

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 1**



Centro de Informatización Universitaria

**Título:** *Solución de Software para la integración funcional del Sistema de Gestión Bibliotecaria basada en Servicios Web.*

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS.**

**Autor:**

Alberto Blanco Mustelier

Tutor(es): Ing. Dargel Veloz Morales

Cotutor(es): Ing. Edisnel Carrazana Castro

**Ciudad de La Habana. Curso 2011-2012**



*Hubiera yo desmayado, si no estuviera seguro de ver la bondad  
de Adonai en la tierra de los vivos.*

*Espera en Adonai, ten valor y firme corazón, espera en  
Adonai.*

*Rey David (Salmos 27:13-14)*

## **DECLARACIÓN DE AUTORÍA**

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Firma del Autor  
Alberto Blanco Mustelier

---

Firma del Tutor  
Ing. Dargel Veloz Morales

---

Firma del Tutor  
Ing. Edisnel Carrazana Castro

## **DEDICATORIA**

A mis padres, esos que lo dieron todo para que yo estuviera aquí, que se entregaron en alma y cuerpo para que yo lograra lo que la vida les robó.

A Maida por hacer de esto un sueño.

A Darleici por trazarme las metas y enseñarme a vencer las barreras.

A mi familia, esos que creyeron que podría dar más y obtener más aún cuando yo pensaba en rendirme.

A los que como yo, alguna vez o con frecuencia piensan que no pueden, que ya es suficiente, que no tienen fuerzas; cuando me sentí así, Dios estuvo ahí, y ahora, con él, yo estoy aquí.

## AGRADECIMIENTOS

A mis padres por cada una de las horas invertidas en mí.

A mi madre por ser mi maestra, por enseñarme que **toda gran recompensa requiere grandes sacrificios**, por haberse gastado por mí; espero un día ser la recompensa a tu sacrificio.

A mi padre por la confianza que puso en mí y por animarme a esforzarme y seguir adelante.

A mis abuelas, a Nidia por convertirme en un hombre de bien, por protegerme y amarme con pasión y devoción.

A mis tíos por estar siempre dispuestos, a Darleici por retarme con cada paso que daba en su vida y más que un tío ser el hermano que no tuve, a Maida por recibirme como su hijo y convertirse en mi segunda madre.

A mis primos, que siempre han estado ahí, que han pasado conmigo las dulces y las amargas, me han soportado y enseñado.

A mis hermanos, esos que Dios me regaló y que se han convertido en mi familia, por todos los momentos que me permitieron vivir junto a ellos, por todo lo que me enseñaron y por los recuerdos que llevo grabados para siempre en mi mente y corazón.

A mis tutores, por su ayuda, por guiarme y por ser parte de esa familia especial, por haber sido un ejemplo para mí.

A mis compañeros de grupo, los que me soportaron mis imperfecciones y lucharon a mi lado, a ese equipo maravilloso que tanto me ayudó a avanzar.

A mis profesores, los muy buenos y los que no tanto, los que me inspiraron a seguir y los que me obligaron a seguir, a los que me pusieron obstáculos y los que me enseñaron a vencerlos.

A mis amigos, los buenos y los más buenos, sin necesidad de dar sus nombres ellos saben cuánto les quiero, gracias por estar ahí cuando lo necesite, gracias por sus muchas palabras y por su silencio, sus regaños y abrazos, gracias por mostrarme que a pesar de mi perfección me equivoco bastante.

A **mi amigo**, el **mejor** de todos, al que me enseñó a hablar, a andar, a vivir, a correr, **a mi SEÑOR**, mi **DIOS**, mi PASTOR, mi HACEDOR, mi TODO, A **JESÚS**; gracias por amarme, por salvarme, por cuidarme en este lugar, gracias por estar ahí, gracias por traerme aquí, gracias por todo lo que me has dado, Gracias por hacerme ver tu bondad en esta tierra; Gracias porque cuando yo decía no puedo, no quiero, tu decías:"sigue adelante YO ESTOY CONTIGO"; GRACIAS, GRACIAS, **GRACIAS DIOS**, GRACIAS POR TODO LO QUE HAS HECHO EN MI, **GRACIAS POR TODO LO QUE SOY!!!!!!!!!!**

## Resumen

Con la aparición del software libre y el auge que ha alcanzado el mismo en condiciones como las de nuestro país, muchas bibliotecas han decidido migrar todos sus sistemas informáticos a Sistemas Operativos Libres. En la Universidad de las Ciencias Informáticas se cuenta con varios sistemas de que apoyan los procesos realizados en su biblioteca almacenando la información que se maneja en la misma; un portal que ofrece información referente a esta área, además de la utilización de un SIGB basado en Koha para gestionar dichos procesos bibliotecarios y un repositorio que almacena la producción científica de la institución, son los componentes de la solución de software empleada. Cada uno de estos sistemas tiene funciones específicas y actúa de manera independiente sin que exista una relación funcional entre los mismos. Constituye entonces objetivo fundamental del presente trabajo, elaborar un análisis que concluya con una propuesta de servicios web que permita la integración de los sistemas que forman parte de la actual Solución de Software para Bibliotecas utilizada en la universidad. Para ello se realizó un análisis de los tres sistemas atendiendo a sus principales características y funcionalidades, así como la importancia que se le otorga dentro de la solución. Atendiendo a las necesidades actuales para la integración, se construye un conjunto de servicios web que actúan como un vínculo funcional entre estos sistemas, de manera que puedan compartir información y abre las puertas al trabajo común. Las tendencias actuales en la web hacia una Arquitectura Orientadas a Servicios y la implementación de este módulo dentro del Sistema Integrado de Gestión Bibliotecaria (SIGB) utilizado en la universidad prometen una integración universal de estas aplicaciones con los sistemas del entorno como la intranet, el Portal del Graduado, Portal FEU y otros.

**Palabras Claves:** Arquitectura de Software, biblioteca, integración, Koha, perl, SIGB, Servicios Web, SOA.

## Tabla de contenido

<b>TABLA DE CONTENIDO</b> .....	<b>7</b>
<b>ÍNDICE DE IMÁGENES</b> .....	<b>9</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>9</b>
<b>INTRODUCCIÓN</b> .....	<b>10</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>14</b>
1.1 SISTEMA DE GESTIÓN BIBLIOTECARIA.....	14
1.1.1 Repositorio Institucional.....	14
1.1.2 Sistema Integrado de Gestión Bibliotecaria. ....	16
1.1.3 Sistema de Gestión para la Dirección de Información. ....	17
1.2 OPCIONES DE INTEGRACIÓN.....	17
1.3 PATRONES DE INTEGRACIÓN. ....	18
1.3.1 Entity Aggregation (Agregación de Entidades) .....	18
1.3.2 Process Integration (Integración de Procesos) .....	19
1.3.3 Portal Integration (Portal de Integración) .....	20
1.3.4 Message Broken (Intermediario de mensajes) .....	20
1.3.5 Publish/Subscribe (Publicación/Suscripción).....	20
1.3.6 Gateway (Entidad de enlace) .....	21
1.4 SOA .....	21
1.4.1 ¿Qué es SOA?.....	22
1.4.2 ¿Qué son los Servicios Web? .....	23
1.4.2.1 XML (Extensible Markup Language) .....	23
1.4.2.2. SOAP (Simple Access Object Protocol).....	24
1.4.2.3. WSDL (Web Service Description Language).....	24
1.4.2.4. UDDI (Universal Description, Discovery and Integration) .....	24
1.4.3 Razones para usar Servicios Web .....	25
1.5 SISTEMAS QUE UTILIZAN SOA. ....	26
1.6 HERRAMIENTAS Y LENGUAJES UTILIZADOS.....	30
1.6.1 Lenguajes de programación .....	30
1.6.2 Lenguaje de modelado .....	30
1.6.3 Herramientas.....	31
1.7 CONCLUSIONES .....	32
<b>CAPÍTULO 2 PROPUESTA DE SOLUCIÓN</b> .....	<b>33</b>
2.1 DESCRIPCIÓN DE LOS PROCESOS FUNDAMENTALES DEL SIGB. ....	33

2.2 FUNCIONALIDADES NECESARIAS PARA LA INTEGRACIÓN .....	35
2.3 DESCRIPCIÓN DEL MÓDULO DE SERVICIOS WEB .....	36
2.3.1 <i>Concepción inicial del módulo de Servicios Web</i> .....	36
2.3.2 <i>Dependencias y Relaciones con otros módulos</i> .....	37
2.3.3 <i>Modelo de Dominio</i> .....	38
2.3.4 <i>Levantamiento de requisitos</i> .....	40
2.3.5 <i>Definición de los casos de uso</i> .....	41
2.3.6 <i>Actores del Sistema</i> .....	44
2.3.7 <i>Diagrama de Clases</i> .....	44
2.3.7.2 <i>Módulos y Clases utilizadas</i> .....	47
2.3.8 <i>Modelo de Datos</i> .....	49
2.4 GUÍA PARA LA CREACIÓN DE UN SERVICIO WEB .....	50
2.4.1 <i>Uso del WSDL</i> .....	50
<b>CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBA .....</b>	<b>52</b>
3.1 DIAGRAMA DE DESPLIEGUE .....	52
3.2 DIAGRAMA DE COMPONENTES .....	53
3.3 CASOS DE PRUEBAS .....	56
3.4 RESULTADOS OBTENIDOS .....	58
<b>CONCLUSIONES .....</b>	<b>60</b>
<b>RECOMENDACIONES .....</b>	<b>61</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>62</b>
<b>BIBLIOGRAFÍA .....</b>	<b>63</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>66</b>
<b>ANEXOS .....</b>	<b>67</b>

