedas personalizadas.
- El usuario puede ordenar el despliegue de los resultados encontrados en su búsqueda de acuerdo a : título, autor, fecha (los más recientes).
- Si bien, el diseño actual se apoya en SQL Server, es posible transportar los datos a un SDBD alternativo, realizando los ajustes necesarios en los parámetros de conexión utilizados.
- La seguridad en cuanto al consumo de los Servicios Web desarrollados esta implementada mediante la autenticación basada en formularios, lo cuál implica que los usuarios deben de proporcionar un login y password para el inicio de sesión. La misma situación aplica para desarrolladores que requieran acceder a los Servicios Web para integrarlos en sus aplicaciones.
- El consumo de los Servicios Web desarrollados para la biblioteca digital fue especificado en tiempo de diseño, sin embargo, se presenta un ejemplo de cómo podría realizarse el consumo dinámico empleando UDDI y clases Proxy generadas en tiempo de ejecución.
- El desarrollo y operación de la biblioteca digital se apoya en productos comerciales Microsoft, por lo que se requiere contar con las licencias respectivas para su distribución y operación.

1.5 Beneficios Esperados

El desarrollo de la biblioteca digital basada en Servicios Web permite obtener lo siguiente:

- Un conjunto de Servicios Web que pueden ser utilizados en desarrollos de bibliotecas digitales en diferentes plataformas con el propósito de otorgar diferentes funcionalidades.
- Capacidad de almacenamiento, organización y transmisión de información al usuario a través de la Web.
- Un ambiente de biblioteca digital amigable para el usuario, donde pueda realizar diferentes tipos de búsquedas de materiales, incluso seleccionarlos y guardarlos en un ambiente personalizado, además de poder visualizar la información de otros usuarios que almacenan los mismos materiales.
- En el aspecto computacional, se desea despertar el interés de estudiantes y profesores para utilizar como ejemplo los Servicios Web desarrollados e implementar nuevas funcionalidades de bibliotecas digitales.

1.6 Organización de la tesis

El contenido restante de la tesis esta estructurado de la siguiente manera:

- En el Capítulo 2 se presenta el estado del arte de las Bibliotecas Digitales y los Servicios Web. Se mencionan algunos conceptos generales de estas tecnologías incluyendo algunos estándares de organización de información para biblioteca digitales. Se describe la arquitectura general de biblioteca digital, además se describen algunos sistemas de bibliotecas digitales importantes. Se mencionan también la utilidad y los beneficios que representan la utilización de Servicios Web.
- En el Capítulo 3 se presentan las fases realizadas de análisis y diseño para la arquitectura de la aplicación desarrollada. Se describen los servicios comunes que una biblioteca digital debe proporcionar y los servicios que la biblioteca digital desarrollada proporciona. Se mencionan los objetivos de la funcionalidad de la aplicación y se establecen los requerimientos.
- En el Capítulo 4 se mencionan las herramientas y estándares empleados para el desarrollo de la aplicación y se describen los programas e interfaces que componen la biblioteca digital. Se presentan también las pruebas de funcionamiento de la biblioteca digital utilizando Servicios Web. Se explica el consumo dinámico de Servicios Web y se hace un comparativo de funcionalidades de bibliotecas digitales.

- En el Capítulo 5 se presentan un caso de estudio sobre el funcionamiento de la biblioteca digital.
- En el Capítulo 6 se describen las conclusiones obtenidas después del desarrollo e implementación de la biblioteca digital basada en Servicios Web y los resultados finales.

Capítulo II

Marco Teórico.

En el presente capítulo se presenta el estado del arte relacionado con la tesis. Se define el concepto de biblioteca digital y se describen sus características principales. Se describe la arquitectura general de una biblioteca digital. Se definen conceptos generales de servicios *Web* con el propósito de dar a conocer su significado y alcance. Se analizan los trabajos previos relacionados con biblioteca digitales y tecnologías de desarrollo.

2.1 Bibliotecas Digitales

Durante siglos el recuerdo de la biblioteca de Alejandría ha marcado el sueño de los hombres, que quieren contar con un almacén físico para albergar copias de todo el conocimiento de la humanidad.

Con la invención de la imprenta de Gutenberg se duplico la creación de libros y desde luego las posibilidades de la lectura individual. Como nunca antes en la historia, actualmente proliferan las bibliotecas y ahora con la explosión de Internet, la digitalización de los libros abre un nuevo espacio a las bibliotecas virtuales, accesibles desde cualquier punto del planeta y en cualquier momento.

La iniciativa más importante en Internet es el proyecto Gutenberg iniciado en 1971 cuando Michael Hart consiguió acceso al sistema de computadoras de la Universidad de Illinois, buscando darle un uso adecuado a esos recursos, pensó en la posibilidad de colocar textos en formato electrónico que pudieran ser copiados por los usuarios.

En la actualidad, el proyecto Gutenberg es uno de los sitios más premiados por su aporte a la cultura, tiene servidores de acceso en los cinco continentes y en septiembre de 2004 tenía casi 14.000 libros en su colección, actualmente cuenta con 18.000

La organización y el almacenamiento de información son dos de los elementos que caracterizan a una biblioteca, y hacen dos espacios importantes a ser considerados para ingresar al medio digital y la razón fundamental de la construcción de bibliotecas digitales es la creencia que se tiene que pueden proporcionar mejor la información a diferencia de cómo se hacía en el pasado. Las bibliotecas tradicionales son parte fundamental de toda sociedad pero desafortunadamente no son perfectas.

Ante el gran crecimiento de información en la actualidad, que ha provocado la red informática, el papel de las bibliotecas en la organización, conservación y puesta a disposición de los materiales digitalizados sigue ocupando sin duda alguna un lugar fundamental y las biblioteca digitales BD dan un paso a una nueva forma de ubicación, administración, cuidado, consulta y adquisición de las publicaciones disponibles en

formatos de texto, audio y video. Trabajar con este nuevo modelo implica el esfuerzo de diversos grupos de personas.

Características de las bibliotecas digitales

Las bibliotecas digitales son una colección de recursos electrónicos que proveen directa o indirectamente acceso a una colección de documentos organizados sistemáticamente [2]. Entre los servicios *mínimos* con los que debe contar una biblioteca digital se encuentran:

- la *preservación*,
- la *búsqueda*,
- y el *acceso universal a la información*.

El ideal es que la biblioteca digital ofrezca servicios, organización, clasificación, catalogación, acceso personalizado, recuperación de información y desde luego la presentación del material en forma electrónica. Estos materiales o recursos electrónicos pueden estar en diferentes formatos multimedios como por ejemplo audio, texto, e imágenes, etc. El poder generar una búsqueda dentro de una biblioteca digital presenta grandes ventajas para la localización de algún material en específico, ya que arroja resultados de una manera rápida y sencilla.

Actualmente gracias al avance de la tecnología, un usuario puede desde una computadora personal consultar recursos electrónicos que están almacenados en diferentes computadoras alrededor del mundo.

Existen *estándares* de organización de información y comunicación en las bibliotecas digitales tales como Z39.50 [9], Marc[10], Dublin Core [10], etc. Y que son protocolos dirigidos a facilitar la búsqueda y recuperación de información. Algunos de estos comenzaron como una simple descripción de los recursos almacenados en sus colecciones de datos pero ahora han captado la atención de las comunidades científicas para la descripción formal de muchas entidades, tales como organizaciones y empresas. A continuación se describen estos estándares:

Z39.50

La ANSI/NISO Z39.50 es una Norma Nacional Americana que se aplica a la Recuperación de Información, definición del servicio y especificación del protocolo para las interconexiones de sistemas bibliotecarios abiertos. La NISO, es un órgano de la ANSI (American National Standards Institute), que emite normas de desarrollo de bibliotecas, información, y editores. Aprobó la norma original en 1988 (referida como Z39.50-1988 o versión 1).

La ANSI/NISO Z39.50 define un medio normalizado para comunicar dos computadoras con el propósito de transferir información. La Z39.50 facilita el uso de las bases de datos con grandes volúmenes de información a través de la normalización de procedimientos y características de la búsqueda y transferencia de información. Específicamente, la Z39.50 permite la recuperación de información en ambientes distribuidos, cliente - servidor, en los

cuales una computadora opera como cliente que envía un requerimiento de búsqueda (es decir, una consulta) a otra computadora que actúa como servidor de información.

El software en el servidor ejecuta la búsqueda en una o más bases de datos y crea como resultado un conjunto de registros que concuerdan con el criterio del requerimiento de búsqueda. El servidor devuelve el conjunto de registros al cliente para que sean procesados. La fortaleza de la Z39.50 es que separa la interfaz del usuario del lado del cliente, de la información de los servidores, de los motores de búsquedas y de las bases de datos. La Z39.50 proporciona una visión consistente de la información de una amplia variedad de fuentes, y esto ofrece a los desarrolladores de sistemas la capacidad de integrar la información desde un conjunto de bases de datos y servidores.

La arquitectura distribuida cliente/servidor proporciona las bases para la comunicación Z39.50 y las especificaciones de recuperación de información. El software para la interacción del usuario final realiza las búsquedas, y devuelve los resultados, está al margen del software que administra la información. La Z39.50 no especifica detalladamente la interfaz del usuario, pero si contiene especificaciones y procedimientos de protocolo que determinan al cliente Z39.50, tal como es el inicio de la consulta para recuperar información y cómo este requiere las operaciones específicas al servidor. Contiene también especificaciones y procedimientos del protocolo que se refieren al servidor Z39.50, tal como la manipulación del conjunto de resultados y los formatos en los cuales se entregan los registros al cliente.

MARC/USMarc

El formato más popular para la descripción bibliográfica de documentos ha sido el formato MARC ("Machine Readable Catalogue Format, Formato de Catálogo Computacionalmente Legible"), definido por la ISO 2709 "Format for bibliographic Information Interchange on Magnetic Tape, Formato para el Intercambio de Información Bibliográfica por medio de Cinta Magnética". MARC, o alguna de sus variantes nacionales, se utilizan en infinidad de bibliotecas y aplicaciones bibliográficas en todo el mundo. Sin embargo, MARC fue creado para la descripción de documentos impresos, y sólo en los últimos años, con la popularización de Internet, se inició su actualización por la librería del Congreso de EEUU. Concretamente el formato USMARC ya admite la inclusión de documentos electrónicos accesibles en Internet. En EEUU existen proyectos que utilizan estas modificaciones como el catálogo INTERCAT de OCLC. También son bastantes las bibliotecas que comienzan la introducción de recursos electrónicos en sus catálogos en línea. Entre otras, la Biblioteca de la Universitat Oberta de Catalunya, lo utiliza para describir y dar acceso a recursos electrónicos que han seleccionado.

El principal problema del MARC es que se produce un considerable retraso entre las actualizaciones que se hacen al formato y el momento en que se incluyen en el software de gestión de los catálogos en línea. Esto, unido a la diversidad de sistemas de gestión actualmente implementados en las bibliotecas, convierte al MARC en inapropiado para la catalogación inmediata de recursos electrónicos. Para este propósito, hasta el momento ha demostrado ser más adecuados el formato Dublin Core.

Dublin Core

Una biblioteca digital tiene que cumplir una serie de características que le den el valor que necesita para difundir los contenidos facilitando las búsquedas y así optimizar la recuperación de la información de manera eficaz. La información tiene que ser recuperable mediante metadatos (información sobre los datos), estos a su vez tienen gran importancia en la composición de las bibliotecas digitales, ya que permiten la catalogación que facilita la búsqueda efectiva y precisa. En este campo, hay que destacar los trabajos normativos de Dublin Core.

El presente trabajo desarrollado está soportado por una base de datos cuyo modelado cumple con las normas del Dublin Core, este estándar fue elegido ya que proporciona un simple pero eficaz conjunto de elementos que permiten describir una amplia gama de recursos, entre los que se encuentran los artículos de investigación del área computacional que son en primera instancia los materiales que la biblioteca digital implementada debe publicar.

Dublin Core se encuentra entre los estándares de catalogación o de metadatos más comúnmente utilizados en las bibliotecas digitales y está conformado por 15 elementos los cuales tienen nombres descriptivos que pretenden transmitir un significado a los mismos, cada elemento es opcional y además, los elementos pueden aparecer en cualquier orden.

Los elementos del Dublin Core se pueden clasificar en tres grupos que indican la clase o el ámbito de la información que se guarda en ellos:

1. Elementos relacionados principalmente con el contenido del recurso.
2. Elementos relacionados principalmente con el recurso cuando es visto como una propiedad intelectual.
3. Elementos relacionados principalmente con la instanciación del recurso.

Contenido	Propiedad Intelectual	Instanciación
Title	Creador	Date
Subject	Publisher	Type
Description	Contributor	Format
Source	Rights	Identifier
Language		
Relation		
Coverage		

Tabla 2.1 Elementos del Dublin Core

En México, así como en varios países se destinan recursos económicos para promover la investigación, desarrollo y aplicación de biblioteca digitales, de tal manera que los avances logrados en el área han facilitado el desarrollo de actividades de estudio e investigación colaborativa. Uno de los grandes beneficios es la disponibilidad del conocimiento reciente para comunidades académicas y de investigación.

El tema de las bibliotecas digitales es actualmente uno de los puntos de giro en cuanto al cambio de paradigmas en la ciencia de la información y la bibliotecología, en tanto supone el acceso universal a la información, que engloba varios aspectos:

- La biblioteca digital no existe como una institución aislada, sino en cooperación con bibliotecas o instituciones afines, para que el intercambio de información se convierta en algo fluido.
- La tecnología, como elemento indispensable en el que se sustenta la evolución de la biblioteca tradicional.
- La mentalidad de los profesionales y usuarios de las bibliotecas debe progresar aún más y orientarse hacia las nuevas metodologías y procedimientos de la sociedad de la información e incorporarse a la nueva faceta digital de las bibliotecas.

2.1.1 Conceptos Generales

Las palabras "biblioteca digital" tienen significados diferentes para sectores distintos y para algunos es la automatización de bibliotecas tradicionales; para otros una colección de manuscritos digitalizados y para algunos profesionales del área computacional, es un sistema de información distribuido basado en texto; una colección de servicios de información distribuidos, un espacio distribuido de enlaces de información o un sistema de información de múltiples medios. Sun Microsystems define una biblioteca digital como "la extensión electrónica de funciones que los usuarios típicamente realizan y los recursos que ellos acceden en una biblioteca tradicional" [11]. Las definiciones varían según el punto de vista del usuario, pero la característica común entre todas ellas es que una biblioteca digital debe contener servicios esenciales de las bibliotecas tradicionales y además, debe explorar las ventajas de almacenamiento, búsqueda y comunicación digital. En su concepto más simple, una biblioteca digital es un espacio en donde la información es almacenada y procesada en formato digital.

La definición tomada por la *Digital Libraries Federation* [12], es una de las más completas: "Las bibliotecas digitales son organizaciones que proveen los recursos, incluyendo personal especializado, para seleccionar, estructurar, distribuir, controlar el acceso, conservar la integridad y asegurar la persistencia a través del tiempo de colecciones de trabajos digitales que estén fácil y económicamente disponibles para usarse por una comunidad definida o para un conjunto de comunidades".

Actualmente, Internet es un repositorio de información sin estructura y altamente dinámico. Internet no es una biblioteca y una biblioteca no es simplemente un cuarto lleno

de libros, si no un cuarto lleno de libros junto con servicios. El mínimo servicio esperado de una biblioteca es el indexado, de tal manera que pueda encontrarse la información deseada de manera relativamente fácil. Las mejores bibliotecas proporcionan mejores servicios; la *Web*, por sí sola, no ofrece tales servicios y la meta de la investigación en BD es desarrollar técnicas y herramientas de software para proporcionar estos servicios.

Las bibliotecas digitales no son simples versiones digitales de la biblioteca tradicional o el reservorio de libros, fotos, películas, videos, etc; son mucho más que una alternativa, son la respuesta que se necesita en la sociedad de la información. Es la estrategia de supervivencia que debe articularse, a partir de una adecuada interpretación del contexto en el cual se desarrolle según Christine Borgman [13].

Las bibliotecas digitales son sistemas que proveen a una comunidad de usuarios el acceso coherente a un gran depósito organizado de información y conocimientos, con recursos electrónicos y capacidades técnicas asociadas para crear, buscar y usar la información. El contenido incluye los datos, meta datos que describen varios aspectos de los datos -por ejemplo: representación formal, autor, dueño de los derechos de publicación, etc.- y meta datos que consisten en enlaces o relaciones con otros datos u otros meta datos, que pueden ser internos o externos a la biblioteca digital.

En cuanto a lo anterior pueden listarse los rasgos comunes de las diferentes definiciones de biblioteca digital:

- No son entidades individuales.
- Requieren tecnología para enlazar recursos de muchas bibliotecas digitales y servicios de información.
- Los enlaces entre las múltiples bibliotecas digitales y servicios de información deben ser transparentes para los usuarios finales.

Su meta es el acceso universal a los recursos de información y a las bibliotecas digitales de manera global.

2.1.2 Arquitectura General de una Biblioteca Digital.

Desde el punto de vista tecnológico y como lo muestra la Figura 2.1, un ambiente de BD desarrollado completamente involucra los siguientes elementos [14]:

- 1.- Conversión inicial del contenido de su forma física a su forma digital.
- 2.- La extracción o creación de meta datos o indexado de la información describiendo el contenido para facilitar la búsqueda y descubrimiento, así como de meta datos estructurales y de administración para asistir en la preservación, administración y visualización.
- 3.- Almacenamiento del contenido digital y los meta datos en un repositorio multimedia apropiado. El repositorio puede incluir capacidades de administración de derechos para hacer cumplir, si es requerido, los derechos de propiedad intelectual. La funcionalidad de

comercio electrónico podría estar disponible si se requiere el manejo de cuentas de acceso y servicios de facturación.

4.- Servicios para el cliente, principalmente consultas.

5.- Entrega de contenido vía transferencia de archivos o flujo de medios (*streaming media*).

6.- Patrón de acceso a través de una aplicación cliente o *Browser*.

7.- Una red pública (Internet) o privada.

Los componentes de una BD deben ser adaptados para capturar, codificar y entregar información de acuerdo a las prácticas estándares adoptadas por la industria bibliotecaria. Debido al rápido paso de cambio tecnológico, algunos estándares son concretos y otros están emergiendo.

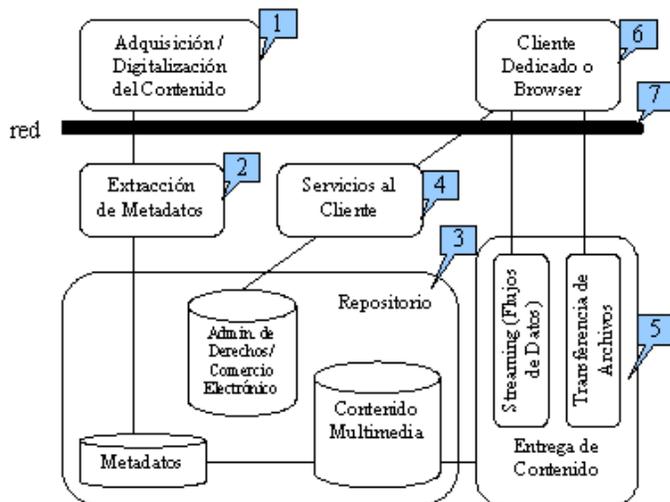


Figura 2.1 Componentes funcionales de una biblioteca digital.

2.1.3 Sistemas de Bibliotecas Digitales.

Actualmente en México como en muchos países, existen un gran número de proyectos que involucran el desarrollo de bibliotecas digitales, ya que se destinan recursos económicos, para promover la investigación, desarrollo y puesta en marcha de este tipo de aplicaciones.

Desde el surgimiento de los primeros proyectos de bibliotecas digitales en Internet, aparecieron estudios e investigaciones que pretenden evaluar su funcionamiento, eficacia y validez en competencia con las bibliotecas tradicionales.

Las bibliotecas digitales en Internet pueden tener, según su cobertura temática, una amplia cobertura de diversas ramas del conocimiento o determinado grado de especialización en ciencias, arte, literatura, filosofía, religión, entre otras.

Algunos de los desarrollos importantes de bibliotecas digitales que marcan importantes modelos por su desempeño en la red se encuentran las siguientes:

- *Berkerley Digital Library Sun Site* (<http://sunsite.berkeley.edu>)

Es una iniciativa de la universidad de Berkeley y de la empresa informática Sun, constituye una de las mejores bibliotecas digitales del mundo con varias colecciones de interés.

- Proyecto Esmeralda (URL)

Fue fundado en 1967 por un grupo de jóvenes académicos insatisfechos con los medios internacionales de difusión del conocimiento científico. Ellos crearon su propia casa editorial: alternativa, orientada a manipular la información y las estrategias de difusión internacionales del conocimiento. En el 2001, el proyecto adoptó el nombre de Esmeralda. Su objetivo es mejorar el nivel de los servicios a los usuarios para mejorar continuamente el acceso a sus contenidos por parte de sus usuarios con una distribución a todo el mundo. Es una biblioteca digital líder en publicaciones sobre gestión de información y bibliotecas.

- Iniciativa Digital de Bibliotecas. Fase 2 (<http://www.dli2.nsf.gov/>)

Por el carácter multidisciplinario e integrador de importantes proyectos de bibliotecas digitales de las universidades de Estados Unidos, la participación de reconocidos especialistas en la materia de todo el mundo y la aparición de los más actuales estudios sobre el tema, se debe destacar el proyecto "Iniciativa Digital de Bibliotecas. Fase 2", desarrollado por las divisiones de Información y Sistemas Inteligentes (*Information and Intelligent Systems* (IIS) y la División de Computación y Ciencias de la Información de la Fundación Nacional de la Ciencia de Estados Unidos (*Division of the Directorate for Computer and Information Science Engineering*, CISE por sus siglas en inglés. *National Science Foundation*, NSF por su siglas en inglés).

- Biblioteca Digital de Filosofía (www.filosofia.org)

Este proyecto de acceso gratis es uno de los más completos que hallados en la red, y por su interrelación con las artes y la enseñanza artística, así como por su amplitud, se consideró sumamente importante y de mención imprescindible.

- Biblioteca Jurídica (www.bibliojuridica.org)

Esta biblioteca es un buen ejemplo de biblioteca especializada en español. Su magnifico esquema de navegación, así como la sencillez y organización excelente de la información que contiene la presentan como de obligada consulta al hablar de este tipo de bibliotecas en la red. Atesora fundamentalmente información jurídica de México, pero abarca también

los clásicos del derecho internacional. En formato PDF brinda más de 10 000 títulos en la materia.