## Índice de imágenes

FIGURA 1. DIAGRAMA DE PAQUETES CON DEPENDENCIAS DEL MÓDULO DE SERVICIOS WEB .....	38
FIGURA 2. MODELO DE DOMINIO DEL SISTEMA DE GESTIÓN BIBLIOTECARIA.....	39
FIGURA 3. CASOS DE USO DEL MÓDULO DE SERVICIOS WEB.....	41
FIGURA 4. MODELO DE CLASES DEL MÓDULO DE SERVICIOS WEB.....	45
FIGURA 5. DIAGRAMA DE SECUENCIA DEL CU_MOSTRAR ÚLTIMAS ADQUISICIONES. ....	46
FIGURA 6. DIAGRAMA DE SECUENCIA DEL CU_ INFORMACIÓN PERSONAL. ....	46
FIGURA 7. DIAGRAMA DE SECUENCIA DEL CU_ MOSTRAR REPORTE DE VISITAS. ....	46
FIGURA 8. DIAGRAMA DE SECUENCIA DEL CU_ MOSTRAR TÍTULOS MÁS PRESTADOS.....	47
FIGURA 9. DIAGRAMA DE SECUENCIA DEL CU_ MOSTRAR TEMAS MÁS CONSULTADOS. ....	47
FIGURA 10. DEPENDENCIAS DEL MODULO PERL WSDL::GENERATOR.....	48
FIGURA 11. DEPENDENCIAS DEL MODULO PERL SOAP::LITE. ....	49
FIGURA 12. MODELO DE DATOS DEL MÓDULO SERVICIOS WEB. ....	50
FIGURA 13. DIAGRAMA DE DESPLIEGUE.....	53
FIGURA 14. DIAGRAMA DE COMPONENTES.....	54
FIGURA 15. PANTALLA DEL SERVICIO WEB MOSTRARULTIMASADQUISICIONES.....	56

## Índice de tablas

TABLA 1. DESCRIPCIÓN DEL CU MOSTRAR ÚLTIMAS ADQUISICIONES. ....	42
TABLA 2. DESCRIPCIÓN DEL CU MOSTRAR INFORMACIÓN PERSONAL.....	42
TABLA 3. DESCRIPCIÓN DEL CU MOSTRAR REPORTE DE VISITAS .....	43
TABLA 4. DESCRIPCIÓN DEL CU MOSTRAR TÍTULOS MÁS PRESTADOS.....	43
TABLA 5. DESCRIPCIÓN DEL CU MOSTRAR TEMAS MÁS CONSULTADOS .....	43
TABLA 6. ESPECIFICACIÓN DE ACTORES DEL SISTEMA. ....	44
TABLA 7. CASO DE PRUEBA PARA EL CU MOSTRAR ÚLTIMAS ADQUISICIONES .....	57
TABLA 8. CASO DE PRUEBA PARA EL CU MOSTRAR INFORMACIÓN PERSONAL.....	57
TABLA 9. CASO DE PRUEBA PARA EL CU MOSTRAR REPORTE DE VISITAS .....	58
TABLA 10. CASO DE PRUEBA PARA EL CU MOSTRAR TÍTULOS MÁS PRESTADOS .....	58
TABLA 11. CASO DE PRUEBA PARA EL CU MOSTRAR TEMAS MÁS CONSULTADOS.....	58

## Introducción

La capacidad de razonar que hace a los humanos diferentes de los animales, les permite utilizar la experiencia como una de las principales fuentes de desarrollo y perfeccionamiento humano; de ahí que la experiencia personal o la de otra persona pueda ser empleada para no cometer errores, hacer mejoras o crear algo nuevo y más eficiente.

A través de los tiempos ha sido necesaria para el hombre la transmisión de sus conocimientos con el objetivo de lograr que sus descendientes, familiares y amigos, que toda la humanidad en sí, aprendan de lo que ha vivido y se pueda construir un futuro mejor. A fin de lograr esto se designó guardar todo este conocimiento en un mismo lugar al que toda persona que se sintiera interesada pudiera acceder a consultarlo, así surgen las bibliotecas en Egipto y Mesopotamia donde se guardaban escritos en papiro; más adelante los chinos inventan el papel que hace aumentar la posibilidad de almacenamiento de información y conocimiento y provoca también un auge en la utilización de las bibliotecas. Con el paso de los años las bibliotecas se van haciendo cada vez más importantes y actualmente son muchos los materiales que en ellas pueden ser consultados y muy diversos los servicios que brindan las mismas; hoy son las bibliotecas un lugar de inevitable visita si se quiere aumentar los conocimientos en cualquier área o esfera de la ciencia, la política e incluso la vida social, el carácter y el espíritu.

A medida que fue tomando auge la tecnología digital y la informatización era también inevitable que las bibliotecas, institución con tanto que ofrecer y a donde se dirigían incluso aquellos que desarrollaban esta tecnología, no se vieran involucradas en esta transformación. Con la aparición de los sistemas automatizados, y dada la lentitud a la que se exponía una biblioteca por el creciente acceso a estas, fue también necesario automatizar los servicios que ofrecen.

El desarrollo de la informática como ciencia y la necesidad de optimizar los procesos bibliotecarios dieron paso a la creación de los Sistemas Integrados de Gestión Bibliotecaria (SIGB) los cuales se constituyen por una aplicación informática destinada a automatizar los sistemas y entornos bibliotecarios que se puede aplicar a las funciones y servicios propios de toda biblioteca, tanto públicos como de carácter técnico. Uno de estos sistemas es el SIGB Koha, alternativa libre para la automatización de los procesos bibliotecarios que contiene gran cantidad de funcionalidades incluyendo módulos para circulación, catalogación, adquisición, publicaciones seriadas, reservas, gestión de usuarios y otras.

Desde noviembre de 2011 la biblioteca de la Universidad de las Ciencias Informáticas, cuenta con su propio Sistema de Gestión Bibliotecaria o Solución para Bibliotecas, conformado por tres sistemas que actúan de manera independiente con funciones específicas, los sistemas que forman parte de esta solución son: el Repositorio Institucional basado en DSpace (*software que provee herramientas para la administración de colecciones digitales*) y utilizado fundamentalmente para almacenar información; el Portal de la Dirección de Información, interfaz web para las

publicaciones de la dirección que ofrece a los usuarios información actualizada sobre los servicios que pone a su disposición y sobre la organización de la dirección, y un Sistema Integrado de Gestión Bibliotecaria basado en Koha (*SIGB* desarrollado en Nueva Zelanda *por la empresa Katipo Communications*); sin embargo, solo es posible acceder a los servicios que ofrece la biblioteca visitando el local de la misma o accediendo a uno de estos sistemas, destacando además que cada sistema actúa de manera independiente, siendo necesario en ocasiones trabajar sobre todos ellos a la vez, a esto se le puede sumar la no existencia de relaciones funcionales entre los mismos sino que solamente se comunican mediante un vínculo colocado en el portal (Sistema de Gestión) de la Dirección de Información.

Dada la situación anterior surge el siguiente **problema científico**: ¿Cómo integrar funcionalmente los tres sistemas que conforman la actual solución integral para la biblioteca de la Universidad de las Ciencias Informáticas? Para dar solución a dicha problemática se toma como **objeto de estudio** el proceso de integración de sistemas con una Arquitectura Orientada a Servicios (SOA) y se centra en el **campo de acción** Servicios Web para la gestión bibliotecaria. Se persigue como **objetivo general** de la investigación desarrollar una solución para la integración funcional entre los sistemas que forman parte de la Solución para la Biblioteca de la Universidad de las Ciencias Informáticas utilizando servicios web que permitan interacción directa entre estos sistemas, derivándose el mismo en los siguientes **objetivos específicos**:

- Valorar el Estado del Arte sobre las opciones de integración de software.
- Definir los procesos necesarios para la integración.
- Proponer un conjunto de servicios web para la integración entre los sistemas.
- Desarrollar los servicios web propuestos para la integración.
- Validar la propuesta de solución.

Todo esto **defendiendo la idea** de que si se establece una integración entre los sistemas que forman parte de la solución para la biblioteca de la Universidad de las Ciencias Informáticas basada en servicios web, se facilitará la creación de servicios de negocio interoperables y estrechamente relacionados que puedan fácilmente compartirse dentro de la universidad, fomentando la reutilización y agilidad en los servicios que ofrece la biblioteca, así como la adaptación de estos a las demandas de los usuarios.