Entre otros desarrollos importantes también se encuentran los proyectos Phornesis, desarrollados por el Campus Monterrey del Tec y el proyecto Marian de Virginia *Polytechnic Institute* de Estados Unidos [15].

El sistema Phornesis es actualmente para el Tecnológico de Monterrey una herramienta de software libre para la creación de biblioteca digitales distribuidas por Internet. Esta herramienta permite el envío, búsqueda, recuperación y administración de una biblioteca digital vía WWW [16], y ha sido construida integrando componentes de software libre, estándares abiertos y el sistema MG [17]. Las funcionalidades de Phornesis son búsqueda en el texto completo de documentos en español e inglés; interfaz bilingüe; almacenamiento eficiente, así como recuperación de imágenes, audio, video texto y cualquier otro tipo de documento digital. Este sistema es eficiente en almacenamiento, indexación, recuperación de documentos y la construcción de colecciones.

Por otra parte, Marian es un proyecto de Virginia Tech [18], tiene como resultado un innovador sistema de búsqueda en bibliotecas digitales. Sirve hasta el momento como catálogo electrónico de la biblioteca digital de Virginia Tech, que da soporte de búsqueda de texto libre. Este proyecto es un sistema multiusuario, que ofrece servicios de edición y refinamiento de búsquedas, de documentos y progresivamente mejora su portabilidad [19].

Los sistemas Phornesis y Marian no contaban con una interoperabilidad que les permitiera el intercambio de información de sus repositorios. Cada uno conservaba su propio acervo de información, por lo que enfocó a lograr la interoperabilidad entre sus colecciones digitales.

En la actualidad la interoperabilidad es uno de los retos de las bibliotecas, las colecciones y los servicios son provistos por diferentes organizaciones con diferentes sistemas de cómputo. Además la información proviene de diferentes fuentes es procesada de diferentes maneras y manejada por diferentes procesos de calidad. Por lo tanto, el concepto de interoperabilidad se puede definir como la capacidad de las bibliotecas digitales para compartir documentos, búsquedas y servicios. La interoperabilidad también aplica a la habilidad de dos o más sistemas para aceptar servicios de otros sistemas para operar efectivamente en conjunto [20].

Actualmente, la plataforma de Servicios Web ha sido desarrollada lo suficiente para crear aplicaciones que se comunican enviando mensajes SOAP sobre HTTP. Esto no significa que no existan herramientas o marcos de trabajo de servicios *Web* que puedan hacer más cosas (por ejemplo, enviar mensajes SOAP sobre SMTP). Sin embargo, la mensajería basada en HTTP es la única opción de comunicación más generalmente soportada. La mayoría de las herramientas de Servicios *Web* mapean mensajes SOAP a invocaciones de métodos en objetos [21]. Algunas proporcionan además un acceso directo al cuerpo de los mensajes SOAP, exponiéndolos como XML.

La biblioteca digital desarrollada, esta formada por tecnologías que ayudan a resolver el problema de interoperabilidad, por ejemplo cuenta con el XML (*Extensible Markup*

Language, Lenguaje Extensible de Marcas) [22] , que sin duda alguna hoy por hoy ha sido aceptado como un estándar para el intercambio de información, además de que se han desarrollado muchas tecnologías entorno a este estándar, una de ellas también utiliza en la implementación de esta tesis, los Servicios Web, que también han tenido una aceptación importante para la interacción entre aplicaciones.

En la actualidad la programación basada en componentes se ha convertido en más popular que nunca, es difícil no encontrarnos con alguna aplicación que de alguna forma no involucre el aprovechar componentes.

A mediados de la década del los 90 y con la aparición del Internet y el auge que este tuvo, ha existido siempre la necesidad e inquietud por parte de las empresas desarrolladoras de software de buscar o contar con la manera de lograr la integración entre sistemas heterogéneos, obviamente la referencia es tanto al software como al hardware. Para lograr tal integración muchas compañías fueron creando en forma individual la manera de lograr esta integración. Muchas empresas comenzaron una aventurada carrera para generar la mejor tecnología integradora de sistemas, pero a medida que la competencia se hacia cada vez más fuerte, la integración se hacía cada vez más difícil.

Las empresas se dieron cuenta que era imposible crear una plataforma integrada de forma individual, así que decidieron atacar el problema de raíz, por cuál en vez de crear la mejor plataforma integradora, buscaron un lenguaje común de intercambio de información, desde luego aprovechando los estándares existentes en el mercado y es así, como surgen los Servicios Web basados en XML.

2.2 Servicios Web

2.2.1 Conceptos Generales

Existen varias definiciones sobre lo que son los Servicios Web, lo que muestra su complejidad a la hora de dar una adecuada definición que encierre todo lo que son y desde luego lo que implican. Una definición posible sería hablar de ellos como un conjunto de aplicaciones o tecnologías que tienen la capacidad para interoperar en la *Web* [23]. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la *Web*.

En [24], se define a un Servicio Web como una aplicación de red habilitada para Internet mediante protocolos que sirven como estándares de comunicación entre aplicaciones, por medio de interfaces bien definidas y que son descritas, utilizando un lenguaje estándar. Otra definición tomada de [25], nos dice que un Servicio Web se define como una aplicación modular, generalmente independiente y que se puede describir a sí misma, y esta localizada e invocada a través de Internet o una Intranet. Otra definición que se da en [24], nos dice que los Servicios Web son componentes de software que permiten a los usuarios usar aplicaciones de negocios que comparten datos con otros programas

modulares, vía Internet. Son aplicaciones independientes de la plataforma que pueden ser fácilmente publicadas, localizadas e invocadas mediante protocolos *Web* estándar, como XML, SOAP, UDDI o WSDL. El objetivo principal de este tipo de estándares es crear un directorio en línea de Servicios Web, desde luego que pueda ser localizado por un gran número de personas o empresas y que a su vez tenga una alta fiabilidad.

La funcionalidad de los protocolos antes mencionados es la siguiente:

- **XML (*eXtensible Markup Language*)**: Un Servicio Web es una aplicación *Web* basada en XML
- **WSDL (*Web Services Definition Service*)**: Este protocolo se encarga de describir el Servicio *Web* cuando es publicado. Es el lenguaje XML que los proveedores emplean para describir sus Servicios *Web*.
- **SOAP (*Simple Object Access Protocol*)**: Permite que programas que corren en diferentes sistemas operativos se comuniquen. La comunicación entre las diferentes entidades se realiza mediante mensajes que son rutados en un sobre o mensaje SOAP.

UDDI (*Universal Description Discovery and Integration*): Este protocolo permite la publicación y localización de los servicios. Los directorios UDDI actúan como guía telefónica de Servicios *Webs*.

2.2.2 Utilidad y beneficios de los servicios Web

Los Servicios Web proporcionan mecanismos de comunicación estándar entre diferentes aplicaciones, que interactúan entre sí, con el propósito de presentar información dinámica al usuario.

Para que puedan proporcionar interoperabilidad y extensibilidad entre aplicaciones, y para que puedan realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.

Un Servicio Web es equivalente a una rutina, la cuál es como una caja negra que encierra cierto proceso o algoritmo, y que cumple una función clara. Muchas rutinas y un guión central componen un *programa* en lo que se llama "programación estructurada". Un servicio *Web* viene hacer una rutina en Internet. Los protocolos que soportan los servicios *Web* se comunican normalmente por el puerto 80, basándose en HTTP, métodos GET y PUT, esto permite tener acceso a ellos, de igual forma que accesa a una página *Web*. La diferencia que existe, es que la página *Web*, es visitada por cualquier usuario interesado, mientras que el Servicio Web sólo es visitado por los programas o aplicaciones que lo requieren.

Permiten que las aplicaciones compartan información, incluso pueden invocar funciones o métodos de otras aplicaciones independientemente de cómo se hayan creado, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados

para obtener acceso a ellas. Aunque los servicios *Web* XML son independientes entre sí, pueden vincularse y formar un grupo de colaboración para realizar una tarea determinada.

Algunos beneficios proporcionados por esta tecnología son los siguientes [26]:

- **Accesibles:** Pueden ser accedidos por diferentes dispositivos de comunicación.
- **Autodescriptibles:** Los Servicios Web pueden describir, publicar e invocar aplicaciones enteras.
- **Modulares:** En un momento determinado pueden solo dejar cierta parte de la descripción de sus operaciones publicadas, ocultando así la implementación.
- **Universales:** Se basan en XML.
- **Independientes:** Pueden ser desarrollados en un lenguaje de programación y ser utilizados por otro, incluso no dependen de la plataforma en que se desarrollen.
- **Interoperabilidad:** Al ser interoperables, permiten que la asociación entre empresas se puedan realizar de manera dinámica y automática, además de facilitar el intercambio de información entre empresas de diferentes áreas.
- **HTTP:** Al apoyarse en HTTP, los servicios *Web* pueden aprovecharse de los sistemas de seguridad *firewall* sin necesidad de cambiar las reglas de filtrado.

Existen en el mercado un gran número de aplicaciones que permiten construir y publicar servicios *Web* (.NET, Java(JUDDI), Axis, MS SOAP Toolkit, Apache SOAP, Oracle, entre otros)

La aplicación desarrollada en esta tesis, parte del desarrollo de un conjunto de Servicios Web, que fueron implementados con la tecnología .NET utilizando el lenguaje de programación de C#, que dan la lógica y funcionamiento a la biblioteca digital. Los Servicios Web implementados están registrados en un servidor UDDI para Servicios *Web* (JUDDI) [27], y pueden ser utilizados por cualquier usuario ó empresa.

2.2.3 Arquitectura de los Servicios Web

La arquitectura de Servicios Web plantea algo más que una técnica para el desarrollo de aplicaciones *Web*, representa un completo modelo de computación. Para hacer realidad esta tecnología, se requiere de tres entidades participantes:

- "*Service providers*" quienes brindan el servicio en la *Web*.
- "*Service brokers*" quienes registran y categorizar los Servicios *Web* ofrecidos por los "Service Providers"
- "*Service requesters*" quienes utilizan a los "*service brokers*" para encontrar el servicio que necesitan.

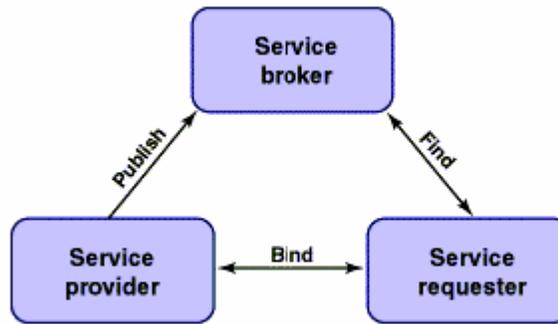


Figura 2.2 Arquitectura básica de los servicios Web.

Existen tres operaciones básicas que tienen lugar entre el "service provider", el "service broker" y el "service requester":

- a) "*Publishing & unpublishing*", involucra la publicación utilizando WSDL – de servicios en registros públicos (*publishing*) y el borrado de dichas entradas (*unpublishing*). El "*service provider*" contacta al "*service broker*" para publicar y remover un servicio.
- b) "*Find*", o búsqueda realizada por el "*service requester*" y el "*service broker*" en forma conjunta. El "*service requester*" describe el tipo de servicio que necesita y el "*service broker*" responde con los resultados que mejor satisfacen el requerimiento.
- c) "*Bind*", utilizando SOAP como protocolo el "*service provider*" permite que el "*service requester*" invoque sus servicios.

Los servicios provistos por un Servicio *Web* determinado son descritos utilizando WSDL. Utilizando WSDL, los "*service providers*" pueden describirse a sí mismos, los "*service requesters*" pueden describir el tipo de servicios que buscan y los "*service brokers*" cuentan con una forma automatizada de determinar que par "*requester provider*" es el más viable. Una vez establecido el vínculo (*bind*), el servicio es invocado a través de SOAP, un protocolo sencillo, de invocación remota basada en XML.

Las tres operaciones descritas, y de allí el éxito de esta tecnología y la amplia difusión y aceptación que ha logrado, se realiza a través de un estándar universal basado en XML, llamado UDDI: *Universal Description, Discovery and Integration* [29]. Este mecanismo permite a empresas distintas, que utilizan distintos lenguajes de programación y distintas plataformas de ejecución vincularse a través de Servicios *Web*, publicándolos (*publishing & unpublishing – description*), encontrándolos (*find – discovery*) e invocándolos (*bind – integration*) sobre la *Web* permitiendo la integración ínter empresarial.

2.2.4 Simple Object Access Protocol (SOAP).

Deriva de un protocolo creado por David Winer, XML-RPC en 1998 [28]. Es un protocolo de intercambio de información estructurada basado en XML. Entre otras cosas, SOAP

puede representar llamadas a procedimientos remotos (RPC), sus respuestas proporcionan un mecanismo para el intercambio de información y servicios entre aplicaciones [24].

La estructura de un mensaje SOAP se presenta en la Figura 2.3.

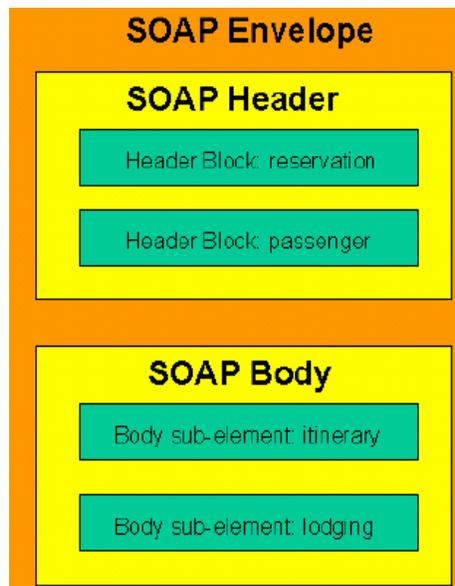


Figura 2.3 Estructura de Mensaje SOAP.

Un mensaje SOAP se compone de un sobre que contiene el cuerpo del mensaje y cualquier información de cabecera que se utiliza para describir el mensaje. A continuación se muestra un ejemplo de mensaje SOAP.

```
Mensaje SOAP
<?xml version="1.0"?>
<soap:Envelope xmlns:soap=http://schemas.xmlsoap.org/soap/envelope/>
  <soap:Header>
    <!--Aquí va información opcional de cabecera.-->
    <To> Scout </To>
    <From>Suzanne</From>
  </soap:Header>
  <soap:Body>
    <!--Aquí va el mensaje.-->
    Por favor , compra leche cuando vuelvas a casa del trabajo.
  </soap:Body>
```

El elemento raíz del documento es el elemento *Envelope*. El ejemplo contiene dos subelementos, *Body* y *Header*. Un mensaje SOAP válido también puede contener otros elementos hijo en el sobre. El sobre puede contener un elemento *Header* opcional, que contiene información sobre el mensaje. En el ejemplo anterior la cabecera contiene dos elementos que describen a quien compuso el mensaje y el posible receptor del mismo. El sobre debe contener un elemento *Body*, el cuál contiene la carga de datos del mensaje. En

el ejemplo, el cuerpo contiene una simple cadena de caracteres. Es necesario tener en cuenta que todos los elementos propios del SOAP tienen el prefijo del espacio de nombres *soap*. Este prefijo se define en el elemento *Envelope* y apunta al esquema de SOAP que describe la estructura de un mensaje SOAP. El prefijo se añade a todos los elementos definidos en el espacio de nombres SOAP. Estos elementos están totalmente cualificados. El prefijo *soap* indica que el elemento *Envelope* es una instancia del tipo *Envelope de SOAP*.

Algunas de las ventajas de SOAP son:

- No depende, ni está asociado a algún lenguaje de programación
- No se asocia especialmente a ningún protocolo de transporte
- No depende de alguna infraestructura de objetos distribuidos en especial
- Permite la interoperabilidad entre múltiples entornos.

2.2.5 Web Service Description Language (WSDL)

Es un formato XML que se utiliza para describir Servicios *Web*. Describe los servicios de red como un conjunto de puntos finales que procesan mensajes contenedores de información orientada tanto a documentos como a procedimientos. Las operaciones y los mensajes se definen de manera abstracta y después se enlazan a un protocolo de red y a un formato de mensaje concreto para definir un punto final de red. Los puntos finales concretos relacionados se combinan en puntos finales abstractos (servicios).

WSDL es extensible, por lo que se permite la descripción de puntos finales de red y sus mensajes, independientemente de los formatos de los mensajes o protocolos de red utilizados para comunicarse.

Dentro de un documento WSDL, los servicios se definen utilizando cinco elementos principales:

- **Types:** son las definiciones de los tipos de datos, que se utilizan para describir el intercambio de mensajes.
- **Message:** representa una definición abstracta de los datos que se están transmitiendo. Se compone de partes lógicas, y cada una se asocia a una definición dentro de algún sistema de tipos.
- **PortType:** conjunto de operaciones abstractas. Cada operación hace referencia a un mensaje entrante y saliente.
- **Binding:** determina las especificaciones del protocolo concreto y del formato de datos para las operaciones y mensajes definidos por un portType.
- **Port:** especifica la dirección para un enlace.
- **Service:** define una colección de puertos relacionados.

2.2.6 Universal Description, Discovery and Integration (UDDI).

Como parte de la tecnología de Servicios *Web*, UDDI [29] es un registro público diseñado para almacenar de forma ordenada, la información sobre empresas y los Servicios *Web* que éstas ofrecen. Se pueden utilizar sistemas taxonómicos estándar para clasificar estos datos y poder encontrarlos posteriormente en función de una categorización. Algo importante que tiene UDDI, es que contiene información sobre la interfaces técnicas de los servicios de una empresa. A través de un conjunto de llamadas a API XML basadas en SOAP, se puede interactuar con UDDI tanto en tiempo de diseño como de ejecución para descubrir datos técnicos de los servicios que permitan invocarlos y utilizarlos. De este modo, UDDI sirve como infraestructura para una colección de software basado en Servicios Web.

2.2.7 Servicios Web y bibliotecas digitales.

Una definición muy general de Servicio Web (SW), se refiere a cualquier servicio que emplea el protocolo HTTP para recuperar contenido en lenguaje XML. El protocolo SOAP representa en la actualidad uno de los enfoques más empleados para tal fin. SOAP opera bajo un esquema por procedimientos, para lo cual especifica funciones, parámetros de funciones y valores de retorno. La operación de SOAP se apoya en el lenguaje WSDL, mismo que facilita la implementación SW en servidores y su consumo por parte de clientes. En contraparte, SOAP tiene la desventaja de mayor sobrecarga de mensajes. Las complicaciones que pueden surgir por el uso de este protocolo, pueden ser resueltas en la implementación particular de algún servidor HTTP, por ejemplo Apache [30]. Un ejemplo de una biblioteca digital basada en SOAP es *Search and Retrieve on the Web* (SRW por sus siglas en inglés) [31].

Los Servicios Web representan bloques interoperables en la construcción de aplicaciones, como lo puede ser por ejemplo una infraestructura de biblioteca digital. La implementación podría incluir servicios para: búsquedas, autenticación de usuarios, solicitudes de préstamos interbibliotecarios, traducción de documentos y pagos en línea. Estos servicios podrían combinarse con una aplicación de biblioteca digital particular, para ofrecer un ambiente que ofrezca específicos a una comunidad de usuarios definida.

A grandes rasgos, la arquitectura de una biblioteca digital es compatible con la arquitectura requerida en el soporte vía Servicios Web. La comunidad que accede a la biblioteca digital generalmente hace uso de verbos para publicar y ubicar documentos. La arquitectura de Servicios *Web* vincula directamente tales verbos con aplicaciones. De hecho, la infraestructura de Servicios *Web* puede ser concebida como una biblioteca digital que ofrece servicios por encima de información. Ciertos rasgos como el uso de meta datos para descubrimiento, autenticación y modelos de negocios para acceso a propiedad intelectual protegida, también son aplicables a Servicios *Web* y por ende, pueden ser resueltos en el contexto de éstos últimos.

Tipos de Servicios *Web* en biblioteca digitales.

Desde la perspectiva posible de las necesidades de un usuario, a continuación se mencionan las funciones requeridas con mayor frecuencia. Estas pueden ser implementados mediante servicios *Web*, aprovechando los beneficios y ventajas antes mencionados.

Registro. Los servicios de registro proporcionan identificadores persistentes para conceptos u objetos; estos pueden proporcionar datos acerca de los datos (metadatos) en incluso datos sobre los metadatos. El registro puede ser sobre objetos físicos (entre los cuáles se incluyen personas o instituciones) así como digitales. Un ejemplo de servicio para registro son las extensiones a Dublin Core [32].

Búsquedas. Se trata del servicio reconocido con mayor familiaridad para la mayoría de los usuarios. El objetivo principal generalmente es obtener una lista de registros a partir de una cadena de búsqueda. Los servicios *Web* de búsqueda encapsulan esta funcionalidad y son consumidos por programas de alto nivel bajo petición de los usuarios.

Autenticación de usuarios. Servicio para identificar usuarios del sistema. En el caso del desarrollo de Servicios *Web*, la autenticación de usuarios no deja de ser un tema importante. Los Servicios *Web* pueden funcionar como parte central de cualquier aplicación, incluyendo a las biblioteca digitales, debido a que sirven como nexo entre una aplicación y otra o entre una aplicación y sus datos.

2.2.8 Tecnologías para desarrollo de Servicios *Web*.

Algunas de las herramientas que permiten la implementación de Servicios *Web* son las siguientes:

- **Microsoft .Net** [33]. El entorno .NET es una plataforma de nueva generación de Microsoft para generar aplicaciones *Web* y servicios *Web*. Se ha diseñado para satisfacer la necesidad de programadores y consumidores de Servicios *Web*, con una amplia compatibilidad para estándares *Web* como XML y SOAP.
- **Sun Java Enterprise System** [34]. Es una consolidación de la arquitectura de servicios en red de Sun Microsystems [35], para su uso en aplicaciones de Internet. Integra los componentes necesarios para la creación de servicios *Web* y los sistemas de red.
- **WebSphere** [36]. Ofrece soporte para J2EE y servicios *Web*, poniendo en uso los estándares de servicios *Web* que IBM ayudó a desarrollar para que los clientes puedan cumplir tareas de integración complejas. Utilizando los servicios *Web*, los desarrolladores pueden diseñar gráficamente y controlar el flujo de aplicaciones y transacciones complejas y diversas, en toda la red.