Para dar cumplimiento a los objetivos previstos se proponen las siguientes **tareas de investigación**:

- Identificación de las opciones de integración de software existentes.
- Fundamentación de la selección de una opción para la integración de los sistemas.
- Descripción de los conceptos más importantes asociados a Arquitectura Orientada a Servicios.
- Análisis de sistemas que empleen Arquitectura Orientada a Servicios.
- Realización de una caracterización de los procesos del SIGB.
- Identificación en el Sistema Integrado de Gestión Bibliotecaria utilizado de la UCI los

procesos que favorezcan la integración de los sistemas que forman la solución.

- Elaboración del diseño de una propuesta de integración que utilice servicios web.
- Implementación de los servicios web propuestos.
- Realización de pruebas de caja negra a la propuesta de integración.

Los **métodos científicos** empleados para desarrollar la investigación fueron el **Análisis Histórico Lógico** teniendo en cuenta el desarrollo y evolución de los Servicios Web en los sistemas de gestión bibliotecaria y sus principales características y enfoques, así como para seleccionar el lenguaje en que se desarrollarían los mismos; el **Analítico-Sintético** para una vez realizado el estudio sobre los procesos fundamentales de los sistemas que forman la solución para la Biblioteca UCI, y encontrados los puntos de contacto y diferencias entre estos, escoger los que serían necesarios para la integración del Sistema de Gestión Bibliotecaria en la UCI; y la **observación**, método que hizo posible conocer la realidad de la integración entre los componentes del actual Sistema de Gestión Bibliotecaria mediante la percepción directa de los sistemas que la componen, y entender la esencia de la problemática definida ayudando al planteamiento de la idea que se defiende y facilitando el conocimiento del proceso definido como objeto de estudio. Como **estrategia de Investigación** se empleó la **exploratoria** permitiendo esta la familiarización del investigador con el tema objeto de estudio, la situación en que se encuentra y los métodos y técnicas utilizados para dar solución al problema planteado.

Una vez concluido el presente trabajo **se espera** que el Sistema de Gestión Bibliotecaria cuente con una vía sencilla para lograr sistemas estrechamente relacionados entre sí, que además de cumplir con funciones específicas para las cuales se han especializado, también puedan ofrecer servicios competentes a la biblioteca para los cuales quizás no fueron diseñados sin necesidad de que el usuario deba acceder al sistema que se especializa en atender este tipo de petición; igualmente se espera que dichos servicios puedan ser utilizados por otros sistemas informáticos dentro o fuera de la Universidad de las Ciencias Informáticas, poniendo al alcance de todos los interesados la información guardada en nuestros sistemas.

El presente documento está dividido en tres capítulos con la siguiente **estructura**:

**Capítulo 1 Fundamentación Teórica:** Trata aspectos fundamentales concernientes a los Servicios Web así como los principales conceptos y definiciones que brindan sustento teórico a la investigación, incluye un estado del arte sobre las opciones de integración de software y un estudio sobre sistemas que utilizan una Arquitectura Orientada a Servicios, se define una propuesta de integración teniendo en cuenta sus principales características, definiciones y ventajas, así como las características del Sistema de Gestión Bibliotecaria en la UCI y se determinan también los lenguajes y herramientas a emplear.

**Capítulo 2 Propuesta de Solución:** se identifican los procesos fundamentales del Sistema Integrado de Gestión Bibliotecaria como parte fundamental de la solución para la Biblioteca UCI, se definen los servicios más necesarios, y se propone el diseño de los servicios web para la

integración de los sistemas que forman parte de la actual Sistema de Gestión Bibliotecaria.

**Capítulo 3 Implementación y Prueba:** se desarrolla la implementación de la propuesta de solución planteada, describiéndose su funcionamiento y despliegue por nodos físicos, y se valida el funcionamiento de los servicios web implementados; se expone además la manera en que estos serán publicados dentro del Sistema de Gestión Bibliotecaria.

# Capítulo 1: Fundamentación Teórica

En el principio de la computación, en las primeras computadoras corría un programa a la vez, así que en la década de los 50's y hasta mediados de los 70's aproximadamente el paradigma de la computación era "una computadora múltiples usuarios", entonces los equipos eran grandes y costosos y los sistemas centralizados; pero a medida que en una misma máquina se comenzaban a correr varios programas las computadoras se hacen más accesibles a los usuarios y dicho paradigma fue cambiado a "una computadora un usuario" ocupando estas un lugar en los hogares y oficinas, surge así la necesidad de un mecanismo de comunicación entre los programas que se ejecutan en estas al que inicialmente se le denominó *Task to Task* (Tarea a Tarea), con el surgimiento de las redes de computadoras nace también la arquitectura cliente servidor y la comunicación debía realizarse entre un programa X que se ejecuta en un ordenador A y otro programa Y corriendo en el ordenador B; a finales de los 80's las redes de computadora comienzan a tomar auge por lo que los sistemas se vuelven descentralizados y a finales de los 90's internet y la web ya son medios masivos de comunicación y las aplicaciones totalmente distribuidas, de aquí que en la década del 2000 el paradigma ya es "un usuario, múltiples computadoras" y todo o casi todo se puede hacer en la web y las aplicaciones han migrado a lo que conocemos como "web services" o servicios web.

En el presente capítulo se pretende abordar lo relacionado a los servicios web, así como en qué consisten y para que se utilizan, exponiendo las razones por las cuales es necesario su empleo en la integración funcional de los sistemas que forman la actual solución para la biblioteca de la Universidad de las Ciencias Informáticas abordando sus características y el impacto que tienen los mismos.

## 1.1 Sistema de Gestión Bibliotecaria.

La Dirección de Información de la Universidad de las Ciencias Informáticas tiene como misión proporcionar servicios y productos de información de alto valor agregado que apoyen las funciones principales de la Universidad (docencia investigación, extensión universitaria y producción) realizando una adecuada selección de las fuentes de información y la utilización de las tecnologías de la información y las comunicaciones. (Dirección de Información, 2012)

La solución se concibe teniendo en cuenta los tres sistemas expuestos a continuación.

### 1.1.1 Repositorio Institucional

Como consecuencia de la crisis de comunicación científica surge el Movimiento de Acceso Abierto a la información científica basado fundamentalmente en poner a disposición de la

comunidad científica y del público en general, los artículos científicos así como materiales docentes y para la investigación mediante su publicación en revistas de acceso abierto y/o depositado en espacios con acceso totalmente gratuito y libre, a estos espacios se les conoció entonces como Repositorios Institucionales o Temáticos.

Al decir de la vigente edición de la Real Academia de Lengua Española, la palabra repositorio proviene del latín «repositorium», armario o alacena, y lo define como «lugar donde se guarda algo», de ahí que en términos informáticos se pueda decir que un repositorio es un *conjunto de servicios de almacenamiento, gestión y diseminación de materiales digitales que contienen la producción científica, intelectual e investigativa de una determinada comunidad académica o institución*. (Lynch, 2003) (Chan, 2004).

La Asociación de Bibliotecas de Investigación (ARL por sus siglas en inglés) define los repositorios institucionales como: *Archivo digital de los productos intelectuales creados por el profesorado, el personal de investigación y los estudiantes de una institución y accesible a los usuarios finales, tanto dentro como fuera de la institución, con pocas o ninguna barrera*. El contenido de un repositorio institucional es definido por la institución, Académico, Acumulativo y perpetuo, Abierto e interoperable.

*Pueden ser de acceso público, o pueden estar protegidos y necesitar de una autenticación previa. Los depósitos más conocidos son los de carácter académico e institucional y tienen por objetivo organizar, archivar, preservar y difundir la producción intelectual resultante de la actividad investigadora de la entidad*. (Salamanca, 2009)

La Universidad de las Ciencias Informáticas con el fin de hacer su producción intelectual accesible, fácilmente recuperable y asegurar su permanencia en el tiempo, almacena la misma en un repositorio institucional basado en Dspace, software open source desarrollado en java del que recibe sus características, entre las que podemos citar su distribución bajo la licencia BSD (Berkeley Software Distribution), que es multiplataforma, basado en tecnología web y adaptable, además de poseer un gran número de handles o manejadores que ofrecen múltiples posibilidades al usuario. Soporta una gran variedad de datos, incluyendo libros, tesis, fotografías, videos, datos de investigación y otras formas de contenido. Los datos son organizados como ítems que pertenecen a una colección; cada colección pertenece a una comunidad.

Dicho repositorio tiene dos módulos fundamentales que agrupan sus funcionalidades, uno para la administración y el otro para la búsqueda avanzada. El primero es el módulo principal y que permite todo el control del sistema, el trabajo con las comunidades y colecciones así como la administración de usuarios y grupos, flujos de trabajo, estadísticas y otros servicios del sistema; a este pueden acceder solo los usuarios con permisos de administración. El segundo módulo presenta la búsqueda como un aspecto esencial para el repositorio, esta se puede realizar de múltiples maneras: por referencia externa utilizando *handles* (manejadores), por palabras claves en los metadatos o extraídas de texto completo, por título, autor fecha o índice del tema, con miniatura opcional.

### 1.1.2 Sistema Integrado de Gestión Bibliotecaria.

La automatización de las bibliotecas comienza en los años 60, pero no es hasta la década de 1980 que aparece un verdadero SIGB (Sistema Integrado de Gestión Bibliotecaria).