- **Borland *Web Services Kit for Java*** [37]. Diseñado para su integración con Borland JBuilder, el entorno de desarrollo de java, simplifica el desarrollo de servicios *Web* basados en estándares SOAP, WSDL, UDDI y WS-Inspection.
- **Axis** [38]. Es una iniciativa de código abierto de la Apache Software Foundation[33]. Axis es un motor SOAP compatible JAX-RPC [39]. Puede integrarse con contenedores *Web* como Tomcat [40], lo que permite utilizar varias de sus características como la seguridad, los repositorios de recursos, multi-threads [41], etc.

Capítulo III

Análisis y Diseño

La palabra biblioteca. (del griego βιβλιοθήκη *biblion* = libro y *thekes* = caja). Literalmente significa "caja de libros" o de guardar libros, pero la palabra "caja" se sustituye por edificio, departamento o habitación. Pero una buena definición de la palabra es: **"Institución cuya finalidad consiste en la adquisición, conservación, estudio y exposición de libros y documentos"** [42]. Esta definición de biblioteca hace referencia a un conjunto de servicios que provee una biblioteca tradicional, pero que se pueden aplicar a otros esquemas de bibliotecas. La utilización de la computadora en las bibliotecas permitió simplificar muchas de las tareas y servicios que presta una biblioteca a la comunidad, por ejemplo permitió registrar los libros en una base de datos para que la consulta sea más ágil para el usuario. Posteriormente las bibliotecas evolucionaron, ya no eran lugares donde solo se almacenan libros, una buena biblioteca debía tener también videos, programas, CD-ROMs, discos láser, etc. Por lo cual en muchos lugares se les cambio el nombre de Biblioteca a centro de medios. Posteriormente la Internet vuelve a hacer una transformación en las actividades que una biblioteca ofrece, al hacerla accesible en cualquier lugar y en cualquier momento. La liberación de los denominados Servicios *Web* ha traído consigo una redefinición de la arquitectura de las denominadas bibliotecas digitales, ya que permiten una mayor interoperabilidad entre diferentes sistemas, además de que cualquier aplicación sea *Web* o de escritorio, puede consumir estos servicios ofreciendo posibilidades como clientes ricos en medios e independientes de la plataforma que utilicen. La finalidad de la presente tesis fue desarrollar una biblioteca digital utilizando el estándar del Dublin Core y hacer uso de los Servicios *Web*, mostrando los beneficios que estos ofrecen en las aplicaciones basadas en *Web*.

Este capítulo muestra las fases realizadas de análisis y diseño para la arquitectura de la aplicación desarrollada utilizando UML (*Unified Model Language*, Lenguaje Unificado de Modelado). El diseño plantea el dominio y reconocimiento del problema, mientras que el diseño abarca el dominio de la solución y la forma de implementación.

3.1 Análisis del sistema

Entre los servicios comunes que una biblioteca digital puede proporcionar se encuentran los siguientes:

- Registro de materiales
- Registro de usuarios
- Autenticación de Usuarios
- Búsquedas de materiales

En la mayoría de las biblioteca digitales estos servicios son implementados mediante *middleware* con tecnologías tales como JSP [43], Servlets [44], PHP [45] entre otras y son parte fundamental de cada aplicación, además de no ser reutilizable para otros tipos de tecnologías. La propuesta de esta tesis, es crear estos servicios a través de la tecnología de Servicios *Web*, con lo que se pretende mejorar la interoperabilidad, facilitando a distintas aplicaciones de bibliotecas digitales implementadas en distintos lenguajes de programación ser ejecutadas sobre cualquier plataforma con tan solo utilizar los Servicios *Web* implementados, que finalmente darán la lógica y el funcionamiento a dichas aplicaciones.

3.1.1 Objetivos del sistema

El desarrollo de este proyecto contempla la implementación de una biblioteca digital cuya funcionalidad permita lograr los siguientes objetivos:

- Implementar la funcionalidad de BD con base al estándar de metadatos especificada por el Dublin Core.
- Habilitar la posibilidad de acceder al sistema de BD a través de la Web
- Proveer una interfaz programática para el acceso a la lógica de la BD sin requerir de un cliente específico para permitir el acceso a la misma desde aplicaciones de terceros.
- Brindar al menos una interfaz de usuario que brinde acceso a la funcionalidad de la BD y que:
 - Cuento con un módulo, que permita mediante interfaces graficas a un usuario realizar operaciones de administración de usuarios y recursos de la biblioteca digital.
 - Cuento con un módulo, que permita a un usuario recuperar y visualizar los recursos existentes en la base de datos, además de ofrecer diferentes criterios de búsqueda y ordenamiento de los recursos existentes.

3.1.2 Requerimientos

A continuación se da una descripción de las principales funciones que se espera lograr por parte del sistema de biblioteca digital.

3.1.2.1 Módulo de administración

Este módulo se encarga principalmente de la administración de los materiales electrónicos que están contenidos en la biblioteca y también de la administración de usuarios.

Las tareas cubiertas en esta parte de la aplicación son:

- **Administración de materiales:** en esta sección se presenta una forma de registro de recursos electrónicos a un usuario administrador, estos recursos pueden ser cualquier tipo de archivo (doc, pdf, txt, html, avi, etc), que contiene la redacción de algún artículo de carácter científico, que previamente fue presentado en algún congreso de divulgación. El usuario administrador debe de llenar la forma de registro presentada, con la información de los meta datos del artículo, como los son: el título del documento, autor, palabras claves, descripción, editor, fecha de registro, formato, identificador o ISBN, y el evento donde fue publicado el recurso, cabe señalar que estos meta datos son una descripción de referencia de los elementos del conjunto de Dublin Core [10]. Así mismo se tiene la opción de realizar modificaciones a los documentos registrados.
- **Administración de usuarios:** Este módulo permite al usuario administrador, dar de alta usuarios generales de la biblioteca. Se presenta una forma de registro donde deben llenarse los campos correspondientes: nombre, apellido paterno, apellido materno, grado de estudios, login y password del usuario. Esta información es necesaria para que un usuario pueda iniciar sesión, y pueda hacer uso de los servicios que la biblioteca digital proporciona. En este módulo también existe la opción de realizar cambios a la información de cada usuario.

3.1.2.2 Módulo de usuario principal

Este módulo se encarga de autenticar a los usuarios previamente registrados y les permite iniciar sesión con el sistema, con la finalidad de ofrecerles un espacio personalizado, en donde tienen la opción de hacer uso de diferentes servicios.

- **Buscador principal:** en esta sección el usuario puede realizar búsquedas de materiales, con solo proporcionar al sistema la palabra relacionada con el material a recuperar. El término proporcionado se comparará con la serie de meta datos que conforman cada registro del material y cuando este coincida con alguno de ellos se mostrará entonces una lista con la información principal de los materiales. Los resultados obtenidos pueden ser ordenados por título, autor o bien por fecha. Es posible mostrar los resultados, en forma extendida o resumida, lo que quiere decir que si se utiliza la primera opción se presentará también el resumen registrado para cada material. El usuario puede recuperar el documento completo, con tan solo seleccionar las ligas correspondientes y de esta forma almacenarlo en su propio equipo de cómputo. Otra opción que se tiene, es que se puede especificar en la interfaz el término proporcionado que deba de compararse con algún meta dato en específico (título, autor, palabras clave, descripción).
- **Buscador personalizado:** esta sección contiene los servicios mencionados en el punto anterior. La diferencia consiste en que el usuario puede seleccionar el material deseado y agregarlo a un librero personal.
- **Librero Personal:** se proporciona al usuario una lista de los materiales que ha seleccionado, y que forman parte de su librero personal. Puede también desde esta sección visualizar y recuperar el documento completo.

- **Contactos:** es módulo presenta al usuario una relación con la información referente a otros usuarios, quienes hasta ese momento tienen registrados en su librero personal los materiales que él. Esto con el propósito de poder establecer algún plan colaborativo.
- **Cierre de sesión:** esta opción permite al usuario el cierre de sesión y la salida del sistema.

3.1.3 Arquitectura del sistema

En una descripción generalizada, los clientes se comunican con la aplicación de la biblioteca digital vía *Web*, mediante una arquitectura cliente/servidor. El servidor contiene el *middleware* que proporciona los servicios a los usuarios y la lógica de negocios proporcionada los servicios *Web*. Estos a su vez realizan la validación correspondiente de solicitud, para procesar la petición del servicio y permitir establecer conexión con el manejador de base de datos para realizar operaciones de inserción, actualización y borrado de información, dentro de una base de datos que sigue el estándar Dublin Core.

En la Figura 3.1, se muestra funcionamiento que tiene la biblioteca digital. En el primer caso se aprecia como un cliente hace una invocación a la aplicación biblioteca digital, para posteriormente poder recuperar o registrar materiales ó recursos a una base de datos. En este caso la aplicación se encuentra el mismo servidor al igual que la base de datos, por lo que la definición de la conexión hacia la base de datos es fija. En el siguiente caso ó arquitectura se observa como una aplicación no *Web*, ingresa a un servidor de servicios *Web* UDDI (JUDDI), con la finalidad de recuperar la referencia de los servicios *Web*, que pueden darle la funcionalidad a la aplicación. En este caso la definición de conexión será dinámica, con la finalidad de demostrar la interoperabilidad que los servicios *Web* tienen al poder conectarse a diferentes bases de datos.

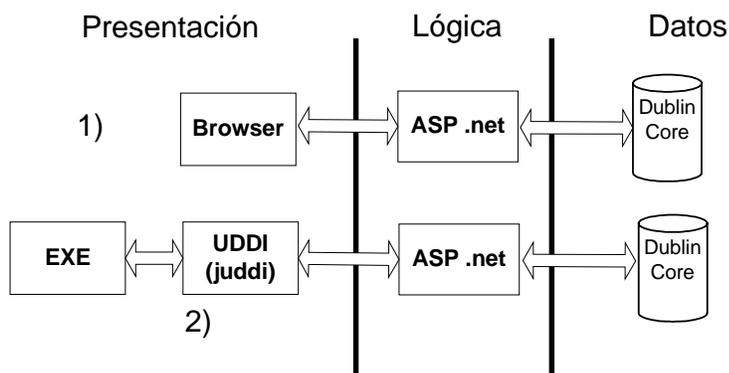


Figura 3.1 Arquitectura del Sistema

3.1.3.1 Casos de Uso

Después de establecer la arquitectura del sistema y tomando en cuenta los requerimientos establecidos, se presenta enseguida los diagramas UML, que se desarrollaron en la fase de análisis de la aplicación.

Actores:

- **Persona:** cualquier ser humano que utilice el sistema.
- **Administrador:** persona que se encarga de la administración del sistema, teniendo como principales tareas, el registro de recursos como de usuarios del sistema.
- **Usuario:** persona que explota el sistema, recuperando y administrando los recursos que la aplicación le proporciona desde un ambiente personalizado.

3.1.3.2 Diagrama de caso de uso general del sistema

La Figura 3.2 muestra los principales módulos de la aplicación a desarrollar, mediante diagramas de casos de uso.

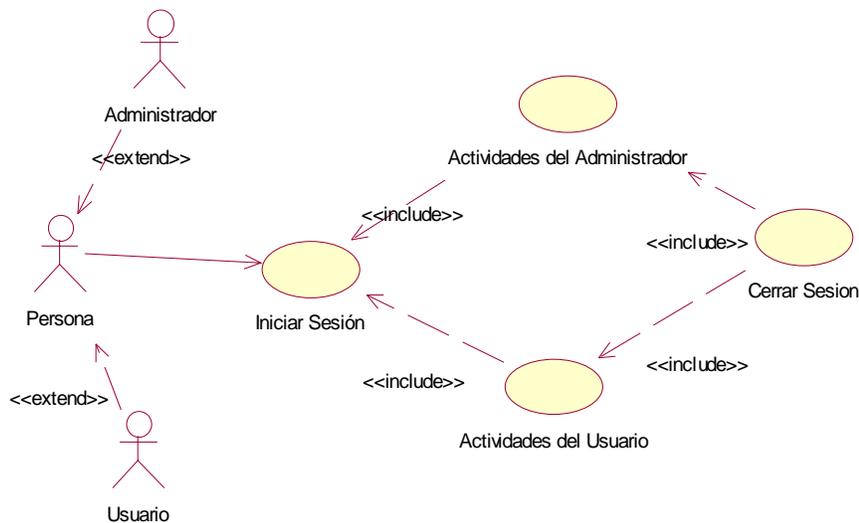


Figura 3.2 Diagrama de caso de uso general del Sistema

Como se observa en el caso de uso, existen dos tipos de personas (administrador y usuario principal) actores principales del sistema y que pueden iniciar o terminar alguna sesión dentro de la biblioteca digital, accediendo servicios que esta ofrece dependiendo del usuario de quien se trate.

Análisis de caso de uso general del sistema Curso Típico de Eventos

Acción del Actor	Respuesta del Sistema
1.- Este caso de uso inicia cuando una persona se identifica ante el sistema como usuario principal o administrador	2.- Solicita login y password de acceso.
3.- Introduce login y password de acceso	4.- Presenta el ambiente correspondiente a la persona identificada (administrador, usuario principal)
5.- La persona identificada realiza selección de actividades	6.- Procesa la actividad seleccionada
7.- Persona cierra sesión.	8.- Cierra sesión

Tabla 3.1 Análisis de caso de uso general del sistema.

-Cursos Alternativos

- **Línea 2:** usuario y clave de acceso incorrectos, se cancela acción y operación.
- **Línea 4:** se presenta el área correspondiente de trabajo para cada persona.

3.1.3.3 Diagrama de secuencia general del sistema

El siguiente diagrama de secuencia muestra la trayectoria ordenada de eventos que surgen entre la interacción del actor persona y el sistema principal. (Véase Figura 3.3)

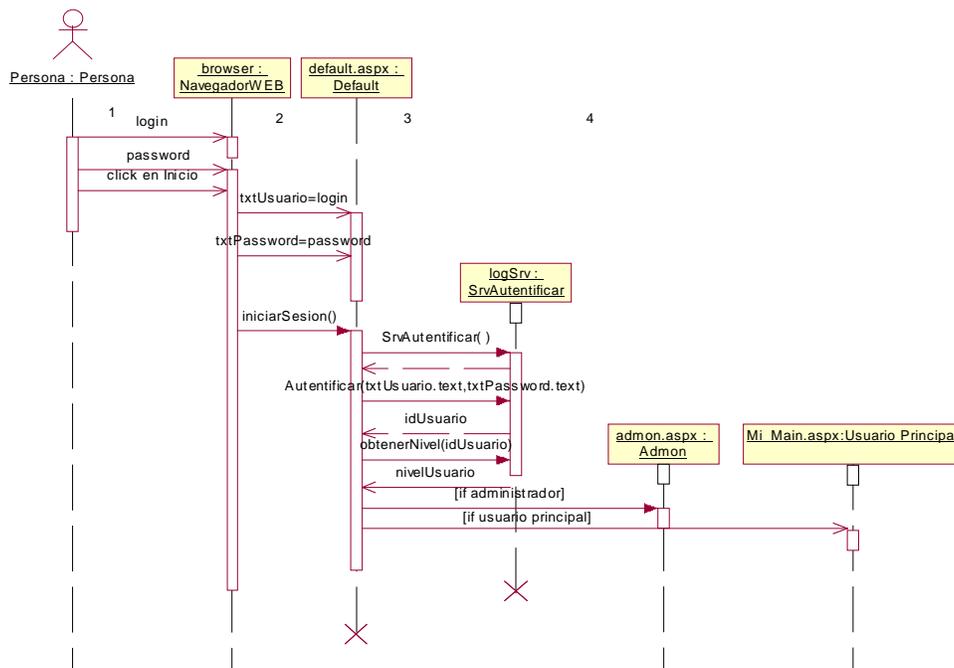


Figura 3.3 Diagrama de secuencia general del sistema

**Análisis de Caso de uso del usuario principal
Curso Típico de Eventos**

Acción del Actor	Respuesta del Sistema
1.-Este caso de uso inicia cuando una persona es identificada por el sistema como usuario principal.	2.-Inicia una sesión de usuario principal y presenta el ambiente correspondiente de actividades.
3.-Elige modulo personalizado de búsqueda de materiales.	4.-Muestra el entorno personalizado de búsqueda de materiales.
5.-Proporciona el termino necesario para realizar la búsqueda de materiales.	6.-El sistema recibe el termino y muestra una lista de materiales coincidentes con el termino al usuario principal.
7.-Agrega los materiales de su interés a su librero personal.	8.-Agrega el material al librero personal del usuario principal.
9.-Visualiza recursos de su librero personal.	10.-Muestra los recursos existentes en el librero personal.
11.-Elimina los recursos del librero personal.	12.-Elimina los recursos del espacio personal.
13.-Visualiza usuarios con recursos similares	14.-Muestra lista de usuarios que tienen los mismos materiales en su librero personal.
15.-Elige modulo principal de búsqueda	16.-Muestra modulo principal de búsqueda
17.-Proporciona termino de búsqueda. Elige mostrar resultados ordenados en forma libre.	18.-Recibe término de búsqueda y muestra lista de recursos ordenada en forma libre.
19.-Proporciona termino de búsqueda. Elige mostrar resultados ordenados por fecha.	20.-Recibe término de búsqueda y muestra lista de recursos ordenada por fecha.
21.-Proporciona termino de búsqueda. Elige mostrar resultados ordenados por autor.	22.-Recibe término de búsqueda y muestra lista de recursos ordenada por autor.
23.-Proporciona termino de búsqueda. Elige mostrar resultados ordenados por titulo del recurso.	24.-Recibe término de búsqueda y muestra lista de recursos ordenada por título del recurso.
25.-Proporciona término de búsqueda. Selecciona búsqueda en metadato. Elige buscar en metadato titulo.	26.-Recibe término de búsqueda. Compara término con metadato titulo. Muestra resultados.
27.- Proporciona término de búsqueda. Selecciona búsqueda en metadato. Elige buscar en metadato autor.	28.-Recibe término de búsqueda. Compara término con metadato autor. Muestra resultados.
29.-Proporciona término de búsqueda.	30.-Recibe término de búsqueda. Compara

<p>Selecciona búsqueda en metadato. Elige buscar en metadato descripción.</p> <p>31.- Proporciona término de búsqueda. Selecciona búsqueda en metadato. Elige buscar en metadato palabras claves.</p>	<p>término con metadato descripción. Muestra resultados.</p> <p>32.-Recibe término de búsqueda. Compara término con metadato palabras claves. muestra resultados.</p>
---	---

Tabla 3.2 Análisis de caso de uso general del sistema.

3.1.3.5 Diagrama de secuencia de usuario principal

El siguiente diagrama de secuencia muestra la trayectoria ordenada de eventos que surgen entre la interacción del usuario principal y el sistema. Véase Figura 3.5

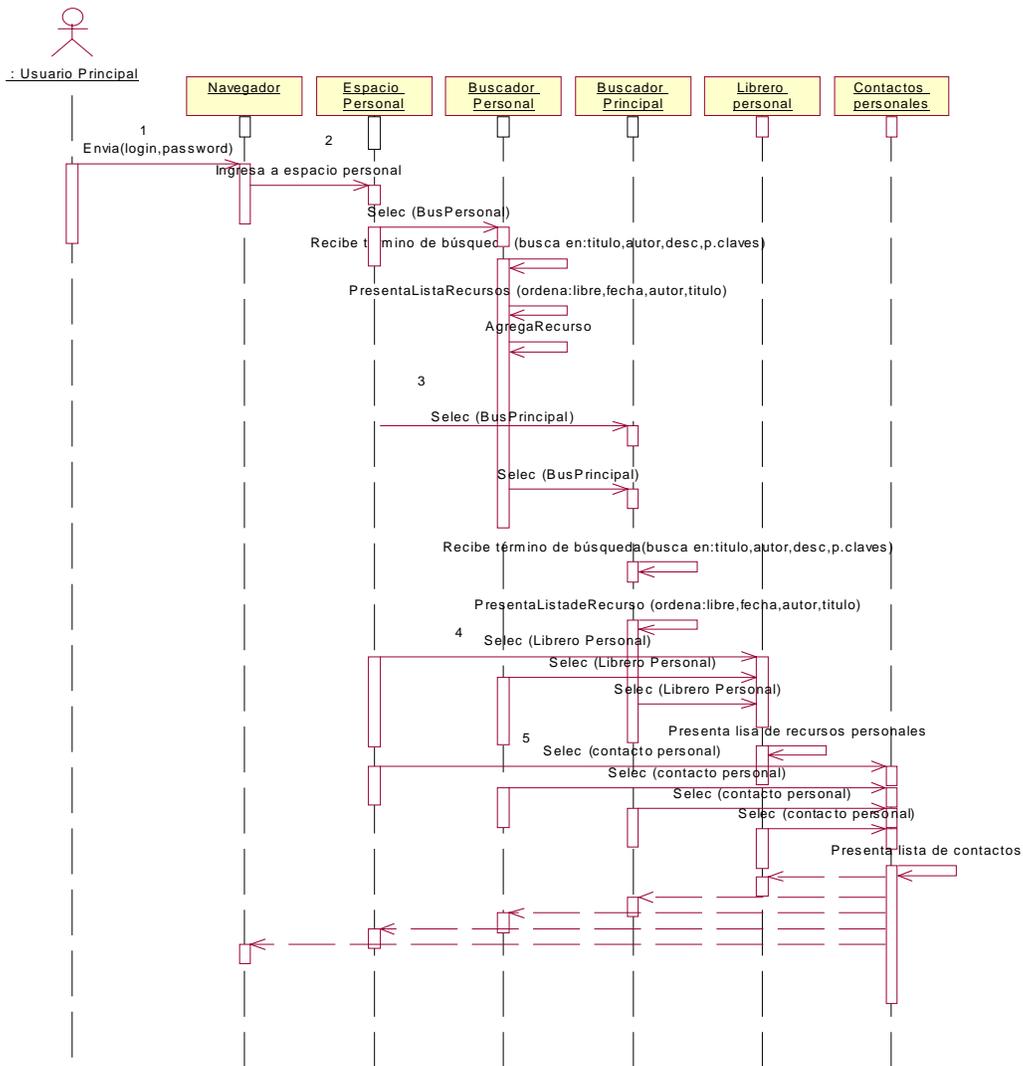


Figura 3.5 Diagrama de secuencia de actividades del usuario principal

Descripción general del diagrama de secuencia del usuario principal:

1. El usuario proporciona login y password y es identificado por el sistema. Se presenta el espacio personal de usuario principal.
2. Dentro del espacio personal, el usuario principal puede navegar hacia el buscador personal, enviar término y realizar búsqueda por metadatos (titulo, autor, descripción y palabras claves). Puede también buscar en cualquier metadato y realizar ordenamientos en forma libre, por fecha, autor y titulo. De la lista de recursos presentada, puede agregar materiales a un librero personal.
3. El usuario principal selecciona el buscador principal. Envía término y realiza búsqueda por metadatos (titulo, autor, descripción y palabras claves). Puede también buscar en cualquier metadato y realizar ordenamientos en forma libre, por fecha, autor y titulo.
4. El usuario principal selecciona el librero personal. Se presenta lista de recursos personales.
5. El usuario principal selecciona contactos personales. El sistema muestra una lista de usuarios principales con recursos en sus libreros iguales a los recursos del usuario principal actual.

3.1.3.6 Diagrama de casos de uso del administrador

La Figura 3.6 muestra el diagrama de casos de uso de un administrador del sistema y describe las actividades que el administrador puede realizar.

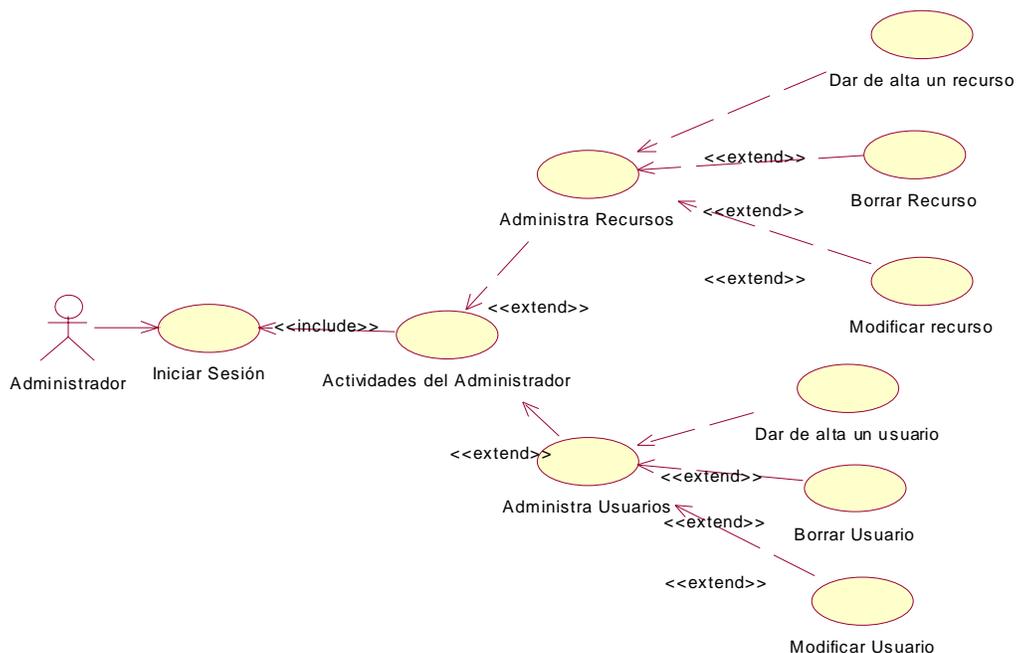


Figura 3.6 Diagrama de casos de uso del administrador

A continuación se presenta el análisis del diagrama de caso de uso anterior. Véase Tabla 3.3.

Diagrama de Caso de uso del administrador Curso Típico de Eventos	
Acción del Actor	Respuesta del Sistema
1.-Este caso de uso inicia cuando una persona es identificada por el sistema como administrador.	2.-Inicia una sesión de administrador y presenta el ambiente correspondiente de actividades.
3.-Elige modulo de alta de recursos.	4.-Muestra el entorno de alta de recursos.
5.-Proporciona los metadatos necesarios para el registro del recurso o material. Realiza registro.	6.-Recibe y guarda la información del recurso o material.
7.-Elige alta de usuarios.	8.-Muestra el entorno de alta de usuarios.
9.-Proporciona la información del usuario principal a registrar. Realiza registro.	10.-Recibe y guarda la información del usuario principal.
11.-Elige el modulo de bajas, cambios de recursos.	12.-Muestra el entorno de bajas, cambios de recursos.
13.-Realiza cambio de información o baja de recursos	14.-Guarda los cambios o realiza la baja pertinente del recurso o material.
15.-Elige modulo de bajas, cambios de usuarios.	16.-Muestra el entorno de bajas, cambios de usuarios.
17.-Realiza cambio de información o baja de usuarios.	18.-Guarda los cambio o realiza la baja pertinente de usuario.

Tabla 3.3 Análisis de caso de uso general del sistema.

3.1.3.7 Diagrama de secuencia del administrador

El siguiente diagrama de secuencia muestra la trayectoria ordenada de eventos que surgen entre la interacción del usuario principal y el sistema. (véase Figura 3.7)

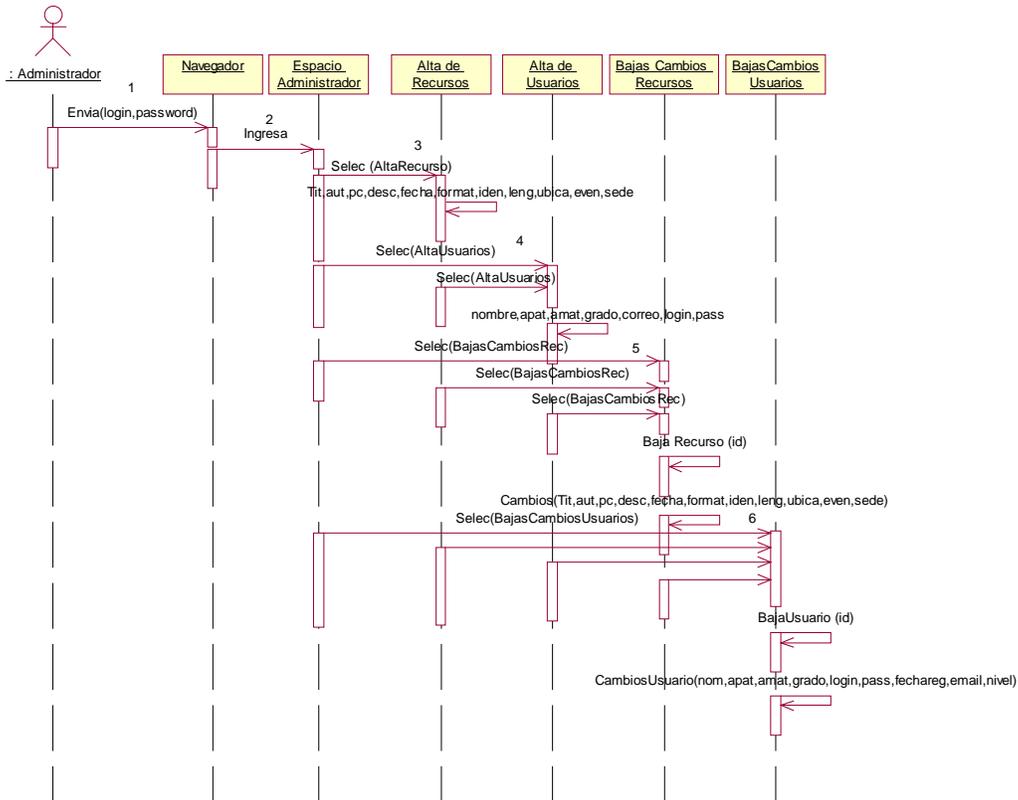


Figura 3.7 Diagrama de secuencia de actividades del administrador

Descripción general del diagrama de secuencia del administrador:

1. El administrador proporciona login y password y es identificado por el sistema. Se presenta el espacio personal del administrador.
2. El administrador ingresa a su espacio de actividades.
3. El administrador ingresa al espacio de actividades de alta de recursos, proporciona los metadatos del recurso (titulo, autor, palabras claves, descripción, fecha, formato, identificador, lengua, ubicación de archivo, evento y sede). Realiza el alta correspondiente.
4. El administrador ingresa al espacio de actividades de alta de usuarios, proporciona la información del usuario principal (nombre, apellido paterno, apellido materno, grado de estudios, correo, login y password). Realiza el alta correspondiente.

5. El administrador ingresa al espacio de actividades de bajas y cambios de recurso. Selecciona el recurso para eliminarlo, realiza el borrado. Proporciona nueva información del recurso y actualiza datos.
6. El administrador ingresa al espacio de actividades de bajas y cambios de usuario. Selecciona el usuario principal para eliminarlo, realiza el borrado. Proporciona nueva información del usuario y actualiza datos.

3.1.5 Diagrama relacional

La información de los usuarios, como la de los recursos disponibles en la biblioteca digital, está almacenada en una base de datos. La Figura 3.6 muestra el diagrama entidad-relación de la base de datos diseñada para el almacenamiento de la información del sistema. Cabe aclarar que el modelado de la base de datos esta apegado al estándar de Dublin Core que se aboca a definir un conjunto de elementos básicos para definir recursos electrónicos y facilitar su recuperación [10].

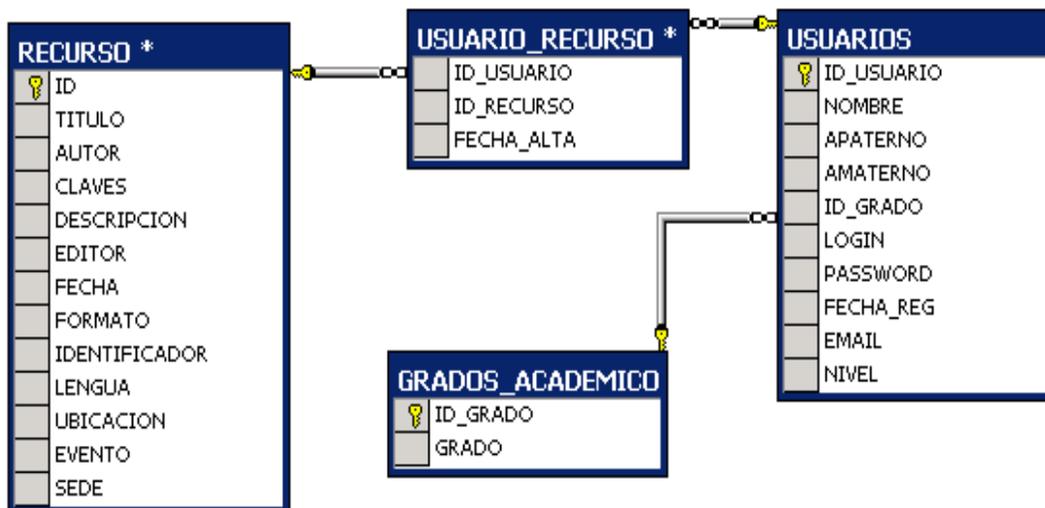


Figura 3.8 Diagrama Relacional.

A continuación se presenta una descripción de cada una de las entidades del modelo: Véase Tabla. 3.4

Tabla	Descripción													
<p>Recurso</p> <table border="1" data-bbox="298 369 613 722"> <tr> <td>Id: int</td> </tr> <tr> <td>Titulo: nvarchar(200)</td> </tr> <tr> <td>Autor: nvarchar(200)</td> </tr> <tr> <td>Claves: nvarchar(200)</td> </tr> <tr> <td>Descripcion: varchar(1000)</td> </tr> <tr> <td>Editor: nvarchar(100)</td> </tr> <tr> <td>Fecha: datetime</td> </tr> <tr> <td>Formato: nvarchar(15)</td> </tr> <tr> <td>Identificador: nvarchar()</td> </tr> <tr> <td>Lengua: nvarchar(2)</td> </tr> <tr> <td>Ubicacion: nvarchar(120)</td> </tr> <tr> <td>Evento: nvarchar(200)</td> </tr> <tr> <td>Sede: nvarchar(50)</td> </tr> </table>	Id: int	Titulo: nvarchar(200)	Autor: nvarchar(200)	Claves: nvarchar(200)	Descripcion: varchar(1000)	Editor: nvarchar(100)	Fecha: datetime	Formato: nvarchar(15)	Identificador: nvarchar()	Lengua: nvarchar(2)	Ubicacion: nvarchar(120)	Evento: nvarchar(200)	Sede: nvarchar(50)	<p>Almacena la información general de un recursos o material de la biblioteca digital:</p> <ul style="list-style-type: none"> • ID: identificador del recurso • Titulo: nombre del Titulo del recurso • Autor: nombre(s) del (os) autores del recurso o material • Claves: palabras claves registradas en el recurso • Descripción: descripción general del recurso • Editor: nombre del editor o editorial del recurso • Fecha : fecha de registro del recurso • Formato: tipo de formato del recurso almacenado • Identificador: ISBN o folio del recurso presentado • Lengua: idioma en el que esta escrito el recurso • Ubicación: ruta del lugar donde se guardará el archivo físico del recurso. • Evento: en donde se presento el recurso • Sede: lugar donde ocurrió el evento
Id: int														
Titulo: nvarchar(200)														
Autor: nvarchar(200)														
Claves: nvarchar(200)														
Descripcion: varchar(1000)														
Editor: nvarchar(100)														
Fecha: datetime														
Formato: nvarchar(15)														
Identificador: nvarchar()														
Lengua: nvarchar(2)														
Ubicacion: nvarchar(120)														
Evento: nvarchar(200)														
Sede: nvarchar(50)														
<p>Usuarios</p> <table border="1" data-bbox="298 888 613 1241"> <tr> <td>Id_usuario: int</td> </tr> <tr> <td>Apaterno: nvarchar(50)</td> </tr> <tr> <td>Amaterno: nvarchar(50)</td> </tr> <tr> <td>Nombre: char(18)</td> </tr> <tr> <td>Id_Grado: int</td> </tr> <tr> <td>Login: nvarchar(20)</td> </tr> <tr> <td>Password: nvarchar(20)</td> </tr> <tr> <td>Fecha_Reg: datetime</td> </tr> <tr> <td>Email: nvarchar(50)</td> </tr> <tr> <td>Nivel: char(1)</td> </tr> </table>	Id_usuario: int	Apaterno: nvarchar(50)	Amaterno: nvarchar(50)	Nombre: char(18)	Id_Grado: int	Login: nvarchar(20)	Password: nvarchar(20)	Fecha_Reg: datetime	Email: nvarchar(50)	Nivel: char(1)	<p>Almacena la información general de un recursos o material de la biblioteca digital:</p> <ul style="list-style-type: none"> • Id_Usuario: identificador del usuario • Apaterno: apellido paterno del usuario • Amaterno: apellido materno del usuario • Nombre: nombre del usuario • Id_Grado: identificador de grado de estudios del usuario • Login: clave de usuario para ingresar al sistema. • Password: password de usuario • Fecha_Reg: fecha cuando se fue llevado el registro del usuario • Email: correo electrónico del usuario • Nivel: nivel de usuario. (U,A) usuario, administrador. 			
Id_usuario: int														
Apaterno: nvarchar(50)														
Amaterno: nvarchar(50)														
Nombre: char(18)														
Id_Grado: int														
Login: nvarchar(20)														
Password: nvarchar(20)														
Fecha_Reg: datetime														
Email: nvarchar(50)														
Nivel: char(1)														
<p>Usuario_Recurso</p> <table border="1" data-bbox="298 1371 613 1514"> <tr> <td>Id_Usuario: int</td> </tr> <tr> <td>Id_Recurso: int</td> </tr> <tr> <td>Fecha_Alta: datetime</td> </tr> </table>	Id_Usuario: int	Id_Recurso: int	Fecha_Alta: datetime	<p>Se almacenan los recursos o materiales que cada usuario selecciona:</p> <ul style="list-style-type: none"> • Id_Usuario: identificador del usuario • Id_Recurso: identificador del recurso seleccionado • Fecha_Alta: fecha cuando fue seleccionado el material o recurso por el usuario. 										
Id_Usuario: int														
Id_Recurso: int														
Fecha_Alta: datetime														
<p>Grados_Academicos</p> <table border="1" data-bbox="298 1654 613 1757"> <tr> <td>Id_Grado: char(18)</td> </tr> <tr> <td>Grados: char(18)</td> </tr> </table>	Id_Grado: char(18)	Grados: char(18)	<p>Catalogo de grados académicos de los usuarios.</p> <ul style="list-style-type: none"> • Id_Grado: identificador de los grados de estudios • Grados: descripción del grado académico. 											
Id_Grado: char(18)														
Grados: char(18)														

Tabla 3.4 Entidades del Modelo

3.1.6 Paquetes

Los paquetes que forman parte del sistema están conformados por las diferentes clases, que otorgan la funcionalidad a la aplicación. El diagrama de paquetes se muestra en la Figura 3.9.

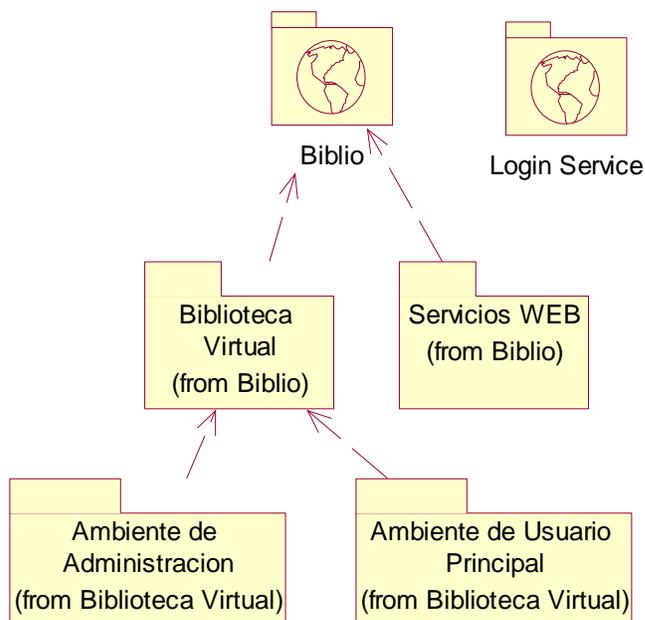


Figura 3.9 Diagrama de paquetes

Descripción:

- **Biblio:** contiene las clases principales de la aplicación.
- **Login service:** contiene las clases principales que hacen posible la autenticación de usuarios.
- **Biblioteca virtual:** contiene las clases, que dan funcionalidad a las interfaces gráficas del sistema de la biblioteca digital.
- **Ambiente de administración:** contiene las clases que proporcionan la interfaz gráfica del ambiente de administración.
- **Ambiente de usuario principal:** contiene las clases que proporcionan la interfaz gráfica del usuario principal.
- **Servicios Web:** contiene las clases de servicios *Web*, que dan la lógica de funcionamiento a la aplicación en general.

3.1.7 Diagramas de clases

A continuación se describen en forma general los diagramas de clases, que generan los ambientes antes mencionados, y se da una breve explicación de los métodos en ellas contenidos.

3.1.7.1 Diagrama de clases del modulo de administración

Las clases a continuación descritas forman parte del paquete de "ambiente de administración" y se muestran en la Figura 3.10, primordialmente cumplen con las tareas de administrar, a los usuarios del sistema y los materiales o recursos que se almacenan en la biblioteca digital.

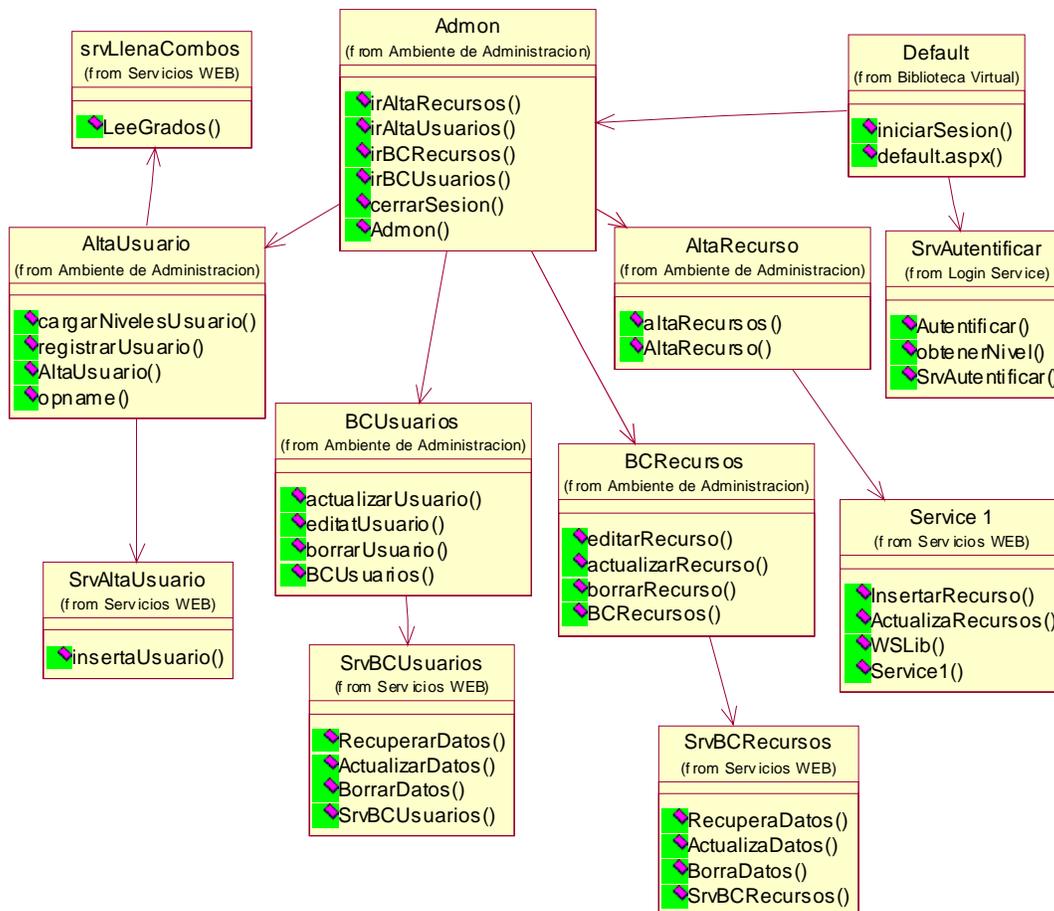


Figura 3.10 Diagrama de clases para "Ambiente Administración"

Descripción de clases:

- **Admón:** dirige al usuario administrador a la interfaz correspondiente, según la operación que este desee realizar. Estas opciones pueden ser: el alta de un usuario, recurso ó material al sistema, o bien la actualización o modificación de la información del usuario o recurso.
- **SrvLlenaCombos:** esta clase proporciona la información necesaria a las interfaces que incluyen componente de listas de valores, en donde un usuario debe elegir una opción de entre varias. Realiza una conexión con la base de datos, recuperando la información de los catálogos del modelo entidad-relación. Por ejemplo los grados de estudio.
- **AltaUsuario:** esta clase genera una interfaz para el registro de usuarios. Utiliza las clases *srvLlenaCombos*, y *SrvAltaUsuarios* para completar su lógica de funcionamiento.
- **SrvAltaUsuario:** esta clase tiene la función de insertar la información correspondiente de un usuario en la base de datos que soporta el sistema.
- **BCUsuarios:** proporciona la opciones de edición, actualización y borrado de usuarios del Sistema, utilizando desde luego las clases generadas por los servicios *Web*.
- **SrvBCUsuarios:** esta clase, da la lógica y funcionamiento a la clase *BCUsuarios*.
- **Default:** identifica a un usuario administrador, y lo dirige al módulo correspondiente dentro de la aplicación, utiliza la clase *SrvAutenticar* para realizar la operación de autenticación del usuario correspondiente.
- **AltaRecurso:** proporciona la interfaz necesaria al administrador, para que este pueda registrar un recurso o material en el sistema.
- **SrvAutenticar:** identifica si un usuario administrador, se encuentra registrado en el sistema.
- **BCRecursos:** proporciona las opciones de edición, actualización y borrado de usuarios del sistema, utilizando desde luego las clases generadas por los servicios *Web*, utiliza para su funcionamiento la clase *SrvBCRecursos*.
- **SrvBCRecursos:** proporciona la lógica, para realizar las operaciones necesarias a la clase *BCRecursos*.
- **Service1:** clase que proporciona la funcionalidad y la lógica para registrar un nuevo recurso en la base de datos.