Guadalupe González Herrera (2010), en su tesis "*Software libre vs propietario: una evaluación de sistemas Janium vs. Koha*", capítulo 2, cita a varios autores (Oscar Arriola Navarrete, Félix de Moya Anegón y Luis Ángel García Melero) y expone algunos conceptos de utilidad para definir lo que es un SIGB:

- *Es un conjunto de módulos de aplicación integrados en un solo programa y que comparten una base de datos bibliográfica en común y que ayuda a la gestión de procesos y servicios de las unidades de información.*
- *Sistemas para el proceso automatizado o informático, de información estructurada y no estructurada, sobre actividades y documentos, adaptable a la estructura organizativa de la Biblioteca.*
- *Es un conjunto organizado de recursos humanos que utiliza dispositivos y programas informáticos, adecuados a la naturaleza de los datos que deben procesar, para realizar procesos y facilitar los servicios que permiten alcanzar los objetivos de la Biblioteca: almacenar de forma organizada el conocimiento humano contenido en todo tipo de materiales bibliográficos para satisfacer las necesidades informativas, recreativas y de investigación de los usuarios. (García Melero; García Camarero, 1999).*

Se puede decir que un SIGB es básicamente una aplicación informática para automatizar los sistemas y entornos bibliotecarios y aplicable a los servicios y funciones propios de cualquier biblioteca.

La biblioteca de la Universidad de las Ciencias Informáticas ha automatizado sus servicios mediante un SIGB basado en Koha que es un software de tipo open source, distribuido con licencia GPL, desarrollado en Nueva Zelanda por la empresa Katipo Communications en 1999 y mantenido por una comunidad de desarrolladores de distintos países, de este recibe o hereda el sistema de la UCI múltiples funcionalidades y módulos, y le debe el ser multiplataforma, integrable, adaptable, basado en tecnología web, open source y gran cantidad de funciones y servicios complementarios. Cuenta con módulos para adquisición, catalogación, circulación, informes, catálogo de acceso público en línea (opac), peticiones y usuarios, posee interfaces independiente para la administración (la intranet para el bibliotecario) y el catálogo en línea (para el acceso de los usuarios), y cuenta con una base de datos dual que hace posible que el sistema sea escalable y soporte el volumen de transacciones de datos de cualquier biblioteca sin importar el tamaño de los mismos y lograr una organización eficaz de estos. El sistema utiliza además estándares y protocolos empleados por las bibliotecas.

### **1.1.3 Sistema de Gestión para la Dirección de Información.**

La Dirección de Información mediante un portal Web con una interfaz amigable, garantiza el acceso a la información digital y la integración con el resto de los componentes de la solución. Pretendiéndose implementar proveedores de servicios para el consumo de información externa al sistema, tales como bases de datos y repositorios institucionales a través de protocolos para el intercambio de información (OAI-PMH, Z39.50). Incorporar elementos de aplicaciones 2.0 como: integración con redes sociales, etiquetado y personalización. A su vez, es un sistema de gestión que implementará los servicios más importantes de la Dirección de información, entre ellos certificación de publicaciones, reproducción de documentos y vigilancia tecnológica, entre otros. Dicho portal proporciona acceso rápido y actualizado a la información desde cualquier lugar que se tenga acceso y en cualquier momento, mediante una interfaz gráfica sencilla y amigable al usuario, y a través de aplicaciones web permitirá gestionar el contenido, los procesos y los servicios de la Dirección de Información.

## **1.2 Opciones de Integración.**

Integración es la habilidad de agrupar elementos o aspectos de una entidad o de varias en un ente o conjunto de organismos. En la industria del software y específicamente en la arquitectura de software busca la relación entre el espacio interior y el exterior. La arquitectura de integración persigue una manera eficiente y flexible de combinar recursos y reutilizar soluciones con el objetivo de mejorar operaciones, brindando una vista única donde se consolidan conectores que se definen y especifican el comportamiento e interacción entre elementos de software que abstraen el negocio; estas constituyen la conexión y la colaboración mutua entre actividades o procesos, lo que sugiere que la integración es más que una actividad simple de unir partes.

La integración de sistemas de software puede resultar en una tarea difícil en la que intervienen innumerables factores como plataformas, lenguajes de desarrollo, tecnologías, tiempo y costo, entre otros indicadores y se puede establecer una dependencia del ambiente para realizar la integración, los componentes a integrar, la preparación que tienen estos componentes para integrarse, complejidad de la integración y los procedimientos y criterios para la integración en los cuales pueden estar involucrados los sistemas a integrar haciendo de esta una tarea más o menos compleja.

Existen muchas opciones de integración, cada una de ellas adaptada a las necesidades del entorno para el que fueron concebidas. Los casos más relevantes en cuanto a la integración tienen que ver con la integración de componentes creados por terceras partes con aplicaciones que se encuentran en proceso de desarrollo, o para lograr interoperabilidad entre sistemas que ya se encuentran funcionando pero no se estableció un mecanismo cuando se construyeron para asegurar la comunicación y/o intercambio de información entre ellos.

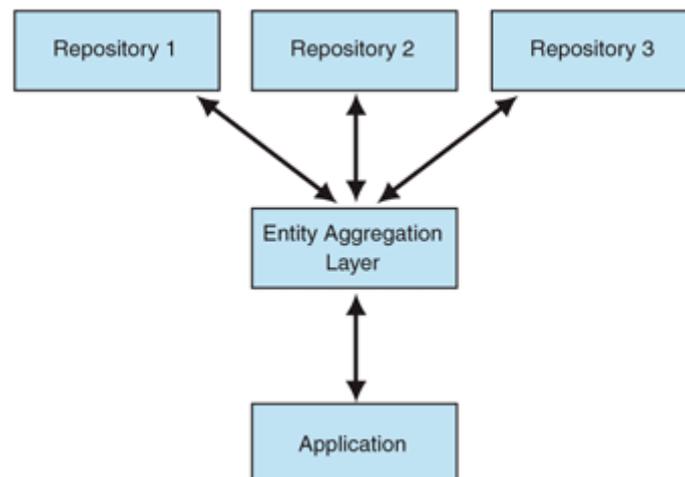
### 1.3 Patrones de Integración.

Los patrones de integración en la informática son la solución a muchos problemas comunes en el desarrollo de software, donde es necesario conectar sistemas aislados y heterogéneos dentro de una o varias organizaciones posibilitando su funcionamiento como un solo sistema. A continuación se exponen los de mayor significación en el proceso de integración de software.

#### 1.3.1 Entity Aggregation (Agregación de Entidades)

*Este patrón introduce una capa de agregación de entidades la cual provee una conexión física con las mismas, hacia sus fuentes originales de datos, las cuales soportan el acceso y manipulación de las instancias de dichas entidades. La capa Entity Aggregation proporciona una representación lógica de las entidades a nivel empresarial con conexiones físicas que apoyan el acceso y la actualización en sus instancias respectivas en los repositorios. (Microsoft Corporation, 2010)*

Este patrón presenta una vista general de toda la información recopilada en los repositorios al usuario. Los datos presentes en el repositorio son ilustrados en la imagen siguiente.



El establecimiento de Entity Aggregation implica un proceso de dos pasos:

- 1) Definición de la representación de toda la empresa que proporciona una representación coherente de la entidad unificada.
- 2) Implementación de una conexión física bidireccional entre esta representación y sus instancias respectivas en los repositorios de servicios.

**Beneficios:** (Microsoft Corporation, 2010)

- **Consenso.** Fuerza el consenso a través de las empresas y unidades funcionales sobre la manera en que las entidades están representadas en un nivel empresarial.
- **Vista única.** Permite una vista única de las entidades empresariales claves, como cliente, cuenta, producto.
- **Mejora del acceso a la información.** Un punto de vista a nivel empresarial de las entidades claves del negocio permite a las aplicaciones tener acceso inmediato a la información correspondiente a estas entidades. El acceso a la información no está limitado por los depósitos

que los albergan.

– **Reducción de la disonancia semántica.** Elimina la disonancia semántica a través de las aplicaciones existentes que trabajan en los mismos elementos de datos desde múltiples repositorios.

– **Céntrico.** Es compatible con una ubicación central para la validación de los datos dentro de los repositorios.

– **Reducción del impacto del cambio.** Reduce el impacto potencial de los cambios en los repositorios. Dependiendo de la naturaleza de los cambios realizados, esta capa puede continuar sirviendo a las necesidades de las aplicaciones, mientras que estos cambios están en marcha.

**Riesgos:** (Microsoft Corporation, 2010)

– **Capa adicional.** Introduce una capa adicional de arquitectura que puede afectar negativamente el rendimiento final de la solución.

– **Consenso.** Es necesario un consenso a través de las unidades de negocio en la definición de la escala de representación de las entidades empresariales.

– **Reingeniería de las aplicaciones.** Las aplicaciones existentes que están estrechamente unidas a una serie de repositorios tendrían que ser rediseñadas para dar cabida a la capa de la arquitectura nueva. Además, no siempre es posible rediseñar algunas soluciones, sobre todo para las aplicaciones empaquetadas.

Este patrón ofrece la mejor solución a emplear para garantizar el proceso de integración que se quiere lograr en el Sistema de Gestión Bibliotecaria utilizado en la universidad. Además, ofrece ventajas sobre otros patrones de integración debido a que es fácil de implementar, ofrece una visión general de la información que se encuentra en los repositorios donde la lógica y la complejidad quedan ubicadas en capa añadida por el patrón.