3.1.7.2 Diagrama de clases modulo usuario principal

A continuación se describen las clases contenidas en el paquete "ambiente de usuario principal". Figura 3.11. Tienen principalmente la función de proporcionar un ambiente que permita explotar el contenido de materiales y recursos existentes en la base datos. Ofrecen servicios de búsqueda de materiales, así como información de usuarios y sus respectivos materiales.

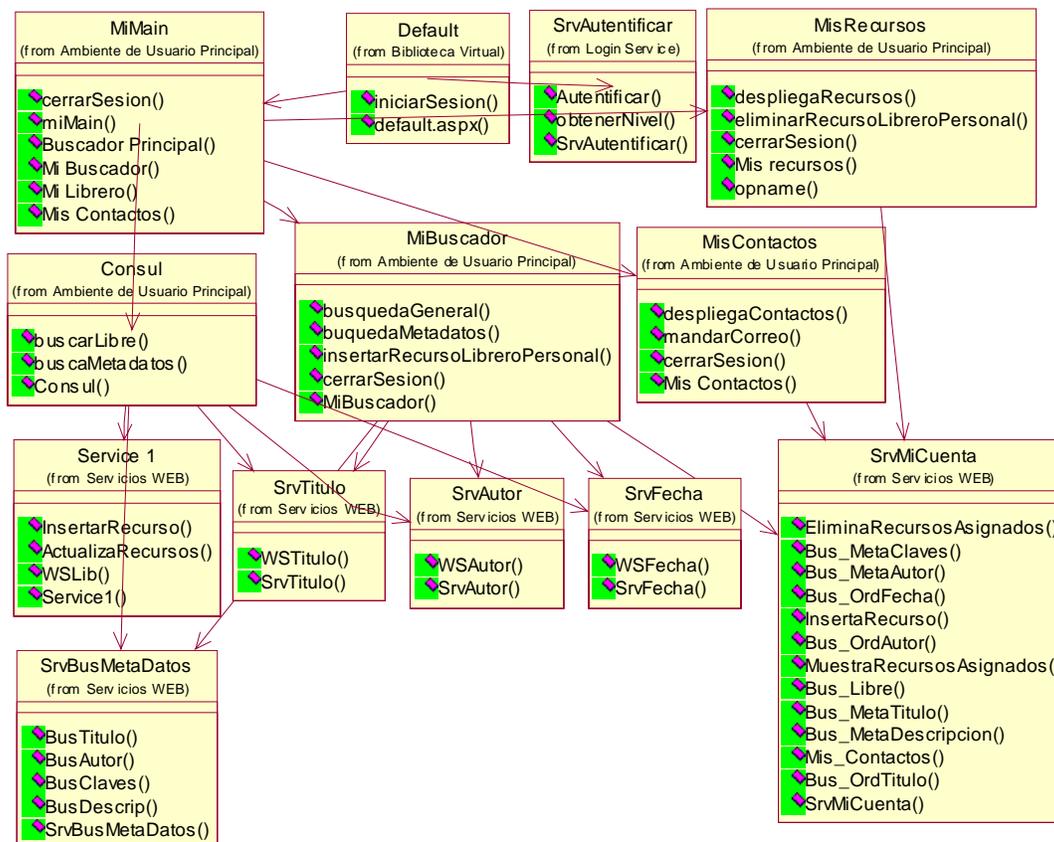


Figura 3.11 Diagrama de clases para "Ambiente Usuario Principal"

Descripción de clases:

- **MiMain:** clase que presenta la interfaz de inicio a un usuario principal. Tiene la finalidad de dirigir al usuario hacia pantallas que ofrecen diferentes opciones, como por ejemplo los módulos de búsqueda general y personalizada, área de información de perfiles afines, y visualización de materiales en librero personal.

- **Default:** identifica a un usuario principal, y lo dirige al módulo correspondiente dentro de la aplicación, utiliza la clase SrvAutenticar para realizar la operación de autenticación del usuario correspondiente.
- **SrvAutenticar:** clase que identifica si un usuario principal, se encuentra registrado en el sistema.
- **Consul:** clase que proporciona la interfaz gráfica al usuario principal, conteniendo está diferentes opciones de búsqueda de recursos o materiales. Esta clase se vale de otras para cumplir su lógica y funcionamiento. (Service1, SrvBusMetadatos, SrvTitulo, SrvAutor, SrvFecha).
- **MiBuscador:** proporciona la interfaz gráfica, de un ambiente personalizado de búsqueda de materiales, en donde el usuario principal tiene la posibilidad de elegir de entre los resultados de la búsqueda, materiales y agregarlos a un librero personal. Esta clase se utiliza las clases SrvBusMetadatos, SrvTitulo, SrvAutor, SrvFecha.
- **MisContactos:** proporciona la interfaz gráfica, donde un usuario principal visualiza una lista de usuario, que han agregado los mismos materiales a sus respectivos librereros. Utiliza la clase SrvMiCuenta, para obtener la lógica y funcionamiento.
- **MisRecursos:** proporciona la interfaz gráfica, donde el usuario principal visualiza los recursos o materiales que el mismo ha asignado a su librero personal. Para su funcionamiento utiliza la clase SrvMiCuenta.
- **Service1:** esta clase contiene los métodos necesarios para poder registrar un nuevo recurso en la base de datos del sistema. Contiene también un método que permite realizar búsquedas libres.
- **SrvBusMetadatos:** esta clase contiene los métodos que permiten realizar búsquedas por meta datos. (titulo, autor, palabras claves, descripción).
- **SrvTitulo:** clase utilizada, para realizar búsquedas y ordenamiento de resultados únicamente por el meta dato titulo.
- **SrvAutor:** clase utilizada, para realizar búsquedas y ordenamiento de resultados en base al meta dato autor.
- **SrvFecha:** clase utilizada, para realizar búsquedas y ordenamiento de resultados en base al meta dato fecha.
- **SrvMicuenta:** esta clase es utilizada por el espacio personalizado del usuario principal (miBuscador, misRecursos, misContactos). Contiene métodos que proporcionan lógica y funcionalidad a estas secciones, como por ejemplo: Búsqueda y presentación de recursos por meta datos, diferentes opciones de ordenamiento de información, e inserción y eliminación de recursos del librero personal del usuario.

Capítulo IV

Implementación, Pruebas y Resultados

Es fundamental que las aplicaciones basadas en Servicios Web sean totalmente confiables. Un Servicio Web debe estar siempre disponible para la aplicación que lo requiera. No debe provocar errores, perder solicitudes, sufrir errores ante solicitudes no válidas ni dañar datos almacenados. Siempre debe poder satisfacer las necesidades del cliente con un rendimiento aceptable. En el caso raro de que se produzca un error, el servicio Web debería seguir procesando solicitudes lo mejor posible. Es decir, un servicio Web necesita todas las "características" –escalabilidad, confiabilidad, disponibilidad, etc. Si un servicio Web no es confiable, no serían utilizados.

Para cumplir los requerimientos establecidos y tomando en cuenta lo mencionado anteriormente, se decidió utilizar el entorno .NET Framework[47], es una plataforma de nueva generación de Microsoft para generar aplicaciones web y de servicios web, con una amplia compatibilidad para estándares Web como XML y SOAP. Algunas de las características claves para la utilización de .NET Framework son:

- Un *Common Language Runtime* por lo que puede administrar las necesidades de ejecución de código programado en cualquier lenguaje de programación.
- *ADO .NET*, que proporciona clases para tener acceso a documentos XML y almacenes de datos relacionales.
- *ASP.NET*, proporciona un modelo de programación de bajo nivel para desarrollar aplicaciones Web fáciles de implementar, conocidos como Web Forms y Servicios Web pudiéndose implementar en C# o Visual Basic.

Los servicios web de ASP. NET son la tecnología recomendada para implementar servicios web basados en .NET Framework. Son compatibles con las solicitudes de servicio mediante SOAP a través de http, así como http GET o POST.

En esta sección se muestra la implementación del sistema de biblioteca digital, consumiendo servicios *Web*. La implementación se llevo a cabo utilizando el entorno de Visual Studio .NET[47], dado que es una herramienta que proporciona los elementos necesarios para el desarrollo de este tipo de aplicaciones. Se utilizo específicamente el lenguaje de Microsoft C#[47], dado que es un nuevo lenguaje de programación diseñado para crear un amplio número de aplicaciones, que se ejecutan en .NET Framework, es un lenguaje sencillo, moderno, proporciona diferentes tipos de seguridad y está orientado objetos. C# se presenta dentro del entorno de Visual Studio .NET como Visual C#. Utiliza plantillas de proyecto diseñadores, página de propiedades, asistentes de código, un modelo

de objetos y otras características. La biblioteca para programar en Visual C# es .NET Framework. En el aspecto de los Servicios Web resulta técnicamente mejor que su contraparte de *Java*[48], pero esto podría cambiar en el futuro cercano. Por el momento la mejor implementación de Servicios *Web* esta en la plataforma de Microsoft . NET.

4.1 Descripción del Sistema

El sistema de la biblioteca digital cuenta con los siguientes módulos:

Modulo de administrador:

- Alta de Recursos
- Alta de Usuarios
- Bajas-Cambios Recursos
- Bajas-Cambios Usuarios
- Cerrar Sesión

Modulo de Usuario Principal:

- Login
- Buscador Principal
- Mi buscador
- Mi librero
- Mis Contactos (Perfil Afín)
- Cerrar Sesión

Cada uno de los módulos accede a una base de datos de Sql Server que proporciona la información al sistema. La conexión hacia este manejador de base de datos esta especificada en el archivo WebConfig general de la aplicación. Hay que tomar en cuenta que cada programa .aspx que proporciona la interfaz grafica correspondiente tiene asignada una variable dinámica de conexión, la cuál guarda la referencia. Si se necesitará que la aplicación se conectar a otra base de datos, del mismo tipo, sería suficiente cambiar la cadena de conexión en el archivo general de la aplicación (Web.Config). Véase Anexo 1.

4.1.1 Módulos de administración y usuario principal

En esta sección se describen las interfaces graficas que conforman la biblioteca digital, las cuales fueron implementadas utilizando la tecnología de .NET, en específico Web Forms (Formas Web). Se describen también los servicios web que cada una de estas formas consumen para su funcionamiento.

4.1.1.1 Web Forms (aspx). Modulo de administrador.

Nombre de la Forma:	Admon.aspx
Servicios Web consumidos:	--
Descripción:	Página principal del modulo de administración, contiene las ligas correspondiente a la navegación de las demás secciones. (Alta de Recursos, Alta de Usuarios, BCRecursos BCUusuarios, Inicio)

Nombre de la Forma:	AltaUsuario.aspx
Servicios Web consumidos:	SrvAltaUsuario.asmx, SrvLlenaCombos.asmx
Descripción:	Presenta al usuario administrador la forma de registro de un usuario.

Nombre de la Forma:	AltaRecurso.aspx
Servicios Web consumidos:	Service1.asmx
Descripción:	Presenta al administrador la forma de registro de un recurso o material

Nombre de la Forma:	BCUusuarios.aspx
Servicios Web consumidos:	SrvBCUusuarios.asmx
Descripción:	Presenta al administrador la forma donde puede este realizar Operaciones de actualización y borrado de usuarios.

Nombre de la Forma:	BCRecursos.aspx
Servicios Web consumidos:	SrvBCRecursos.asmx
Descripción:	Presenta al administrador la forma donde puede este realizar Operaciones de actualización y borrado de registros.

4.1.1.2 Servicios Web (asmx). Modulo de admistrador

Servicio Web:	SrvAutenticar.asmx
Métodos (1)	Autenticar()
Parámetros que recibe:	Login, password del usuario
Regresa:	Idusuario
Descripción:	Verifica que un usuario este registrado en la base de datos. Es utilizado por Default.aspx (Anexo 1), que dependiendo lo devuelto por el servicio web, permite a un usuario el ingreso a la sección correspondiente (administrador,usuario), o bien en caso de que el usuario no este registrado, la aplicación restringe el acceso.

Servicio Web:	SrvBCUsuario.aspx
Métodos	RecuperarDatos(),BorraUsuario(),ActualizarDatos()
Parámetros que recibe:	IdUsuario
Regresa:	--
Descripción:	Realiza operaciones de Consulta, borrado y actualización de registro de Usuarios.

Servicio Web:	SrvBCRecursos.aspx
Métodos	RecuperarDatos(),BorraRecurso(),ActualizarDatosRecurso()
Parámetros que recibe:	IdRecurso
Regresa:	--
Descripción:	Realiza operaciones de Consulta, borrado y actualización de recursos.

Servicio Web:	SrvAltaUsuario.aspx
Métodos	InsertaUsuario()
Parámetros que recibe:	Datos del Usuario
Regresa:	--
Descripción:	Realiza operación de registro de usuarios.

Servicio Web:	SrvLlenaCombos.aspx
Métodos	LeeGrados()
Parámetros que recibe:	--
Regresa:	DataSet
Descripción:	Hace una consulta de grados académicos y devuelve un DataSet con esos datos.

Servicio Web:	Service1.aspx
Métodos	InsertaRecurso()
Parámetros que recibe:	Datos del Recurso
Regresa:	--
Descripción:	Realiza el registro de un nuevo recurso

4.1.2.1 Web Forms (aspx). Modulo de usuario principal

Nombre de la Forma:	MiMain.aspx
Servicios Web consumidos:	--
Descripción:	Página principal del modulo de usuario principal, contiene las ligas correspondientes a la navegación de sus opciones (Consul, Mi Buscador, Mis Contactos, Mis Recursos)

Nombre de la Forma:	Consul.aspx (Anexo 2)
Servicios Web consumidos:	Service1, SrvTitulo, SrvAutor, SrvFecha, SrvBusMetadatos (Anexo 2)
Descripción:	Página principal de búsquedas generales para el usuario principal. Presenta varias opciones de búsqueda

	de materiales. El usuario puede visualizar archivos físicos de recursos seleccionados.
--	--

Nombre de la Forma:	MiBuscador.aspx
Servicios Web consumidos:	SrvTitulo, SrvAutor, SrvFecha, SrvBusMetadatos, SrvMiCuenta
Descripción:	Sección Personalizado de búsqueda de materiales. También Se presenta diferentes opciones de búsqueda. El usuario Puede seleccionar materiales.

Nombre de la Forma:	MisRecursos.aspx
Servicios Web consumidos:	SrvMiCuenta
Descripción:	Permite visualizar a un usuario los materiales asignados por sí mismo, a la vez que puede eliminarlos de su espacio personal.

Nombre de la Forma:	MisContactos.aspx
Servicios Web consumidos:	SrvMiCuenta
Descripción:	Presenta al usuario principal, una lista de usuarios que tienen registrados los mismos materiales que él.

4.1.2.2 Servicios Web (asmx). Modulo usuario principal

Servicio Web:	Service1.asmx
Métodos	WSLib()
Parámetros que recibe:	Cadena de búsqueda
Regresa:	DataSet
Descripción:	Dada una cadena, realiza búsqueda comparando el parámetro recibido en los campos de metadatos del recurso.

Servicio Web:	SrvTitulo.asmx
Métodos	WSTitulo()
Parámetros que recibe:	Cadena de búsqueda
Regresa:	DataSet
Descripción:	Dada una cadena, realiza búsqueda comparando el parámetro recibido en metadato Titulo.

Servicio Web:	SrvAutor.asmx
Métodos	WSAutor()
Parámetros que recibe:	Cadena de búsqueda
Regresa:	DataSet
Descripción:	Dada una cadena, realiza búsqueda comparando el parámetro recibido en metadato Autor.

Servicio Web:	SrvFecha.aspx
Métodos	WSFecha()
Parámetros que recibe:	Cadena de búsqueda
Regresa:	DataSet
Descripción:	Dada una cadena, realiza búsqueda comparando el parámetro recibido en metadato Fecha.

Servicio Web:	SrvBusMetadatos.aspx
Método (1)	BusTitulo
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Utilizado por Consul.aspx, para realizar búsquedas, a partir de una cadena de caracteres. El Servicio web hace una búsqueda libre, en el metadato Titulo
Método (2)	BusAutor
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Utilizado por Consul.aspx, para realizar búsquedas, a partir de una cadena de caracteres. El Servicio web hace una búsqueda libre, en el metadato Autor
Método (3)	BusClaves
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Utilizado por Consul.aspx, para realizar búsquedas, a partir de una cadena de caracteres. El Servicio web hace una búsqueda libre, en el metadato Palabras claves
Método (4)	BusDescrip
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Utilizado por Consul.aspx, para realizar búsquedas, a partir de una cadena de caracteres. El Servicio web hace una búsqueda libre, en el metadato Descripción

Servicio Web:	SrvMiCuenta.aspx
Método (1)	EliminaRecursosAsignados
Parámetros que recibe:	Idusuario, idrecurso
Regresa:	--
Descripción:	Elimina un recurso del librero personal
Método (2)	BusMetaClaves
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Se utiliza en el espacio de "Mi buscador", para realizar Búsqueda de materiales
Método (3)	BusMetaAutor
Parámetros que recibe:	Cadena de caracteres

Regresa:	DataSet
Descripción:	Realiza búsqueda en el metadato Autor
Método (4)	BusOrdFecha
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Realiza una búsqueda ordenando los resultados por fecha
Método (5)	InsertaRecurso
Parámetros que recibe:	Idusuario, idrecurso
Regresa:	DataSet
Descripción:	Inserta un recurso al librero personal
Método (6)	BusOrdAutor
Parámetros que recibe:	Idusuario, idrecurso
Regresa:	--
Descripción:	Elimina un recurso del librero personal
Método (7)	MuestraRecursosAsignados
Parámetros que recibe:	Idusuario
Regresa:	DataSet
Descripción:	Se utiliza en el espacio de "Mi Librero", para visualizar Recursos asignados de un usuario
Método (8)	BusLibre
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Realiza búsqueda en los metadato titulo,autor,descrip,p.claves, fecha.
Método (9)	BusMetaTitulo
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Realiza una búsqueda en el titulo del recurso.
Método (10)	BusMetaDescrip
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Realiza búsqueda en la descripción del recurso
Método (11)	Mis_Contactos
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Presenta al usuario principal, una lista de usuarios con lo mismos recursos seleccionados
Método (12)	Bus_Titulo
Parámetros que recibe:	Cadena de caracteres
Regresa:	DataSet
Descripción:	Busca en el Titulo del recurso

4.1.3 Localización y consumo dinámico de Servicios Web

La implementación de Servicios Web en el entorno Microsoft .NET, requiere que en tiempo de diseño, el programador agregue referencias Web a los servicios que se desea consumir. Estas referencias son procesadas por el entorno de trabajo Visual Studio .NET, y el resultado consiste de un conjunto de clases Proxy, que permiten la invocación de los métodos residentes en servicios remotos, como si se tratara de métodos locales.

Sin embargo, la indicación de referencias en tiempo de diseño, no es condicionante para el consumo de Servicios Web; éste último, también es realizable de forma dinámica, es decir, sin el conocimiento previo incluso del URL donde se localiza el servicio.

En este punto, UDDI juega un papel importante en su rol de registro público, diseñado para almacenar de forma estructurada información sobre empresas y los servicios que éstas ofrecen. La implementación de la presente Biblioteca Digital cuenta con una interfaz de escritorio basada en formularios Windows (WinForms), a través de la cual es posible realizar búsquedas por [título, autor, p.claves, descripción] haciendo uso del mismo Servicio Web que utiliza la aplicación Web. Estas operaciones se realizan sin previo conocimiento de la ubicación (URL) del Servicio Web correspondiente, en contraste con la implementación de interfaz basada en Web.

La aplicación de escritorio, hace uso de las bibliotecas Microsoft UDDI SDK Beta 2, para realizar consultas sobre un servidor UDDI que a su vez se apoya en la implementación jUDDI. La búsqueda se realiza empleando como parámetro la descripción del servicio como a continuación se muestra:

```
Inquire.Url = "http://148.204.45.77:8080/juddi/inquiry";  
FindService fs = new FindService();  
fs.Names.Add("es", "Busqueda por Metadatos");  
ServiceList sl = fs.Send();
```

Al obtener la ubicación (URL) del Servicio Web, se procede al consumo dinámico. Esta operación se realiza con ayuda de la biblioteca de clases DynWSLib 1.5 [49]. Se trata de una herramienta que genera los objetos proxy necesarios en tiempo de ejecución, a partir de la descripción WSDL obtenida desde la consulta a UDDI. La clase que se instancia es DynamicWebServiceProxy, misma a la que debe indicarse el URL del servicio web, el nombre del método que se va a ejecutar y los parámetros correspondientes.

La biblioteca DynWSLib se encarga de crear, enviar y procesar los mensajes SOAP requeridos para la comunicación entre el cliente y el servidor. Una gran ventaja de DynWSLib, es que reconstruye objetos complejos como los DataSet a partir del mensaje correspondiente.

La Figura 4.1 ilustra gráficamente el proceso de ubicación y consumo dinámico [50].

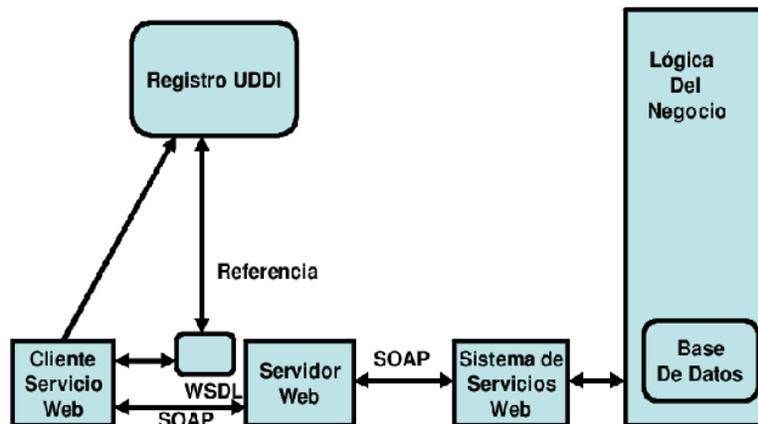


Figura 4.1 Ubicación y consumo dinámico de Servicios Web

La figura 4.1 muestra como el Servicio *Web* es construido y luego descrito por medio de WSDL y registrado bajo el estándar UDDI, el cliente busca en el registro UDDI (la descripción del Servicio) y obtiene el descriptor WSDL del servicio que necesita, lo invoca haciendo uso de SOAP el cual también es utilizado para comunicar la petición entre los diferentes componentes del servidor que aloja el servicio Web, para entregar una respuesta utilizando nuevamente SOAP.

4.1.4 Pruebas y resultados

A continuación se muestran las pruebas efectuadas al sistema y los resultados obtenidos. Las pruebas se llevaron a cabo con la finalidad de verificar el correcto funcionamiento de la biblioteca digital al consumir los Servicios Web implementados y además:

- Medir el tiempo de recuperación de materiales.
- Explicar el consumo dinámico de Servicios Web.
- Realizar comparativo de funcionalidades entre otras bibliotecas.

4.1.4.1 Tiempo de recuperación de materiales.

En cuanto al tiempo de recuperación de materiales las pruebas realizadas consistieron particularmente en medir el tiempo de desempeño de recuperación de materiales en base a cada tipo de consulta. Además se verificó el cumplimiento de objetivos planteados inicialmente para el módulo del cliente:

- Establecer comunicación vía Web.
- Consumo de Servicios Web y conexión con la base de datos.

Las pruebas realizadas fueron llevadas a cabo en el buscador principal del ambiente de usuario principal. Véase Figura 4.2.

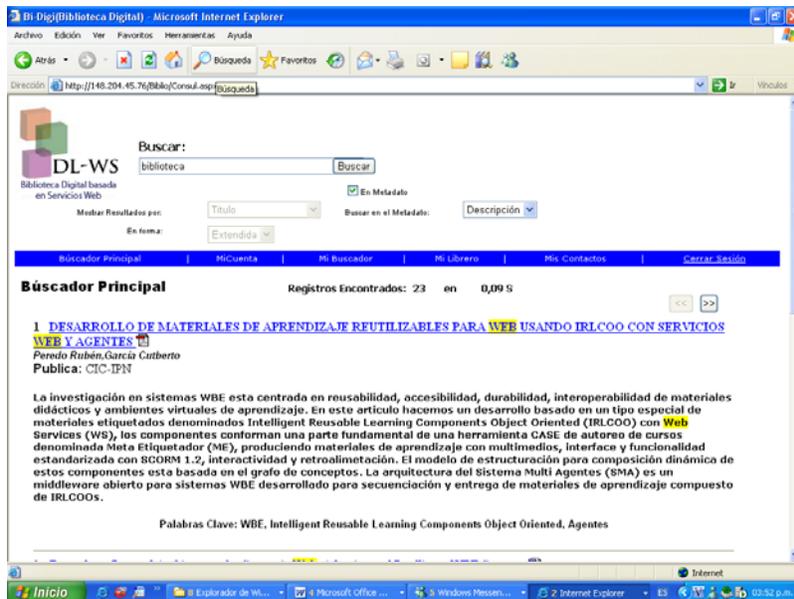


Figura 4.2 Buscador principal del ambiente de usuario principal

Para la prueba de tiempo de desempeño de recuperación de materiales, se proporcionó al sistema la palabra de búsqueda "WEB", término que fue comparado en cada uno de los metadatos de los materiales existentes en la biblioteca digital. Las búsquedas se realizaron entre un total de trescientos ochenta y tres registros almacenados en la base de datos.

Los resultados de medición obtenidos se muestran a continuación. Véase Tabla 4.1.

Comparativo de resultados de tiempo de búsqueda por tipo de consulta (búsqueda por metadato)		
Búsqueda en metadato:	Tiempo de búsqueda	Registros encontrados
Título	0,03 S	21
Autor	0,02 S	0
Palabras Claves	0,02 S	10
Descripción	0,09 S	23

Tabla 4.1 Tiempos de búsqueda por metadatos

4.1.4.2 Consumo dinámico de Servicios Web.

Como prueba de consumo dinámico de servicios *Web*, se implemento una aplicación para Windows, en donde el usuario ingresa una cadena de búsqueda e indica el campo en donde debe realizarse. Dependiendo de la opción seleccionada, es el número del método residente en el Servicio Web que se va a invocar usando el proxy dinámico. El resultado obtenido, de acuerdo al descriptor WSDL, es un objeto de tipo DataSet que alimenta al grid visual correspondiente.

En la figura 4.3 aparece un ejemplo de la aplicación Windows antes de utilizar el servicio Web correspondiente.

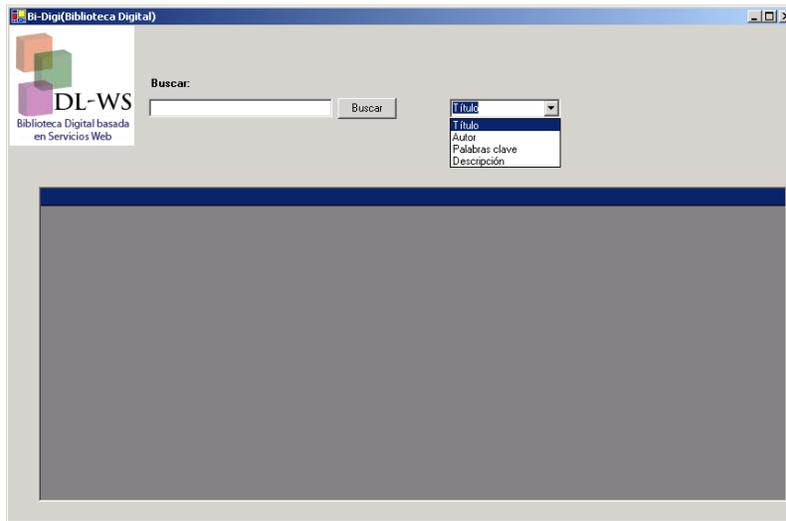


Figura 4.3 Ubicación y consumo dinámico de Servicios Web

La figura 4.4 aparece un ejemplo de la aplicación Windows después de haber llenado la tabla de datos.

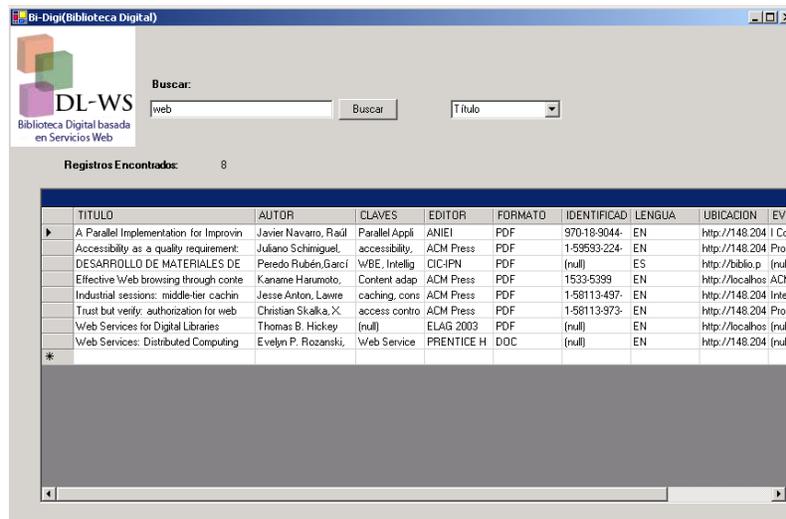


Figura 4.4 Ubicación y consumo dinámico de Servicios Web

4.1.4.2 Comparativo de funcionalidades con otras bibliotecas.

A continuación se presenta un comparativo de funcionalidades entre otros desarrollos de bibliotecas digitales. Véase Tabla 4.2.

Biblioteca Digitales Comparativo de funcionalidades				
Biblioteca Digital (BD)	Opciones de búsqueda	Ordenamiento de materiales	Materiales	Servicios
BD. CIC, IPN	-Busca solo títulos de revistas y libros por metadatos (libre, título, descriptores, issbn)	-orden alfabético	-Solo muestra información del material	-Solo de búsqueda y visualización
BD. de Tesis. U. de Colima	-Búsquedas libres y por metadatos (título, autor, asesor, grado académico, fecha). Búsquedas Asistidas (título y Autor)	-orden alfabético	-Muestra información del material y recupera recurso electrónico	-Solo de búsqueda y recuperación
BD. Tecnológico de Monterrey	-Contiene agentes asistentes de búsquedas. -Las búsquedas se realizan en diferentes bases de datos y catálogos. -Posee lenguaje de consultas definido por el usuario -Realiza búsquedas en texto y en metadatos	-Ordena por orden alfabético, por fecha y por relevancia.	-Muestra información del material y recupera recurso electrónico	-Agente de búsquedas automáticas -Espacio personal para guardar y organizar documentos
BD. Basada en servicios Web (Tesis)	-Búsquedas libres y por metadatos (título, autor, asesor, grado académico, fecha).	-Ordena por orden alfabético, título, autor y fecha	-Muestra información del material y recupera recurso electrónico	-Búsqueda y recuperación de recursos -Espacio personal para buscar, guardar y organizar documentos -Modulo de para visualizar contactos.

Tabla 4.1 Comparativo de funcionalidades

Capítulo V

Caso de Uso

En este capítulo se presenta un caso de estudio acerca del funcionamiento del sistema en una forma típica de cómo la utilizaría cualquier usuario principal o administrador. Como se mencionó anteriormente la biblioteca digital desarrollada es una aplicación que ofrece a un usuario principal, una variedad de servicios entre los que se encuentran: la búsqueda y visualización de documentos, con una gran variedad de opciones para el despliegue de información. Un usuario puede seleccionar cualquier recurso de su interés para guardarlo en un "librero personalizado", a donde el solo puede ingresar. Del mismo modo puede eliminar los materiales que un momento determinado ya no sean de su interés. Por último el usuario también tiene la posibilidad de visualizar la información de las personas que tienen asignados los mismos recursos que él, en sus respectivos libreros.

Para un usuario administrador también se contempla un caso de estudio específico de cómo utilizar el ambiente para la administración de recursos y usuarios. Los requerimientos necesarios para la instalación de la aplicación se presenta en el anexo A.

5.1 Modulo Administrador

El sistema de biblioteca digital, actualmente contiene un solo usuario administrador, previamente registrado en la base de datos, su login y password respectivamente son: admin, admin. Cuando este usuario es validado, el sistema le permite el ingreso al ambiente de administración en donde podrá realizar varias operaciones, tal y como se observa en las Figuras 5.1 y 5.2 :

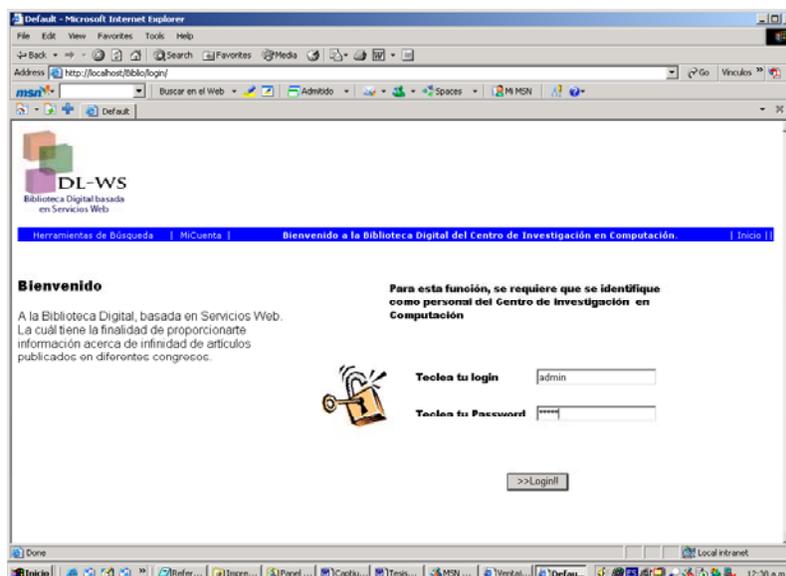


Figura 5.1 Inicio de sesión de administrador

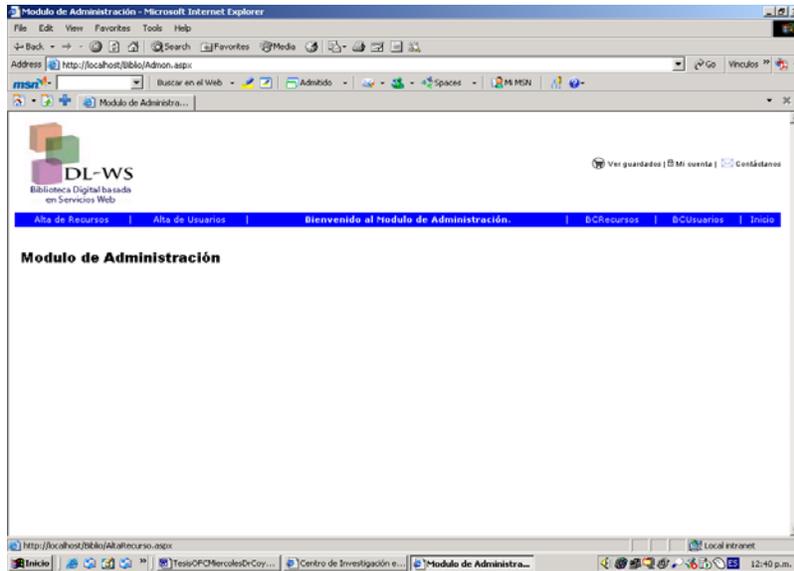


Figura 5.2 Modulo de Administración

La Fig. 5.2 muestra el ambiente de administración del usuario "admin". En esta sección el administrador podrá elegir las opciones: Alta de Usuarios, Alta de Recursos, BCRrecursos, BCUsuarios.

5.1.1 Alta de Recursos

Para ingresar a esta sección, el administrador debe de seleccionar la opción "Alta de Recursos" y hacer clic. La figura 5.3 muestra el formulario de información que el administrador debe llenar para lograr el registro del recurso.

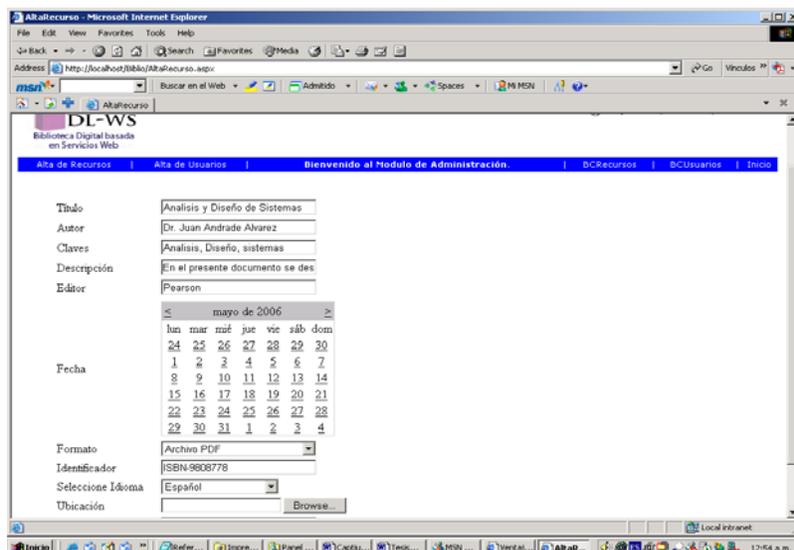


Figura 5.3 Formulario de registro de recursos o materiales

El formulario dispone de un campo "Ubicación", que registra la ubicación física del recurso en la computadora. Para lograr tal efecto el administrador debe dar clic en el botón "Browse". Finalmente para guardar el documento basta con dar clic en el botón registrar de la pantalla. Figura 5.4.

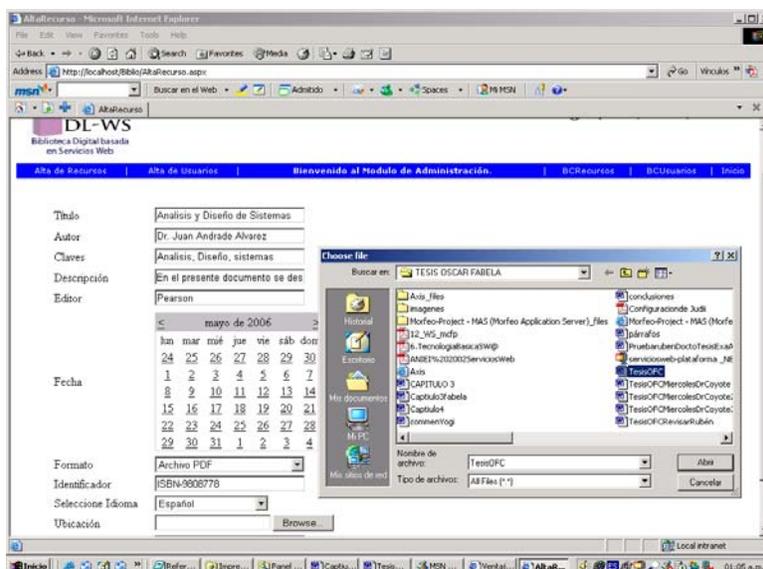


Figura 5.4 Formulario de registro (opción Ubicación)

5.1.2 Alta de Usuarios

Para registrar un nuevo usuario en el sistema, el administrador debe de elegir la opción "Alta de Usuarios", del modulo principal de administración. Ver figura. 5.2. La figura 5.5 muestra el formulario de registro de un usuario principal. Será necesario escribir los datos correspondientes que solicita el formulario y al finalizar hacer clic en el botón registrar.

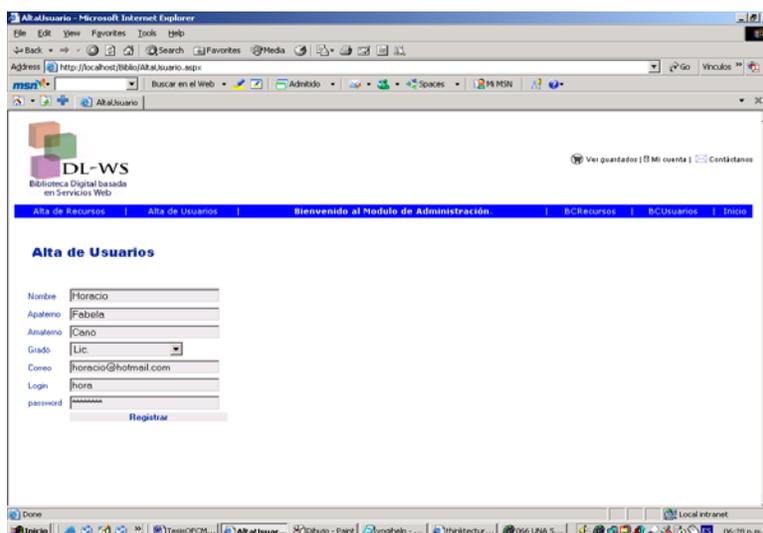


Figura 5.5 Formulario de registro del usuario principal

5.1.2 Edición, actualización , borrado de usuarios y recursos

Para ingresar a esta opción, el administrador debe seleccionar del menú principal la opción "BCUsuarios" figura 5.2. La figura 5.6 muestra la pantalla que brinda las opciones de edición, actualización y borrado de registros de usuarios. En esta sección el administrador debe de activar las ligas de hipervínculo para ejecutar la acción deseada.

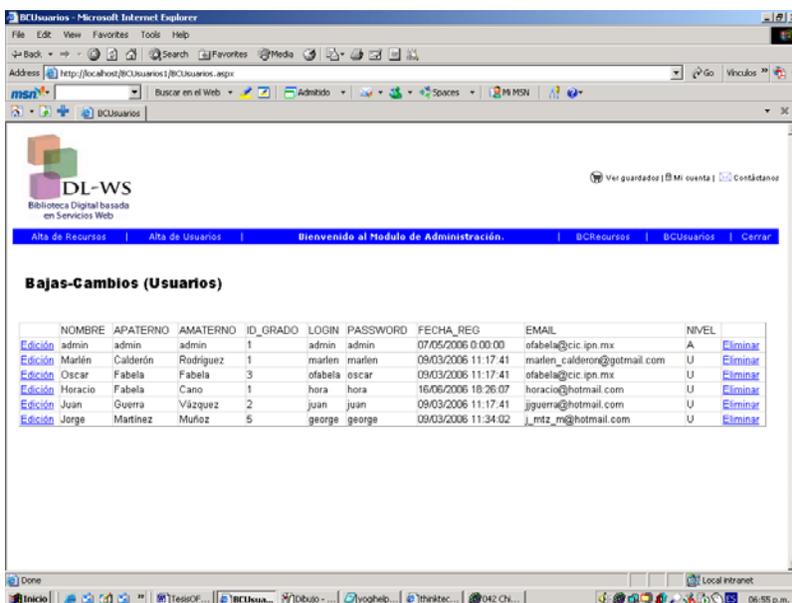


Figura 5.6 Formulario "BCUsuarios" Bajas-Cambios (usuarios)

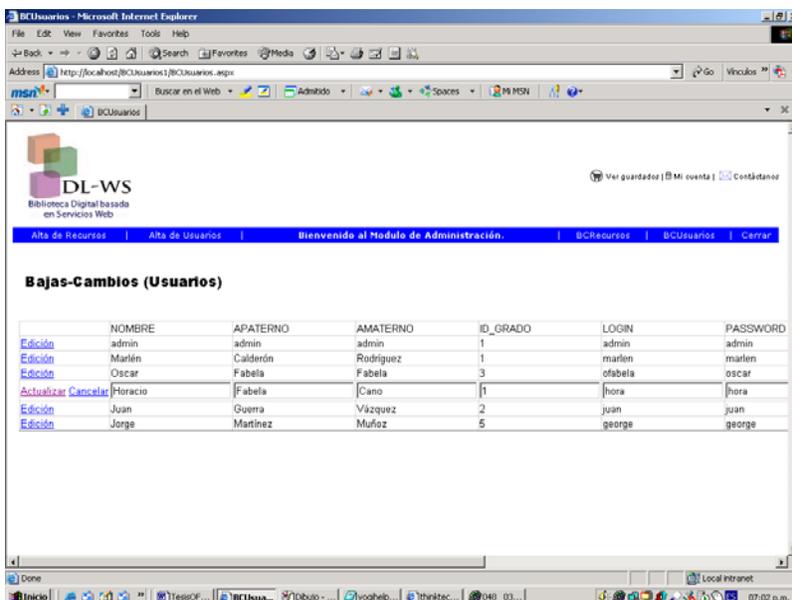


Figura 5.7 Formulario "BCUsuarios" opción actualizar activada

Para la eliminación de un registro de usuario es suficiente con que el administrador de clic el hipervínculo "Eliminar", en ese momento se recargará el formulario de tal manera que ya no se visualizará el registro borrado.