### 1.3.2 Process Integration (Integración de Procesos)

*Define un modelo de proceso de negocio que describe los pasos individuales que componen la función de negocios complejos. Crea un componente de proceso administrador distinto, que puede interpretar varias instancias simultáneas de este modelo y que puede interactuar con las aplicaciones existentes para llevar a cabo los distintos pasos del proceso.* (Microsoft Corporation, 2010)

Este patrón permite una administración centralizada de los procesos que se integran independientemente del dominio de estos, debido a que solo interpreta las estructuras de estos procesos de forma básica. El Sistema necesita integrar componentes y no procesos por consiguiente no es la solución a emplear para solucionar los problemas de integración y publicación de los mismos.

### **1.3.3 Portal Integration (Portal de Integración)**

*Es comparativamente un simple formulario de integración. Es popular porque es generalmente mucho más fácil de aplicar y menos intrusivo que otros tipos sofisticados de integración, como Integración de Procesos. Con este patrón, el negocio de proceso describe la secuencia de tareas, no tienen que estar representados en un modelo de proceso exacto sino que reside en la cabeza del usuario. En el patrón Portal de Integración las aplicaciones individuales interactúan mediante la integración de datos, integración funcional, o la integración de presentación. (Microsoft Corporation, 2010)*

La integración dentro de este patrón puede presentarse como integración de datos, integración funcional o integración de presentación. Este patrón pudiera decirse utilizado en el Sistema de Gestión Bibliotecaria debido a que precisamente uno de los componentes en los sistemas de la Biblioteca de la UCI es un portal, pero este no resuelve todos los problemas o necesidades de integración sino que ofrece solamente links de acceso a los otros componentes de la solución y no centraliza las funcionalidades de estos ni toda la información de los mismos.

### **1.3.4 Message Broken (Intermediario de mensajes)**

*En este patrón la comunicación entre aplicaciones implica sólo el remitente, el intermediario de mensajes y los receptores designados. En lugar de comunicarse entre sí, las aplicaciones se comunican sólo con el intermediario de mensajes. Una aplicación envía un mensaje al intermediario de mensajes, proporcionando los nombres lógicos de los receptores. El intermediario de mensajes busca las aplicaciones registradas bajo esos nombres lógicos y luego pasa el mensaje a ellos. (Microsoft Corporation, 2010)*

Este patrón está basado principalmente en la construcción de una aplicación middleware que se encarga de entender la lógica individual de cada aplicación. Aunque resuelve y simplifica muchos problemas no es una solución óptima a emplear debido a su complejidad a la hora de implementarlo.

### **1.3.5 Publish/Subscribe (Publicación/Suscripción)**

*Amplía la infraestructura de comunicación, generando tópicos o mediante inspeccionado dinámico del contenido del mensaje. Para que una aplicación consumidora pueda consumir un mensaje que le compete, dicha aplicación tiene que suscribirse a un tópico específico para escuchar únicamente los mensajes asociados a dicho tópico. (Microsoft Corporation, 2010)*

Existen tres variantes

- Basado en listas.
- Publicación y Suscripción basado en una notificación masiva.
- Publicación y Suscripción basado en el contenido del mensaje.

Este patrón está basado en la suscripción y publicación de tópicos donde el intercambio de

mensajes es el método de integración. La solución ofrecida no es la ideal para ser utilizada en el polo donde el intercambio de información resulta fundamental y donde no es necesaria la suscripción a un componente para poder intercambiar información con él.

### **1.3.6 Gateway (Entidad de enlace)**

*Gateway es un componente que abstrae el acceso a recursos externos. En caso de que se tengan diversas aplicaciones desarrolladas en distintas plataformas y necesitan comunicarse con recursos externos, en vez de que cada aplicación acceda a dichos recursos o servicios por cuenta propia, se crea una entidad de enlace, la cual tiene conocimiento del lugar en que los servicios o recursos se encuentran.* (Microsoft Corporation, 2010)

Este patrón sugiere un gran esfuerzo de mantenimiento y construcción donde un cambio implica mucho trabajo si se pusiera en práctica como método de integración en la biblioteca; por tanto, no es una solución viable a emplear.

En numerosos proyectos de la Universidad de las Ciencias Informáticas (UCI) se han empleado tecnologías y técnicas de integración en la solución de problemas de heterogeneidad entre sistemas, aplicaciones y componentes. Cada uno de ellos implementa una manera distinta en la solución, por lo que no existe dentro de la universidad el establecimiento de una solución de integración que sea aplicable a todos los problemas, debido a que cada uno de ellos requiere un modelo diferente para la concepción de la integración. Varios proyectos han trabajado en componentes reutilizables que serían expuestos a repositorios de componentes a donde se podría acceder para descargarlos y utilizarlos en soluciones informáticas que se encontrarán en desarrollo. Entre los esfuerzos por crear soluciones estándar de integración y publicación de componentes, se encuentra la de crear aplicaciones que expongan las funcionalidades de los componentes como servicios, este esfuerzo ha dado como resultado la publicación de un directorio en <http://uddi.uci.cu>, desde el cual se publican servicios relacionados con varios componentes al servicio de la comunidad universitaria.

## **1.4 SOA**

Como alternativas para la integración de sistemas homogéneos han surgido las tecnologías middleware y SOA. Middleware introduce una capa entre las aplicaciones que actúa como mediador entre las mismas, es físicamente una aplicación que se encarga de la comunicación permitiéndonos abstraernos de la complejidad y heterogeneidad de las redes de comunicación, sistemas operativos y lenguajes de programación. Por su parte, SOA soluciona los problemas de integración de manera más sencilla y eficiente, aplica la solución más práctica logrando estandarizar la comunicación; la orientación a servicios ofrecida por esta arquitectura es tanto un

marco de trabajo para el desarrollo de software como para la implementación; si se tiene bien definido lo que se desea integrar entonces SOA es la mejor opción a emplear.

### 1.4.1 ¿Qué es SOA?

SOA es una forma de arquitectura para sistemas distribuidos, es una forma de concebir el negocio en términos de servicios interconectados. Esta constituye una arquitectura conceptual sobre cómo organizar las empresas en términos de aplicaciones, servicios y procesos de negocios que existen en ellas y se caracteriza por las siguientes propiedades:

- **Vista Lógica:** El servicio es una abstracción (vista lógica) de los programas, bases de datos, procesos de negocio, etc., definido en términos de lo que hace (llevando a cabo una operación de negocio).
- **Orientación a Mensajes:** El servicio se define formalmente en términos de los mensajes intercambiados entre agentes proveedores y solicitantes, y no está basado en las propiedades de los agentes. La estructura interna del agente (lenguaje de programación, BD, proceso, etc.) se abstrae en SOA. Esto permite incorporar cualquier componente o aplicación a esta arquitectura “decorando” estos componentes con software de gestión y conversión.
- **Orientación a la Descripción:** Un servicio se describe con metadatos procesables. La descripción da soporte a la naturaleza pública de SOA: sólo se incluyen en la descripción aquellos detalles que se exponen públicamente y son importantes para el uso del servicio. La semántica de un servicio debe documentarse, directa o indirectamente, por su descripción.
- **Granularidad:** Los servicios tienden a usar un pequeño número de operaciones con mensajes relativamente complejos.
- **Orientación a la Red:** Los servicios tienden a usarse a través de la red, aunque este no es un requisito absoluto.
- **Neutral a la Plataforma:** Los mensajes se envían en un formato estándar y neutral a la plataforma, distribuido a través de las interfaces (XML).

Una SOA brinda beneficios de negocio tales como la alineación de las tecnologías de la información con los objetivos estratégicos de la organización, la mejora en la agilidad del negocio y el incremento en la reutilización de los servicios con el consiguiente ahorro de costes, mantenimiento y tiempo; a la vez que establece una serie de retos principalmente las necesidades de negocios relativas a estrategias y procesos, requiriendo además de las siguientes interacciones:

1. Un servicio web publica su definición (WSDL) en un repositorio / directorio / registro distribuido (UDDI).
2. El consumidor del servicio web busca la definición del servicio en el repositorio.
3. El repositorio usa la información de la definición (WSDL) para enlazar con el servicio y enviar peticiones (SOAP) al servicio que ofrece el proveedor de servicios.

Un factor muy influyente y que hace que las Arquitecturas Orientadas a Servicios sean más

utilizadas actualmente son los servicios web, ambos son muy apropiados para aplicaciones que deben operar a través de redes en las que la fiabilidad y la velocidad no se pueden garantizar, donde no existe habilidad de gestionar la instalación para que todos los solicitantes(clientes) y proveedores se actualicen a la vez, los componentes de un sistema distribuido se ejecuten en distintas plataformas y productos y/o donde la aplicación existente necesite ser expuesta para ser utilizada a través de la red y pueda decorarse como un servicio web.