Para realizar las mismas operaciones, referentes a recurso de materiales, es necesario que el administrador seleccione del menú principal del modulo de administración la opción "BCRecursos". Figura 5.2. Los pasos a realizar para estos efectos son similares a los explicados anteriormente, por lo que no se describen las pantalla correspondientes.

5.2 Modulo Usuario principal

A continuación se describen las actividades que un usuario principal puede realizar dentro de este modulo, se explican detalladamente las formas de recuperación y visualización de materiales que puede tener este usuario.

El sistema dispone de solo una forma web de autenticación (Figura 5.2), en esta pantalla el usuario principal debe de proporcionar su login y password registrados por el administrador para poder acceder al modulo "Mi Cuenta" (Figura 5.8), el cuál es un espacio personalizado para cada usuario en el cuál se puede realizar diferentes operaciones, las cuáles se describen a continuación.

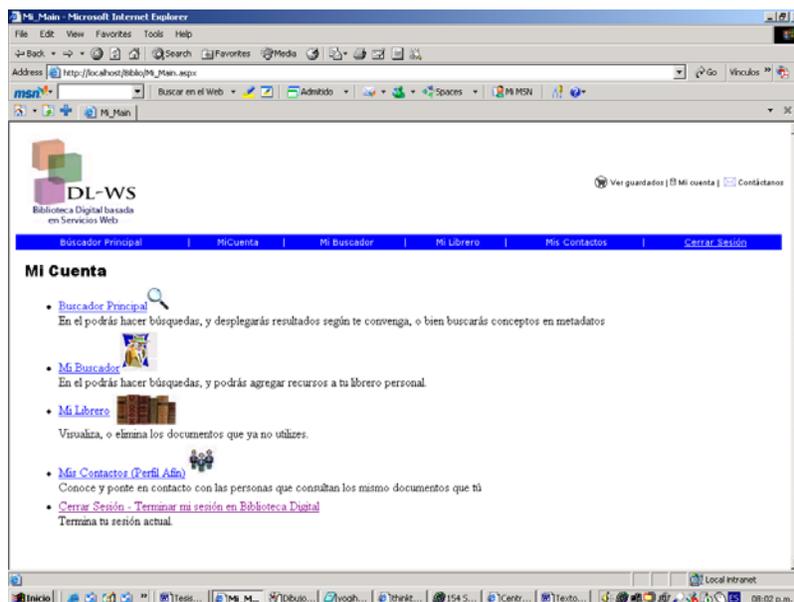


Figura 5.8 Modulo "Mi Cuenta"

Dentro de este modulo el usuario puede acceder a otros submodulos como los son: "Buscador Principal", "Mi Buscador", "Mi Librero", "Mis Contactos". Para llegar a ellos se pueden utilizar tanto el menú horizontal que está en color azul, como los hipervínculos que se encuentran en el centro de la forma web.

5.2.1 Buscador Principal

Para ingresar a este modulo es necesario hacer clic en la opción "Buscador Principal". En esta sección el usuario puede realizar búsquedas basadas en diferentes criterios, para tal efecto el usuario debe de escribir en el campo "Buscar", el término a encontrar. Figura 5.9.

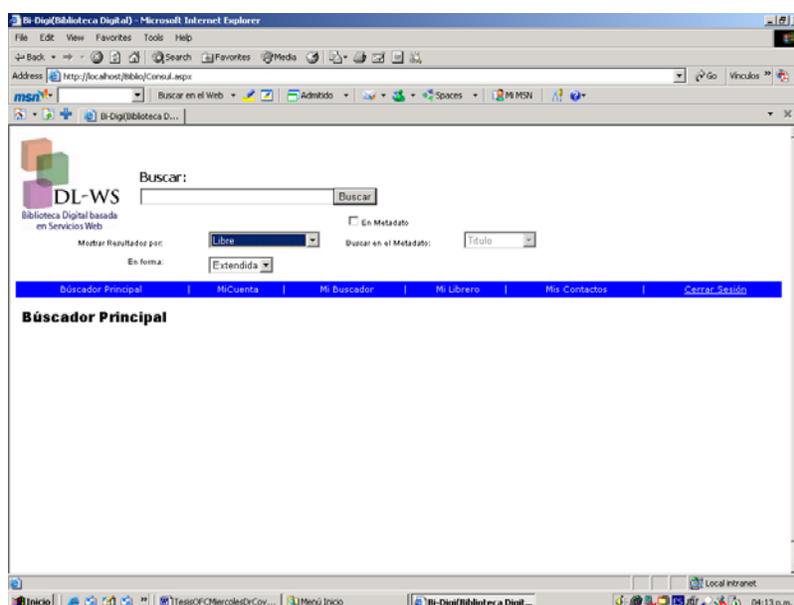


Figura 5.9 Modulo "Buscador Principal"

La opción "Mostrar Resultados por:", indica al sistema realizar una búsqueda del parámetro proporcionado en todos los meta datos registrados del recurso o material (Titulo, Autor, P.Claves, descripción) (Figura 5.10), y los resultados obtenidos son ordenados de acuerdo a la opción seleccionada:

Libre: Se realiza la búsqueda en los meta datos del material, y la lista de resultados se presenta en orden ascendente. (Según el orden en que hayan sido almacenados los registros).

Titulo: La búsqueda se realiza en los meta datos antes mencionados, y el resultado es ordenado de acuerdo al titulo del material

Autor: La búsqueda se realiza en los meta datos mencionados, y el resultado es ordenado por autor.

Fecha: Se busca en los meta datos mencionados y el ordenamiento se realiza por fecha. Los registros se muestran según la fecha de registro. Los más recientemente registrados son mostrados primero.

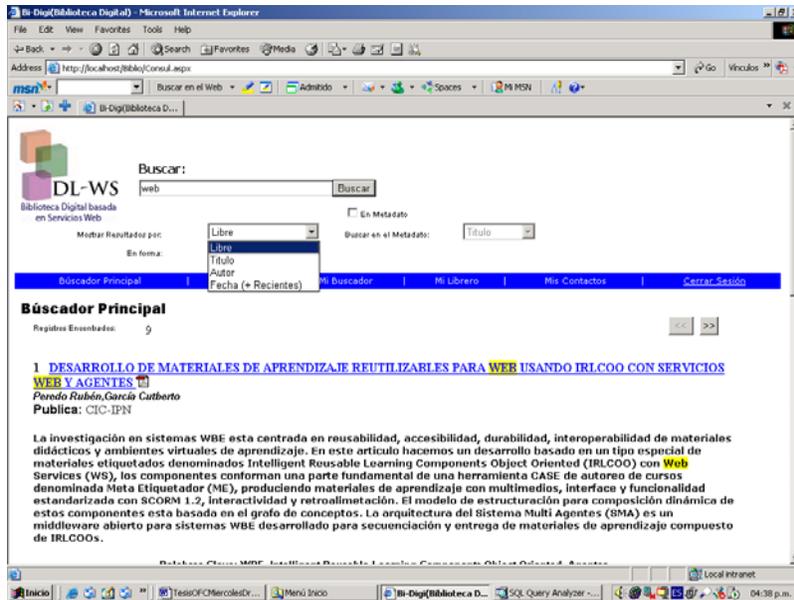


Figura 5.10 Opción "Mostrar resultados por:"

La figura 5.11 muestra activada la opción "En Metadato". Permite realizar la búsqueda y ordenamiento solo en el metadato indicado (Título, Autor, P.Claves, Descripción). Figura 5.11.

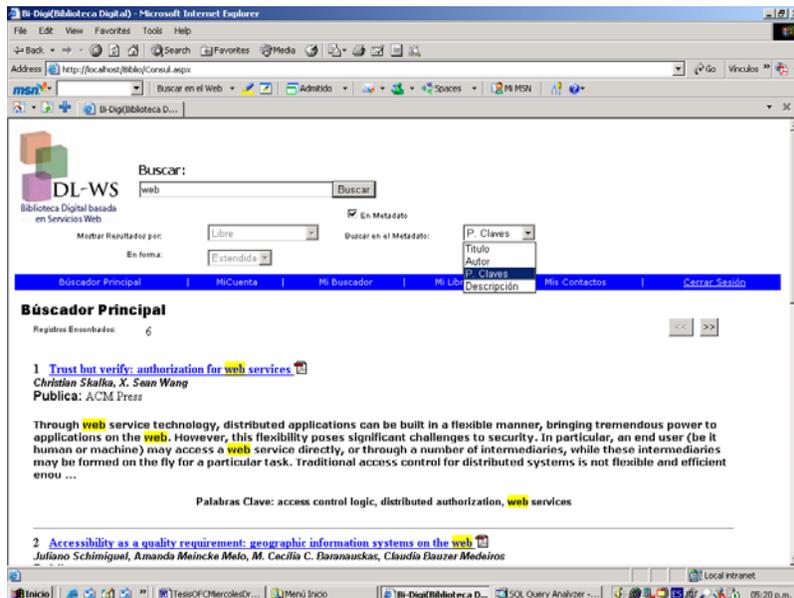


Figura 5.11 Opción "En Metadato:"

5.2.2 Mi Buscador y Mi Librero

Para ingresar es necesario hacer clic en la opción "Mi Buscador", del menú horizontal de color azul, que se encuentra en cada una de las pantallas del modulo. En esta sección el usuario podrá realizar búsquedas de la misma forma que el modulo anterior. La diferencia consiste en la forma de despliegue de la información, ya que esta es contenida en un "grid" de datos. La Figura 5.12 muestra un ejemplo del funcionamiento de este modulo cuando la cadena de búsqueda que proporciona el usuario es la palabra "web".

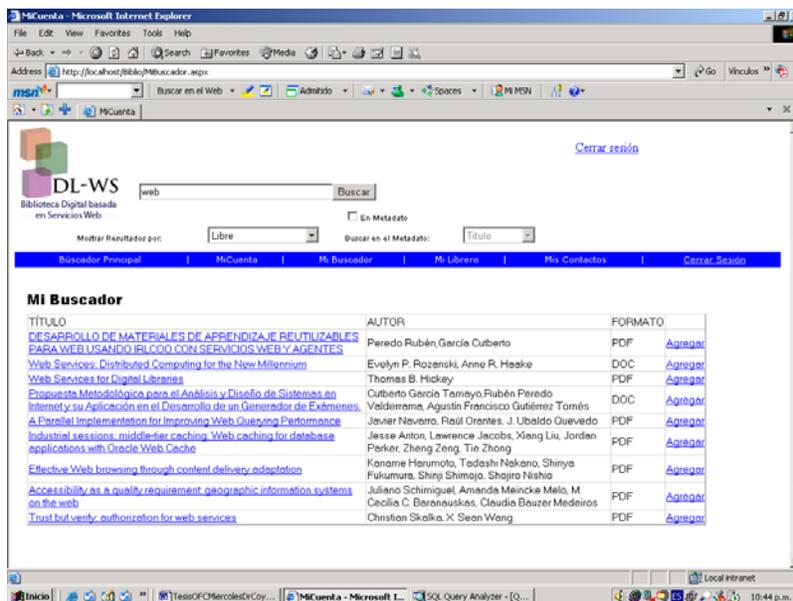


Figura 5.12 Modulo "Mi Buscador"

El "grid" de datos que muestra la información de los materiales que cumplen con la condición de búsqueda, muestra la información referente al Título, autor, y formato (Tipo de formato disponible en la biblioteca). En la columna Título, la información aparece como un hipervínculo, en donde el usuario puede hacer clic en el registro deseado y visualizar el documento físico. Figura 5.13.

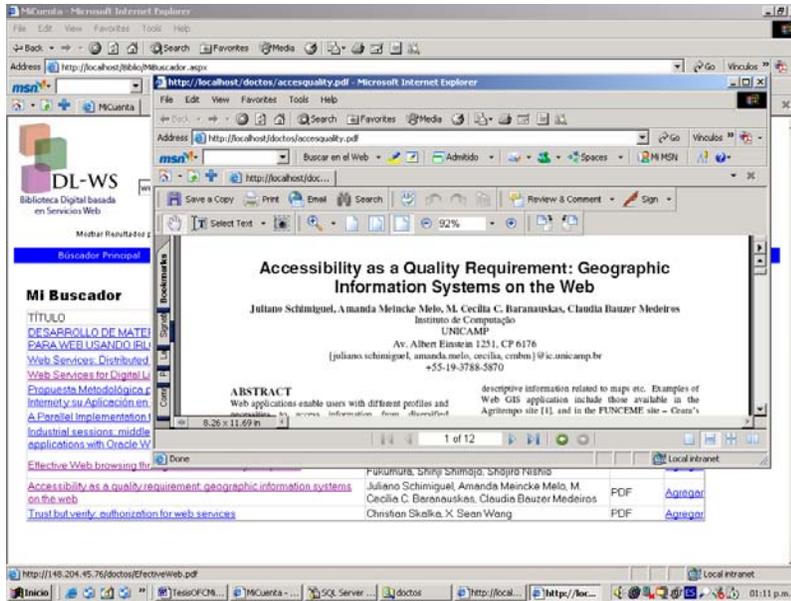


Figura 5.13 Despliegue de un recurso seleccionado

La ultima columna del "grid" de datos, contiene un hipervínculo "Agregar" para cada uno de los registros recuperados en la búsqueda. Un usuario puede guardar el recurso de su interés en un espacio personalizado, de tal forma que al cerrar sesión el documento no se perderá de su espacio personalizado, cuando este inicie de nuevo una sesión podrá dirigirse al modulo "Mi librero" y los registros previamente agregados , serán visualizados. Fig. 5.14. Cuando un usuario desea eliminar un recurso de "su librero", basta con que de clic en el hipervínculo "Eliminar", en ese momento el recurso desaparecerá de estas sección.

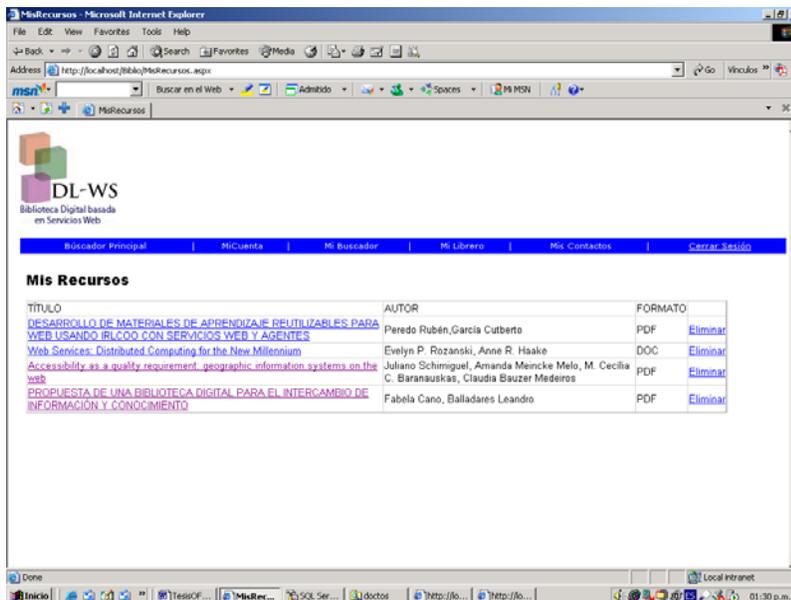


Figura 5.14 Modulo "Mi Librero" Un usuario visualiza los registros anteriormente seleccionados

5.2.3 Mis Contactos

Para ingresar a este modulo, el usuario principal debe hacer clic, en la opción "Mis Contactos" del menú principal. En esta sección el usuario visualiza una lista de usuarios de la misma biblioteca digital, que hasta en ese momento tienen agregados en sus librerías personales los mismos materiales que él. Figura. 5.15.

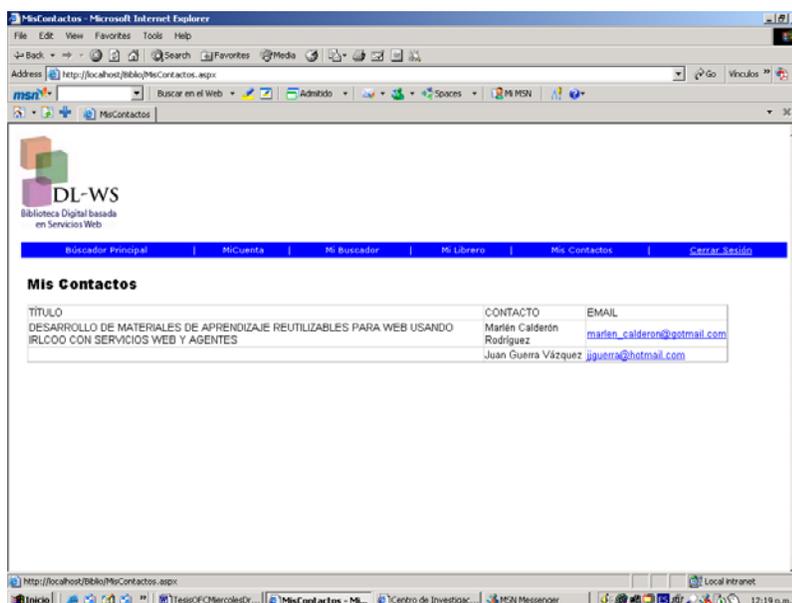


Figura 5.15 Contactos de un usuario

La información de estos contactos es desplegada en un "grid" de datos, que contiene el título del recurso o material, el nombre y correo electrónico del usuario que tiene ese material asignado a su librero personal.

Finalmente cada una de las interfaces de este modulo de "Usuario Principal", contiene un hipervínculo "Cerrar Sesión", donde al hacer clic el usuario abandonará la sesión en la que se encuentra y el sistema lo redirigirá a la pantalla inicial. Figura 5.1.

Capítulo VI

Conclusiones y Trabajos Futuros

6.1 Conclusiones

Las tecnologías de desarrollo *Web* han evolucionado de manera exponencial y en conjunto con XML, han transformado los modelos de desarrollo de la WWW actual, permitiendo interoperabilidad de sistemas basada en estándares ampliamente aceptados en la industria, permitiendo que las organizaciones integren sus aplicaciones "bussines-to.bussines" o "bussines-to-consumer" aprovechando la infraestructura de comunicaciones de Internet y las posibilidades del XML.

Con el desarrollo de este proyecto de la biblioteca digital se logra ofrecer servicios que una biblioteca tradicional proporciona pero con algunas ventajas adicionales como lo son:

- Búsquedas basadas en meta datos siguiendo el estándar de Dublin Core
- Acceso a recursos personalizados de información.
- Eficiente acceso a datos al realizar el almacenamiento y recuperación de información en un servidor de bases de datos (SQL).
- Al estar basada en WS provee una interfaz programática única que puede ser usada por cualquier desarrollador que requiera de incluir los servicios de una BD en su aplicación independientemente de la plataforma y lenguaje de programación utilizado.
- Posibilidad de implementar diversas interfaces de usuario de acuerdo al uso particular, ya que la lógica de la aplicación está implementada en los WS.

Otra de las ventajas fundamentales es que además de ofrecer el servicio de consultas tradicionales mediante SQL, ofrece la posibilidad de consultas mediante el meta etiquetado de Dublin Core. Se eligió este meta etiquetado debido a que Dublin Core es un estándar abierto de meta datos que define un conjunto de elementos básicos que permitió meta etiquetar la información de los recursos o materiales disponibles de la biblioteca digital, para así facilitar la interacción y la recuperación de los datos solicitados por los usuarios.

El sistema permite tener acceso a recursos personalizados de la biblioteca digital, y agrupar usuarios con perfiles afines, además de ofrecer los contactos correspondientes para poder generar comunidades de intercambio de conocimiento.

El desarrollo de esta aplicación nos permitió implementar y entender que los servicios *Web* desarrollados, pueden ser utilizados o bien consumidos por otro tipo de aplicaciones de biblioteca digital, donde para tal efecto es necesario registrar nuestros servicios Web en

un servidor UDDI. En este desarrollo se utilizó JUDDI que es un servidor libre basado en Java, que implementa un servidor UDDI, y donde finalmente se encuentran las referencias de los servicios *Web* utilizados, logrando con esto un mayor grado de interoperabilidad, al permitir trabajar a diferentes plataformas heterogéneas de manera conjunta. El consumo de los servicios *Web* no siempre son directamente referenciados, ya que es posible acceder de manera dinámica en tiempo de ejecución por medio de la referencia obtenida de un servidor JUDDI, permitiendo resolver solicitudes de servicios de manera flexible.

Finalmente podemos decir que los Servicios Web son aplicaciones que pueden ser:

- Descritas mediante un lenguaje de descripción de servicio, como el lenguaje WSDL (*Web Service Description Language*)
- Publicadas al someter las descripciones y políticas de uso en algún registro bien conocido, utilizando el método de registro UDDI (*Universal Description, Discovery and Integration*).
- Encontradas al enviar peticiones al registro y recibir detalles de ligamiento (*binding*) del servicio que se ajusta a los parámetros de la búsqueda.
- Asociadas al utilizar la información contenida en la descripción del servicio para crear una instancia de servicio disponible o proxy.
- Invocadas desde prácticamente cualquier plataforma y lenguaje de programación.

6.2 Resultados finales

Los resultados obtenidos del desarrollo de este proyecto son:

- Desarrollo e implementación de un conjunto de servicios *Web* que dan las características básicas de funcionamiento a una biblioteca digital.
- Desarrollo de una aplicación utilizando la tecnología de .NET que aprovecha los servicios Web implementados, y da funcionalidad a una BD.
- Configuración y puesta en marcha de un servidor de carácter público que implementa UDDI (JUDDI).
- Registro en JUDDI de los servicios *Web* implementados.
- Generación de Proxy dinámico para el consumo en línea de servicios *Web*.