### 1.4.2 ¿Qué son los Servicios Web?

Los Servicios web surgen dada la necesidad de estandarizar la comunicación entre distintas plataformas y lenguajes de programación sin necesidad de hacer distinciones entre estos; de las múltiples definiciones existentes se citan algunas a continuación:

- *“Self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces”* (UDDI consortium).
- *“es un sistema software identificado por una URI, cuyos interfaces públicos y enlaces se definen y describen usando XML. Su definición puede ser descubierta por otros sistemas de software. Estos sistemas pueden interactuar con el servicio Web de la forma prescrita por su definición, usando mensajes basados en XML a través de protocolos estándares de Internet”* (Web W3C).
- *“a standardized way of integrating web based applications using XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone, XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available, and UDDI is used for listing what services are available”* (Webopedia, 2010).
- *Es una interface capaz de recibir una petición, activar unos procesos y devolver los resultados. Todo esto, en Internet y a través de protocolos de red como HTTP, FTP y SMTP* (IDEC: Infraestructura de Datos Espaciales de Cataluña)

Se puede decir entonces que un servicio web es básicamente una función o procedimiento que puede ser accedida vía web por cualquier programa o aplicación sin importar en que plataforma reside el servicio o en que lenguaje ha sido desarrollado, describiendo una forma estandarizada de integración web mediante el uso de XML, SOAP, WSDL y UDDI sobre los protocolos de internet.

#### 1.4.2.1 XML (Extensible Markup Language)

Es un lenguaje desarrollado por el W3C Consortium para permitir la descripción de información contenida en la red de redes a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) pueden buscar, comparar y compartir información en la red. El formato de XML es muy parecido al del HTML aunque no es una

extensión ni un componente de éste; permite a los desarrolladores crear sus propias etiquetas (tags) para habilitar definiciones, transmisiones, validaciones, e interpretación de los datos entre aplicaciones y entre organizaciones. Este es usado para describir los datos.

#### **1.4.2.2. SOAP (Simple Access Object Protocol)**

SOAP es un protocolo de mensajería construido en XML que se usa para codificar información de los requerimientos de los Servicios web y responder los mensajes antes de enviarlos por la red, estos pueden ser transportados por los protocolos que funcionan en la Internet, como SMTP, MIME y HTTP. En su versión 1.2, recomendado por W3C en 2003, define un protocolo que da soporte a la interacción (datos + funcionalidad) entre aplicaciones en entornos distribuidos y heterogéneos, es interoperable (neutral a plataforma y lenguajes de programación, independiente del HW y protocolos). SOAP define cómo organizar información usando XML de forma estructurada y tipada para intercambiarla entre distintos sistemas, por lo que en los servicios web se ocupa de la transferencia de los datos.

#### **1.4.2.3. WSDL (Web Service Description Language)**

WSDL es un lenguaje especificado en XML creado originalmente por IBM, Microsoft y Ariba que se ocupa de definir los Servicio web como colecciones de puntos de comunicación capaces de intercambiar mensajes. Un archivo WSDL es un documento XML que describe los Servicios Web, en particular sus interfaces, debe definir los mecanismos de acceso (protocolos) a los servicios Web y tiene la necesidad de definir la localización de los mismos. WSDL, permite definir interfaces más complejas y expresivas permitiendo definiciones de interacciones asíncronas y diferentes paradigmas de interacción, y la posibilidad de combinar o agrupar operaciones. Este lenguaje es parte integral de UDDI y del registro global de XML, o sea, es un estándar de uso público (no se requiere pagar licencias ni royalties para usarlo) que describe los servicios web disponibles.

#### **1.4.2.4. UDDI (Universal Description, Discovery and Integration)**

UDDI, creado originalmente por IBM, Microsoft y Ariba, y pasado a manos de OASIS desde su versión 3 en 2002 ([www.oasis-open.org](http://www.oasis-open.org)), se concibió como un Servicio de Directorio y Nombrado sofisticado que especifica un marco para describir y descubrir Servicios Web definiendo estructuras de datos y APIs (*parte del sistema operativo que provee a las aplicaciones de una interfaz de uso común o interfaz similar*) para publicar descripciones de servicios en el registro y para consultar el registro buscando descripciones publicadas. Estas especificaciones tienen dos objetivos esenciales: ser un soporte a los desarrolladores para encontrar información sobre servicios web y poder construir clientes, y facilitar el Enlace Dinámico de Servicios Web, permitiendo consultar referencias y acceder a servicios de interés.

UDDI es un directorio distribuido que opera en la Web y que permite a las empresas publicar sus Servicios web, para que otras empresas los conozcan y utilicen, este viene a ser como un repositorio que contiene descripciones de servicios web que se ocupa de dar a conocer cuales están disponibles.

### 1.4.3 Razones para usar Servicios Web

La existencia de tres sistemas que en la Universidad de las Ciencias Informáticas dan lugar a la solución de software para la gestión de la información y los procesos en su biblioteca; y el hecho de que cada uno tenga funcionalidades específicas y diferentes entre sí, unido a que si se desea conocer información contenida en un sistema distinto al que se está trabajando será necesario abrir también el mismo, hacen ver la necesidad de integración funcional entre dichos sistemas.

Existen varias razones para optar por un enfoque SOA, y más concretamente un enfoque SOA basado en servicios web para integrar dichos componentes de software. Los servicios web son aplicaciones o tecnologías que permitirán intercambiar datos entre los sistemas de la Solución para Bibliotecas, manteniendo la coherencia y consistencia de la información en dichos sistemas y haciendo provisión de un conjunto de funcionalidades que podrán ser accedidas desde cualquier lugar. A continuación se enumeran factores que fueron tomados en cuenta para seleccionar dicha solución:

- **Reutilización:** Un principio fundamental de SOA es la reutilización. Con esta arquitectura el sistema podrá reutilizar las funciones de interés que identifique dentro de la biblioteca para que puedan ser expuestas como servicios y ser reutilizadas para cubrir nuevas necesidades de negocio.
- **Interoperabilidad:** SOA resuelve los problemas presentes en el sistema de interoperabilidad, con su aplicación no importará el lenguaje de construcción de cualquier componente para utilizar sus funcionalidades, debido a que estas funcionalidades serán expuestas como servicios, independientes de plataforma, lenguaje de programación y sistema operativo.
- **Escalabilidad:** En SOA los servicios están acoplados débilmente por tanto las aplicaciones que hacen uso de estos servicios en el sistema podrán escalar fácilmente. Esto es debido a la poca dependencia que existe entre las aplicaciones clientes y los servicios usados por estas.
- **Flexibilidad:** Los servicios que serán implementados en el sistema entre otras características poseerán un bajo acoplamiento entre ellos, por tanto, los cambios hechos en los componentes no afectarían las aplicaciones u otros servicios que los usen siempre que se mantenga la interfaz de los componentes.
- **Eficiencia de coste:** Entre las características más importantes de SOA se encuentra la eficiencia en los costes de producción, esto se debe a que los servicios son construidos una sola vez y sus funcionalidades son expuestas para que sean utilizadas. El sistema no necesitaría reconstruir una nueva funcionalidad cada vez que la necesite sino reutilizar las que ya tiene

consumiéndolas como servicios.

## 1.5 Sistemas que utilizan SOA.

### Suite 102

Suite 102 es una herramienta para gestionar los ciclos de vida de los documentos, desde su creación, el almacenamiento, la distribución, la edición,..., hasta la destrucción. Es un software que se construye para adaptarse a todos estos cambios, para ello crea una base de datos centralizada y única basada en los diferentes lenguajes de metadatos actuales que permite rediseñar la estructura sin necesidad de crear nuevos contenedores de datos; de esta forma es capaz de catalogar todo tipo de información sin importar soporte, lenguaje, ubicación, etc..., e incrustando en la descripción cualquier extensión informática que se pueda almacenar, visualizar y/o editar.

En la gestión Bibliotecaria, Suite 102 implementa las normativas para la comunidad documental y bibliotecaria MARC21, Dublin Core, XMLMARC, ISO 2709, Z39.50, FRBR, ISBD, OAI-PMH, OPEN URL, XML y todas las recomendaciones y normativas de accesibilidad y uso de la W3C. Es altamente parametrizable, lo que permite la definición de perfiles de entrada de datos, salida de información, páginas de consultas, tipos de búsqueda, definición de estructuras, criterios de búsqueda, etc...

Está construido con herramientas de desarrollo que permiten la implantación del producto sobre cualquier sistema operativo, el lenguaje es PHP, Java Script, XHTML. El acceso a datos se realiza con motores SQL y XQUERY; los principales motores que maneja son: SQL Server, mySql, Oracle y Postgre SQL. Se basa en la creación de servicios web XML y necesita para su funcionamiento un servidor de bases de datos, un servidor web y un navegador de html. Los clientes solo necesitan un navegador web. Al ser código abierto permite la construcción de una interfaz de escritorio personalizada. Y las funciones integradas en la versión básica son, entre otras: catalogación, consultas normalizadas, carga automática preconfigurada, servicio de préstamo, servicio de reservas, lectores, módulo de estadísticas preconfigurado y consulta pública (OPAC).

### **CiteSeer.ist: *Cientific literature Digital Library***

CiteSeer es un servicio automático que descubre nuevas publicaciones académicas en la web para descargarlas, convertirlas y procesarlas; permite a los usuarios finales buscar en colecciones de documentos a través de citas; identifica publicaciones relacionadas usando características basadas en citas tales como citación y bibliografía activa.

CiteSeer soporta funcionalidades para corrección del error distribuido, lo cual permite a la comunidad de usuarios corregir y actualizar los ítems de meta-datos automáticamente extraídos, y precisamente esta facilidad de acceso a grandes cantidades de meta-datos vía web es lo que lo hace exitoso. Sus servidores han sido comprados a OAI-PMH (Open Archives Initiative protocol for

metadata harvesting) de tal forma que su colección pueda ser accedida por recolectores de metadatos ya que OAI-PMH no provee un soporte extenso para funcionalidades específicas de CiteSeer ni permite una integración directa de este dentro de la Semántica Web.

La integración de servicios web es claramente una de las áreas más activas en las investigaciones de la comunidad de Semántica Web, así como solucionar los problemas de interoperabilidad con los servicios de CiteSeer a partir del diseño e implementación de interfaces semánticas. De ahí que se desarrollara una API para CiteSeer que proporciona un punto de entrada natural al sistema similar a la forma de búsqueda basada en web (métodos de búsquedas), métodos que retornan todos los registros de meta-datos para un recurso dado (métodos de acceso a objetos), métodos de bibliografía orientada (devuelven información básica del documento), una miscelánea de métodos (propiedades adicionales que no son proporcionados por la interfaz web de CiteSeer) y métodos de administración y registro.

### **Red Municipal de Bibliotecas de Murcia (RMBM)**

Murcia es una región española que cuenta con una red de fuentes de información pertenecientes a diferentes bibliotecas integradas entre sí (113 bibliotecas pública de la región de Murcia y centros de lectura repartidos por sus 45 municipios además de 5 unidades móviles) donde agrupa bibliotecas digitales de ámbito autonómico, estatal, internacional, e infantil.

La Biblioteca Regional de Murcia como cabecera de la Red de Bibliotecas Públicas, garantiza la difusión y conservación del patrimonio informativo de la comunidad en cualquiera de sus formas; favorece la coordinación, el impulso, desarrollo y atención a los servicios bibliotecarios de la región. Igualmente funciona como nexo de unión informática entre los puntos de servicios de la Red regional y el Sistema Español de Bibliotecas y pretende ser el centro modelo y asesor de innovaciones tecnológicas en el campo profesional para toda la región. Esta Biblioteca por su parte, ofrece servicios para la ayuda en línea y localización de información en su catálogo, referencia en línea, permite gestionar en línea la renovación de los documentos dejados en préstamo al usuario, etc...

El catálogo colectivo de la RMBM (ccRMBM) contiene información de libros, discos, revistas, mapas, de los que el usuario puede disponer, y ofrece servicios web ligados al carné RMBM para consultar préstamos, renovar préstamos y reservar documentos. Para acceder a cualquiera de estos servicios se debe iniciar sesión introduciendo el carné y contraseña de usuario en el servicio Mi Biblioteca de la Red de Bibliotecas Públicas de la Región de Murcia desde la página del catálogo colectivo.

### **Biblioteca de la FIEE**

La biblioteca de la Facultad de Ingeniería Eléctrica y Electrónica (FIEE) de la Escuela Politécnica Nacional de Quito, en Ecuador ha existido desde hace aproximadamente 32 años en los que ha evolucionado hasta automatizar sus procesos utilizando el SIGB Koha y poseer un sistema abierto

de consulta. Esta última característica supone un sistema de seguridad anti-hurto robusto y confiable; este sistema de consultas también trajo consigo que el trabajo de las personas encargadas fuera excesivo en horarios de máxima afluencia, por lo que el usuario debía esperar un tiempo prolongado para ser atendido. Dada esta situación problemática se elabora un proyecto con el objetivo de incorporar tecnología efectiva en el control de entrada y salida de los libros dentro de una sala de lectura, mejorando la atención al usuario con una mayor agilidad del servicio dentro de la sala al momento de los préstamos de los ejemplares de una colección bibliográfica.

Esta biblioteca desarrolló un sistema de auto préstamo de libros basado en tecnología RFID (Radio Frequency Identification; es una tecnología inalámbrica que puede realizar la identificación de cualquier tipo por medio de ondas de radio, utilizando comunicación bidireccional entre un lector y una etiqueta) donde cada uno de los ejemplares cuenta con una etiqueta RFID pasiva, y se completa con un lector RFID de corto alcance y un par de antenas que forman parte del sistema anti-hurto que con una alarma auditiva alerta a la administración cuando se detecta el bit de seguridad electrónica de la etiqueta del ejemplar.

Para utilizar el sistema los usuarios deben acceder al sitio web diseñado con ese propósito y autenticarse con su número de identificación y contraseña; la interfaz está enlazada por medio de servicios web con la base de datos del SIGB Koha que utiliza la Biblioteca de la FIEE que ofrece la información de los libros prestados al usuario y habilita la opción de auto préstamo. Los servicios web que ofrece el sistema se listan a continuación y están disponibles en <http://biee.epn.edu.ec:8090/cgi-bin/koha/ilsdi.pl>

Servicio	Descripción
AuthenticatePatron	Envía los datos de usuario y contraseña para ser validados dentro del sistema.
GetPatronInfo	Retorna la información del usuario registrado en el sistema.
GetRecordFromBarcode	Retorna los datos del autor y título del libro correspondiente al número almacenado en la etiqueta RFID (actual código de barras).
GetDueDate	Retorna la fecha de devolución del libro registrado.
Issueltem	Envía información para validar el préstamo del libro en el sistema.

### **Pontificia Universidad Católica de Chile**

Cuentan con un sistema de servicios y recursos integrados de información, compuesto por una biblioteca virtual y nueve bibliotecas presenciales que depende de la Vicerrectoría Académica de

la universidad y se relaciona con las diferentes facultades a través de entregas de colecciones y servicios diferenciados.

Su sistema de bibliotecas tiene como misión contribuir al logro de la misión de la universidad y ser parte del proceso de aprendizaje, enseñanza e investigación de la comunidad universitaria, con servicios y recursos de información presenciales y virtuales de excelencia en un entorno colaborativo y global, facilitando a los usuarios el descubrimiento, acceso y uso de la información.

El Sistema de Bibliotecas de la Pontificia Universidad Católica de Chile (SIBUC), líder entre las bibliotecas del país, responde con servicios de información, colecciones, tecnología y espacios físicos adecuados a las necesidades de las distintas unidades académicas de la universidad. Las bibliotecas están repartidas en los cuatro campus de la universidad y en el Hospital Sótero del Río, próximas a las unidades académicas que atienden. Disponen de acceso inalámbrico (Wi-Fi), salas de computadoras y variados ambientes para consultar material impreso, audiovisual, digital o virtual.

El SIBUC posee una de las mejores colecciones universitarias del país, dicha colección se enriquece anualmente con nuevas incorporaciones de material bibliográfico y recursos virtuales (colección en línea); obtienen materiales que no están en su sistema por convenio con otras instituciones de préstamos interbibliotecarios. El catálogo SIBUC contiene colecciones impresas en línea y también dan acceso a catálogos de otras instituciones.

Los procesos bibliotecarios han sido automatizados con el software Aleph 500, este tiene módulos de adquisiciones, catalogación, catálogo en línea, circulación y control de publicaciones periódicas, además de permitir el acceso en línea autónomo y expedito a colecciones reales y virtuales a nivel nacional e internacional que cumplan con el estándar Z39.50.

Este sistema utiliza el concepto de servicio web y cuenta con una componente que realiza esta comunicación en formato XML entre los distintos módulos del sistema e incluso permitir conectividad con otros sistemas. Los servicios web se utilizan para actualizar informaciones de usuarios, operaciones de préstamos, renovación de ejemplares, consultas a otras fuentes de información externas, etc. Al decir de Fernando Testón Reyes, trabajador de la Subdirección de Innovación tecnológica de la Dirección de Informática en la universidad en el área de la biblioteca: “los servicios web son el corazón de los sistemas que permiten la operación de nuestras bibliotecas en esta pieza de software”. El último módulo implementado permite integrar consultas sobre recursos físicos y electrónicos suscritos en distintos proveedores a nivel mundial.

Todos estos sistemas implementan funcionalidades propias de la gestión bibliotecaria en correspondencia con los servicios que brindan y las tecnologías que utilizan, su estudio confirma la tendencia actual a utilizar servicios web y la importancia que se les concede a estos en la web; de manera que pueden ser tomados como referencia para la elaboración de la propuesta de solución planteada.

## **1.6 Herramientas y Lenguajes utilizados.**

Para el desarrollo de la propuesta de solución a la integración funcional entre los sistemas que forman parte del Sistema de Gestión Bibliotecaria se utilizaron diferentes lenguajes de programación, modelado y herramientas que serán descritos a continuación.