6.3 Trabajos Futuros

En esta sección se presentan algunas extensiones probables que podrían realizarse a este trabajo para futuros proyectos:

- Adecuar la aplicación a las nuevas especificaciones sobre servicios *Web*, principalmente aquellas que tienen que ver con aspectos de seguridad.
- Desarrollar un sistema multiagentes que de manera inteligente asista al usuario en su búsqueda de conocimiento en la biblioteca digital.
- Desarrollar sistemas de colaboración en línea para la compartición dinámica de información.
- Desarrollar una biblioteca digital semántica.

Bibliografía

- [1] Web Services: Why and How, Francisco Curbera, William A. Nagy and Sanjiva Weerawarana IBM T.J. Watson Research Center,
<http://www.research.ibm.com/people/b/bth/OOWS2001/nagy.pdf>.
- [2] Charles f. Goldfarb, Paul Prescod; traducción: Beatriz Paredes, Alejandro Ruiz, Belen Venegas Junio 1999, MANUAL DE XML Editorial Prentice Hall.
- [3] Yasser Shohoud, Real World XML Web Services For VB and VB .NET Developers (Pearson Education, Inc. 2003).
- [4] ODBC, URL: <http://www.webopedia.com/TERM/O/ODBC.html>.
- [5] JDBC, URL: <http://java.sun.com/products/jdbc/>.
- [6] ADO, URL <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ado270/htm/dasdkadooverview.asp>
- [7] Using WSDL in SOAP applications, Extensible Markup Language (XML) 1.0 (Third Edition), URL: <http://www.w3.org/TR/REC-xml/>.
- [8] Aaron E Walsh Abril 2002, UDDI, SOAP, AND WSDL: THE WEB SERVICES SPECIFICATION REFERENCE BOOK Editorial Prentice Hall.
- [9] Protocolo Z39.50, URL: <http://www.bcl.jcyl.es/zeta/>
- [10] Especificación Dublin Core, URL: <http://www.rediris.es/metadata>
- [10] Normas Marc, Biblioteca del Congreso. <http://www.log.gov/marc/marcspa.html>
- [11] Pasquelli Art (2002). Digital Library Technology Trends. Sun Microsystems.
- [12] Digital Library Federation. URL:<http://www.diglib.org>
- [13] Borgman CL. Fourth DELOS Workshop. Evaluation of Digital Libraries: Testbeds, Measurements, and Metrics. Disponible en URL:
http://www.dli2.nsf.gov/internationalprojects/working_group_reports/evaluation.html
- [14] Pasquelli Art (2002). Digital Library Technology Trends. Sun Microsystems.
- [15] Garza-Salazar, et al., "Proyecto Phronesis, Tecnología para la Creación de Bibliotecas Digitales en Internet", ITESM-Campus Monterrey, Monterrey, NL, México, 2000.
- [16] WWW, URL: <http://www.w3.org>
- [17] C. Lagoze and J. R. Davis, "Dienst: An Architecture for Distributed Documents Libraries", Communications of the ACM, vol. 38, pp. 47, 1995.
- [18] France, Robert K., Ben E. Cline and Edward A. Fox. "MARIAN Detailed Design". Unpublished technical document, Virginia Tech Computing Center. v. 1.4, 1992.
- [19] E. Fox, R. France, E. Sahle, A. Daoud, and B. Cline, "Development of a Modern OPAC: From REVTOLC to MARIAN", in Proc. 16th Annual Int'l ACM SIGIR Conf. On R&D in Information Retrieval, SIGIR '93. Pittsburgh: ACM Press, 1993, pp. 248-259.
- [20] García Molina, et al., "Interoperability for Digital Libraries: Problems and Directions". Stanford University, 1998.
- [21] Aaron E Walsh Abril 2002, UDDI, SOAP, AND WSDL: THE WEB SERVICES SPECIFICATION REFERENCE BOOK Editorial Prentice Hall.

- [22] Brian E. Travis Agosto 2000, XML AND SOAP PROGRAMMING Editorial Microsoft Press.
- [23] Definición de Servicio Web, URL: http://Teoría WebServices/serviciosweb-plataforma_NET.htm
- [24] Web Services: Why and How, Francisco Curbera, William A. Nagy and Sanjiva Weerawarana IBM T.J. Watson Research Center, <http://www.research.ibm.com/people/b/bth/OOWS2001/nagy.pdf>.
- [25] Building Reliable Web Services Compositions, Paulo F. Pires, Mário R. F. Benevides, and Marta Mattoso, <http://www.netobjectdays.org/pdf/02/papers/ws-rsd/1251.pdf>.
- [26] Java Web Services, URL: <http://java.sun.com/webservices/>
- [27] JUDDI, Apache Web Services Project URL: <http://ws.apache.org/juddi/>
- [28] XML-RPC, URL: <http://www.userland.com>
- [29] UDDI, URL:<http://www.uddi.org/>
- [30] Apache Org. URL: <http://httpd.apache.org/docs/2.0/es/>
- [31] Search and Retrieve on the *Web*
URL:<http://www.oclc.org/research/projects/webservices/default.htm>
- [32] Meta Información Dublin Core, Elementos del Conjunto de metadatos de Dublín Core: descripción de referencia. URL: <http://www.rediris.es/metadata/>
- [33] Microsoft .NET, <http://www.microsoft.com/net/technical/>.
- [34] Sun Java, Enterprise System,
URL:<http://www.sun.com/software/javaenterprisesystem/>
- [35] Sun, URL: <http://www.sun.com>
- [36] Webspher URL: <http://www-306.ibm.com/software/websphere/>
- [37] Borland Web Services Kit for Java URL:
http://info.borland.com/04/jbuilder/jb6_webservices_preview_down.html
- [38] Axis URL: <http://ws.apache.org/axis/>.
- [39] JAX-RPC URL: <http://java.sun.com/webservices/jaxrpc/>
- [40] Apache Tomcat, URL: <http://tomcat.apache.org/>
- [41] Multi-threads URL: <http://webdia.cem.itesm.mx/ac/rogomez/Apuntes/artiThreads.pdf>
- [42] Biblioteca Encarta, URL: <http://es.encarta.msn.com/>
- [43] JSP, URL: <http://java.sun.com/products/jsp/>
- [44] Servlets, URL: <http://java.sun.com/products/servlet/>
- [45] PHP, URL: <http://www.php.net/>
- [46] Google, URL: <http://www.google.com.mx>
- [47] .NET Framework URL:
<http://www.microsoft.com/spanish/msdn/arquitectura/das/distapp.asp>
- [48] Java Web Services, <http://java.sun.com/webservices/>
- [49] Librería .NET para consumo dinámico de Servicios Web
<http://www.thinktecture.com/Resources/Software/DynWsLib/default.html>
- [50] Arquitectura Servicio Web, URL:
http://glud.udistrital.edu.co/glud/areas/doc/articulos/1_articulo_ws/servicios-web.html

Anexo A

Autenticación

Archivo de configuración autenticación por formularios:

A continuación se da una breve descripción de los pasos a seguir para modificar dicho archivo:

1.- Abrir el archivo Web.config

2.- Crear dentro del elemento system.web, un elemento authentication y establecer su atributo mode en Forms.

```
<system.web> </authentication> </system.web>
```

3.- Dentro del elemento authentication, es necesario crear un elemento forms y establecer lo siguiente:

loginUrl: Establecido a "Default.aspx", que es la dirección URL que se utiliza para el redireccionamiento si ASP. NET no encuentra un resultado verdadero de autenticación con la solicitud.

Name: Establecido en ".ASPXFORMSAUTH". De este modo se establece el sufijo del nombre de la solicitud aceptada (cookie), que contiene el vale de autenticación.

De la siguiente forma:

```
<system.web><forms loginUrl="Default.aspx" name=".ASPXFORMSAUTH">  
</forms></authentication></system.web>
```

4.- Dentro del elemento authorization, debemos crear un elemento deny y establecer su atributo users en "?". Esto para limitar el acceso solamente a los usuarios autenticados.

```
<system.web><forms loginUrl="Default.aspx" name=".ASPXFORMSAUTH">  
<forms><authorization><deny users="?" /></authorization></system.web>
```

Anexo B

Forma Web (aspx)

Consul.aspx

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
    using System.Drawing;
using System.Web.Security;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

namespace Biblio
{
    public class Consul : System.Web.UI.Page
    {
        protected DataSet dsDatos;
        protected System.Web.UI.WebControls.DropDownList List;
        protected System.Web.UI.WebControls.Label Label3;
        protected System.Web.UI.WebControls.Button Button1;
        protected System.Web.UI.WebControls.TextBox TextBox1;
        protected System.Web.UI.WebControls.Label Label4;
        protected System.Web.UI.WebControls.Label Label5;
        protected System.Web.UI.WebControls.Label Label6;

        protected int intRegsPorPag = 5;
        protected System.Web.UI.WebControls.Button btnRegresar;
        protected System.Web.UI.WebControls.Button btnAvanzar;
        protected System.Web.UI.WebControls.Label Label1;
        protected System.Web.UI.WebControls.DropDownList.ListaDespliegue;
        protected System.Web.UI.WebControls.Image Image1;
        protected System.Web.UI.WebControls.Label Label2;
        protected System.Web.UI.WebControls.DropDownList ListMeta;
        protected System.Web.UI.WebControls.CheckBox busca;
        protected System.Web.UI.WebControls.Label lblMeta;
        protected System.Web.UI.WebControls.Label Label7;
        protected System.Web.UI.WebControls.LinkButton linkCerrar;
        protected System.Web.UI.WebControls.Label lblPagActual;

        private void Page_Load(object sender, System.EventArgs e)
        {

            // Introducir aquí el código de usuario para inicializar la página

        }

        #region Código generado por el Diseñador de Web Forms
        override protected void OnInit(EventArgs e)
        {

            InitializeComponent();
            base.OnInit(e);
        }
    }
}
```

```

    /// <summary>
    /// Método necesario para admitir el Diseñador. No se puede modificar
    /// el contenido del método con el editor de código.
    /// </summary>
    private void InitializeComponent()
    {
this.Button1.Click += new System.EventHandler(this.Button1_Click);
this.List.SelectedIndexChanged += new System.EventHandler(this.Button1_Click);
this.btnAvanzar.Click += new System.EventHandler(this.Button1_Click);
this.btnRegresar.Click += new System.EventHandler(this.Button1_Click);
this.ListaDespliegue.SelectedIndexChanged += new System.EventHandler(this.Button1_Click);
this.ListMeta.SelectedIndexChanged += new System.EventHandler(this.Button1_Click);
this.busca.CheckedChanged += new System.EventHandler(this.busca_CheckedChanged);
this.linkCerrar.Click += new System.EventHandler(this.linkCerrar_Click);
this.Load += new System.EventHandler(this.Page_Load);

    }
#endregion

private void Button1_Click(object sender, System.EventArgs e)
{
    string strSalida = "";

    int intPagActual = Convert.ToInt32(lblPagActual.Text);
    if(sender == Button1)
    {
        lblPagActual.Text = "1";
    }

    else if(sender == btnAvanzar)
        lblPagActual.Text = (intPagActual + 1) + "";
    else if(sender == btnRegresar)
        lblPagActual.Text = (intPagActual - 1) + "";

    int i,z;
    if(!(sender == ListMeta) && !busca.Checked)
    {
        if (List.SelectedIndex == 0)
        {
            Biblio.Service1 WS = new Biblio.Service1();

            dsDatos = WS.WSLib(TextBox1.Text);
        }
        else if(List.SelectedIndex == 1)
        {
            Biblio.SrvTitulo WS = new Biblio.SrvTitulo();
            dsDatos = WS.WSTitulo(TextBox1.Text);
        }
        else if(List.SelectedIndex == 2)
        {
            Biblio.SrvAutor WS = new Biblio.SrvAutor();
            dsDatos = WS.WSAutor(TextBox1.Text);
        }
        else
        {
            Biblio.SrvFecha WS = new Biblio.SrvFecha();
            dsDatos = WS.WSFecha(TextBox1.Text);
        }
    }
    else
    {
        Biblio.SrvBusMetadatos WS = new Biblio.SrvBusMetadatos();
        if (this.ListMeta.SelectedIndex == 0)
        {
            dsDatos = WS.BusTitulo(TextBox1.Text);
        }
        else if(this.ListMeta.SelectedIndex == 1)

```

```

        {
            dsDatos = WS.BusAutor(TextBox1.Text);
        }
        else if(this.ListMeta.SelectedIndex == 2)
        {
            dsDatos = WS.BusClaves(TextBox1.Text);
        }
        else
        {
            dsDatos = WS.BusDescrip(TextBox1.Text);
        }
    }

    z = dsDatos.Tables[0].Rows.Count;

    Label4.Text = "";
    Label5.Text = "";

    if (z == 0)
    {
        Label6.Text = "Registros Encontrados:";
        Label4.Text = z.ToString();
        btnAvanzar.Visible = btnRegresar.Visible = false;
    }
    else
    {
        Label6.Text = "Registros Encontrados:";
        Label4.Text = z.ToString();

        intPagActual = Convert.ToInt32(lblPagActual.Text);
        int intEmpezarEn = (intPagActual-1)*intRegsPorPag;
        int intTope = intEmpezarEn + intRegsPorPag;
        if(intTope > z)
            intTope = z;

        for(i = intEmpezarEn; i < intTope; i++)
        {

            string sTitulo, sAutor, sEditor, sClaves, sDes , sFormat, sUbicaAr;

```

```

sTitulo = dsDatos.Tables["RECURSO"].Rows[i]["TITULO"].ToString();
sAutor = dsDatos.Tables["RECURSO"].Rows[i]["AUTOR"].ToString();
sEditor = dsDatos.Tables["RECURSO"].Rows[i]["EDITOR"].ToString();
sClaves = dsDatos.Tables["RECURSO"].Rows[i]["CLAVES"].ToString();
sDes = dsDatos.Tables["RECURSO"].Rows[i]["DESCRIPCION"].ToString();
sFormat = dsDatos.Tables["RECURSO"].Rows[i]["FORMATO"].ToString();
sUbicaAr =dsDatos.Tables["RECURSO"].Rows[i]["UBICACION"].ToString();

```

```

string dest = "<span style=\"background-color: #FFFF00\">"+TextBox1.Text+"</span>";
string correct = ReplaceEx(sTitulo, TextBox1.Text, dest);

```

```

        if(sFormat == "PDF")
        {

```

```

strSalida += " <strong> "+ (i+1) + " </strong>&nbsp;&nbsp;<b> <A HREF = \"" + sUbicaAr + "\" target = \"_blank\" > "+
correct +" </A><img src=\"imag/pdf_logo.gif\" width=\"16\" height=\"16\" alt=\"\" border=\"0\" align=\"texttop\"></b>
<br>";

```

```

        }
        else
        {

```

```

strSalida += " <strong> "+ (i+1) + " </strong>&nbsp;&nbsp;<b> <A HREF = \"" + sUbicaAr + "\" target = \"_blank\" > "+
correct +" </A><img src=\"imag/Logo_Word.gif\" width=\"16\" height=\"16\" alt=\"\" border=\"0\"
align=\"texttop\"></b> <br>";

```

```

    }

    strSalida += "<b><small><em><font face=\"Arial\">" + ReplaceEx(sAutor, TextBox1.Text, dest) + "
</font></em></small></b><br>";
    strSalida += "<b><strong><font face=\"Arial\">Publica:</font></strong></b> " + sEditor + "<br>";
    strSalida += "<br>";

    if (ListaDespliegue.SelectedIndex == 0)
    {

    strSalida += "<b><font size=\"2\" face=\"verdana\">" + ReplaceEx(sDes, TextBox1.Text, dest) + " </font></b><br>";
    strSalida += "<br>";

    if (sClaves != "")

    strSalida += "<b><center><small><font face=\"Arial\">Palabras Clave: " + ReplaceEx(sClaves, TextBox1.Text, dest) + "
</font></small></center></b><br>";

    else
    {
    strSalida += "<b><center><small><small><font face=\"Arial\"> " + sClaves + "
</font></small></small></center></b><br>";

    }

    }

    strSalida += "<hr>";

}

float f = (float)z/(float)intRegsPorPag;
int intMaxPags = Convert.ToInt32(Math.Ceiling(f));

if (intMaxPags > 1)
    btnAvanzar.Visible = btnRegresar.Visible = true;
else
    btnAvanzar.Visible = btnRegresar.Visible = false;

if (intPagActual == 1)
    btnRegresar.Enabled = false;
else
    btnRegresar.Enabled = true;

if (intPagActual == intMaxPags)
    btnAvanzar.Enabled = false;
else
    btnAvanzar.Enabled = true;

}
Label5.Text = strSalida;

}

private static string ReplaceEx(string original,
string pattern, string replacement)
{
int count, position0, position1;
count = position0 = position1 = 0;
string upperString = original.ToUpper();
string upperPattern = pattern.ToUpper();
int inc = (original.Length/pattern.Length) *
(replacement.Length-pattern.Length);
char [] chars = new char[original.Length + Math.Max(0, inc)];
while( (position1 = upperString.IndexOf(upperPattern,
position0)) != -1 )
{

```

```

        for ( int i=position0 ; i < position1 ; ++i )
            chars[count++] = original[i];
        replacement = "<span style=\"background-color: #FFFF00\">" + original.Substring(position1, pattern.Length) + "</span>";
        for ( int i=0 ; i < replacement.Length ; ++i )
            chars[count++] = replacement[i];
        position0 = position1 + pattern.Length;
    }
    if ( position0 == 0 ) return original;
    for ( int i=position0 ; i < original.Length ; ++i )
        chars[count++] = original[i];
    return new string(chars, 0, count);
}

```

```

private void busca_CheckedChanged(object sender, System.EventArgs e)
{
    if(busca.Checked)
    {
        this.List.Enabled = this.ListaDespliegue.Enabled = false;
        this.ListMeta.Enabled = true;
    }
    else
    {
        this.List.Enabled = this.ListaDespliegue.Enabled = true;
        this.ListMeta.Enabled = false;
    }
}

private void linkCerrar_Click(object sender, System.EventArgs e)
{
    FormsAuthentication.SignOut();
    Response.Redirect("/Biblio/login/Default.aspx");
}

}
}

```

Anexo C

Servicio Web (asmx)

Servicio Web: SrvBusMetadatos.

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Web;
using System.Web.Services;

namespace Biblio
{
    /// <summary>
    /// Descripción breve de SrvBusMetadatos.
    /// </summary>
    public class SrvBusMetadatos : System.Web.Services.WebService
    {
        public SrvBusMetadatos()
        {
            //CODEGEN: llamada necesaria para el Diseñador de servicios Web ASP .NET
            InitializeComponent();
        }

        private DataSet ds = new DataSet();
        private System.Data.SqlClient.SqlDataAdapter sqlDataAdapter1;
        private System.Data.SqlClient.SqlConnection sqlConnection1;
        private System.Data.SqlClient.SqlDataAdapter sqlDataAdapter2;
        private System.Data.SqlClient.SqlDataAdapter sqlDataAdapter3;
        private System.Data.SqlClient.SqlDataAdapter sqlDataAdapter4;
        private System.Data.SqlClient.SqlCommand sqlSelectCommand1;
        private System.Data.SqlClient.SqlCommand sqlInsertCommand1;
        private System.Data.SqlClient.SqlCommand sqlUpdateCommand1;
        private System.Data.SqlClient.SqlCommand sqlDeleteCommand1;
        private System.Data.SqlClient.SqlCommand sqlSelectCommand2;
        private System.Data.SqlClient.SqlCommand sqlInsertCommand2;
        private System.Data.SqlClient.SqlCommand sqlUpdateCommand2;
        private System.Data.SqlClient.SqlCommand sqlDeleteCommand2;
        private System.Data.SqlClient.SqlCommand sqlSelectCommand3;
        private System.Data.SqlClient.SqlCommand sqlInsertCommand3;
        private System.Data.SqlClient.SqlCommand sqlUpdateCommand3;
        private System.Data.SqlClient.SqlCommand sqlDeleteCommand3;
        private System.Data.SqlClient.SqlCommand sqlSelectCommand4;
        private System.Data.SqlClient.SqlCommand sqlInsertCommand4;
        private System.Data.SqlClient.SqlCommand sqlUpdateCommand4;
        private System.Data.SqlClient.SqlCommand sqlDeleteCommand4;

        #region Código generado por el Diseñador de componentes

        // EJEMPLO DE SERVICIO WEB
        // Para generar, quite la marca de comentario de las siguientes líneas y, después, guarde y genere el proyecto
        // Para probar el servicio Web, presione F5
    }
}
```

```

[WebMethod]
public DataSet BusTitulo(string param)
{
    sqlSelectCommand1.Parameters["@param"].Value = "%" + param + "%";
    sqlDataAdapter1.Fill(ds);
    return ds;
}

[WebMethod]
public DataSet BusAutor(string param)
{
    sqlSelectCommand2.Parameters["@param"].Value = "%" + param + "%";
    sqlDataAdapter2.Fill(ds);
    return ds;
}

[WebMethod]
public DataSet BusClaves(string param)
{
    sqlSelectCommand3.Parameters["@param"].Value = "%" + param + "%";
    sqlDataAdapter3.Fill(ds);
    return ds;
}

[WebMethod]
public DataSet BusDescrip(string param)
{
    sqlSelectCommand4.Parameters["@param"].Value = "%" + param + "%";
    sqlDataAdapter4.Fill(ds);
    return ds;
}
}
}
}

```

Anexo D

Requerimientos

Para la ejecución de la aplicación se necesitan los siguiente requerimientos mínimos:

Hardware y Software.

Componente	Requisitos
.NET Framework	<ul style="list-style-type: none">Compatible con todos los sistemas operativos Windows. <p>Nota: En todos estos sistemas, también es necesario tener Microsoft® Internet Explorer 5.01 o posterior y Microsoft® Windows® Installer 2.0 o posterior.</p>
Servidor de Internet (IIS)	<ul style="list-style-type: none">Servicios de Internet Information Server (IIS) en el servidor de Windows 2000, Windows XP (Professional) y Windows Server 2003. Esto es lo que se precisa para utilizar aplicaciones ASP.NET.
Cliente (Windows Forms y servicios de Windows)	<ul style="list-style-type: none">Pentium a 90 MHz o más rápido (recomendado)Ram 96 MB o más (recomendado)
Servidor	<ul style="list-style-type: none">Pentium a 133 MHz o más rápido (recomendado)Ram 256 MB o más (recomendado)