### **1.6.1 Lenguajes de programación**

#### **Perl**

Perl es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

*Es un lenguaje libre de uso, eso quiere decir que es gratuito. Antes estaba muy asociado a la plataforma UNIX, pero en la actualidad está disponible en otros sistemas operativos como Windows. (Alvarez, 2001 )*

*Perl toma características del C y del lenguaje interpretado bash. La estructura completa de Perl deriva de lenguaje C, es un lenguaje imperativo, con variables, expresiones, asignaciones y bloques de códigos delimitados por llaves y tiene una gran potencia en la manipulación de textos debido a que incluye expresiones regulares que facilitan el trabajo con textos. (Wall, perl.org)*

#### **XML**

*Es un lenguaje de etiquetado extensible que juega un papel fundamental en el intercambio de datos, es simple pero estricto, es muy parecido a HTML pero su función principal es la de describir datos y no mostrarlos, permitiendo la lectura de estos mediante diferentes aplicaciones; en resumen, XML sirve para estructurar, almacenar e intercambiar información. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. (w3c XML 2009).*

**WSDL** (Web Services Description Language).

*Es un protocolo basado en XML que describe los accesos al servicio web. Podríamos decir que es el manual de operación del servicio web, porque nos indica cuales son las interfaces que provee el Servicio Web y los tipos de datos necesarios para la utilización del mismo. (w3c WSDL 2009).*

### **1.6.2 Lenguaje de modelado**

#### **UML**

UML (Lenguaje de Modelación Unificado) es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos.

*UML no es un método de desarrollo, no va a decir cómo pasar del análisis al diseño y de este al código, no son una serie de pasos que te llevan a producir código a partir de unas*

especificaciones. Al no ser un método de desarrollo es independiente del ciclo de desarrollo que vayas a seguir, puede encajar en un tradicional ciclo en cascada, o en un evolutivo ciclo en espiral o incluso en los métodos ágiles de desarrollo. (García, 2005)

### **1.6.3 Herramientas**

#### **Geany**

*Geany es un editor de texto utilizando el kit de herramientas GTK2 con las características básicas de un entorno de desarrollo integrado. Fue desarrollado para proporcionar un IDE pequeño y rápido, que tiene sólo unas pocas dependencias de otros paquetes y es compatible con tipos de archivos múltiples. (GeanyORG, 2012)*

Está disponible para distintos sistemas operativos, como GNU/Linux, Mac OS X, BSD, Solaris y Microsoft Windows. Es distribuido como software libre bajo la Licencia Pública General de GNU. Tiene soporte para muchos lenguajes de programación como C, C++, C#, Java, JavaScript, PHP, HTML, CSS, Python, Perl, Ruby, Fortran, Pascal y Haskell.

Algunas de las características más destacadas de Geany son: autocompletado, soporte multidocumento, soporte de proyectos, coloreado de sintaxis y emulador de terminal incrustado.

#### **Apache**

Es un servidor web al que muchos servidores le reiteran su confianza dada su fácil configuración, robustez y estabilidad que hacen de él un servidor por excelencia. Entre sus principales características se destaca la de ser prácticamente universal pues corre en la mayoría de los sistemas operativos.

*El servidor Apache es usado por múltiples razones como disponibilidad, facilidad de instalación, necesidad de pocos recursos de hardware, precio, disponibilidad del código fuente, entre otras. Existen otros aunque suelen estar especializados en nichos concretos de mercado. Corre en una multitud de sistemas operativos, lo que lo hace una herramienta multiplataforma. (Ciberaula, 2009).*

#### **MySQL**

*MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la licencia GNU/GPL, este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación, es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. (Netpecos, 2009).*

#### **Visual Paradigm**

Visual Paradigm es una herramienta que soporta el ciclo de vida completo del desarrollo de software. Éste software soporta la ingeniería inversa, generación de código, importación desde Rational Rose, exportación e importación XML, editor de figuras, generador de informes, etc. Visual Paradigm es un software gratuito bajo la licencia GPL y comercial. Entre sus principales ventajas tenemos que soporta aplicaciones web; la generación de código para Java y exportación como HTML; es una herramienta fácil de instalar y actualizar y sus ediciones son compatibles entre sí.

## **1.7 Conclusiones**

La utilización del patrón Entity Aggregation (agregación de entidades) que aboga por la creación de una capa abstracta de datos que soporta las conexiones y el acceso a datos en vinculación con una Arquitectura Orientada a Servicios y Servicios Web suponen una integración entre los componentes del actual Sistema de Gestión Bibliotecaria, de manera que ya no sean sistemas aislados sino que se puede compartir la información obtenida desde sus fuentes originales de datos, aún cuando estos son sistemas diferentes desarrollados con tecnologías diferentes podrán contar con datos comunes. El ejemplo recibido de otros sistemas tales como la Red Municipal de Bibliotecas de Murcia y la biblioteca de la Pontificia Universidad Católica de Chile muestran los servicios web como una alternativa en la agilidad de los procesos bibliotecarios y facilidad en los accesos a los mismos, de manera que el estudio hasta aquí realizado indica que un enfoque SOA orientado a servicios web permitirá la integración pretendida en el actual Sistema de Gestión Bibliotecaria de la Universidad de las Ciencias Informáticas.

## Capítulo 2 Propuesta de Solución.

El actual Sistema de Gestión Bibliotecaria utilizado en la Universidad de las Ciencias Informáticas cuenta con tres sistemas, de los cuales el Sistema Integrado de Gestión Bibliotecaria es el que más información y funcionalidades implementa, de ahí que sea utilizado como punto de partida para la integración, facilitando a los demás sistemas la información que solamente él posee y que les pueda resultar de utilidad.

La esencia de este capítulo es elaborar una propuesta de diseño para dar solución a la integración basada en un estudio detallado de los procesos fundamentales del SIGB. Se realizará un levantamiento de requisitos, lo que servirá como punto de partida para la elaboración de servicios web que cumplirán con las funcionalidades necesarias y de manera satisfactoria, realizando además una descripción y representación de estos, y elaborando un Modelo de Diseño que sirva como punto de partida para la implementación de los mismos.

### 2.1 Descripción de los procesos fundamentales del SIGB.

El SIGB está dividido en dos interfaces web, una de ellas para el trabajo de la administración (Intranet Administrativa) y la otra para el acceso del resto de los usuarios, dígame asociados o no, que soliciten un servicio determinado.

Los módulos o subsistemas que componen al SIGB, agrupan programas y estructuras de datos, que modelan y automatizan procesos y servicios de áreas específicas de la biblioteca, estos permiten la reutilización de funcionalidades y ahorro en el tiempo de desarrollo. El término integrado hace referencia a que se agrupan todas las funcionalidades necesarias para la gestión de la Biblioteca, usando para ello una base de datos común a todos los módulos o subsistemas.

El SIGB incluido en la actual Solución para Bibliotecas está dividido en seis módulos o subsistemas: Adquisición, Catalogación, Circulación, Peticiones, Reportes o informes de Circulación, Registro y control de usuarios y un Catálogo de Acceso Público en Línea (OPAC, por sus siglas en inglés) que es el acceso principal de los usuarios al sistema. Cada uno de estos subsistemas responde a un tipo de demanda de los usuarios a la Biblioteca de la Universidad y ha sido adaptado para satisfacer las mismas de manera exitosa, mejorando algunas funcionalidades del sistema base y añadiéndole nuevas.

**Adquisición** es el módulo que permite el control de la obtención de materiales bibliográficos dando inicio al flujo de información en el sistema, esta es la primera etapa en la descripción de los materiales permitiendo la gestión de los mismos, administrar sugerencias, gestionar fondos y presupuestos, así como el resto de los servicios referentes a los materiales que se adquieren. Se han eliminado las funcionalidades de compra venta de materiales dado que no se efectúan dichas

acciones dentro del centro.

Una vez registrados los materiales, se manejan las funciones de **Catalogación** de registros bibliográficos y de autoridades, para ello utiliza el formato de descripción bibliográfica MARC21 pudiendo ser configurado para utilizar un subconjunto de este formato; este subsistema incluye las funcionalidades más importantes en cuanto a la descripción bibliográfica de los materiales.

Una vez que los materiales han sido descritos, pueden estar a disposición de los usuarios y ser prestados, para llevar a cabo esta función y controlar los ejemplares en la Biblioteca se desarrolla el módulo de **Circulación**, que permite además la gestión de usuarios morosos y la devolución de ejemplares, toda esta información puede ser revisada y/o controlada mediante el módulo **Informes de Circulación**.

Para facilitar que los usuarios puedan trasladar ejemplares mediante préstamos se utiliza el módulo de **Peticiones** que permite realizar solicitudes para el traslado de los mismos desde el fondo de la Biblioteca hacia la sala de lectura.

Para brindar información de los procesos y servicios de la Biblioteca se utiliza el subsistema de **Reportes** el cual resulta útil para la toma de decisiones.

Se lleva a cabo además el **Registro y control de usuarios** lo que permite realizar la administración de los usuarios del sistema ya sea para crear, modificar o eliminar un usuario.

Finalmente el **Catálogo de Acceso Público en Línea** brinda a los usuarios acceso a los servicios de la Biblioteca y almacenamiento de información personalizada.

Cada uno de estos módulos o subsistemas están a su vez constituidos por un conjunto de casos de uso que suman un total de sesenta y uno; dichos casos de uso brindan los servicios correspondientes a los procesos realizados en la biblioteca. De ellos, a los que se les concede mayor significación, se agrupan en los módulos de Catalogación y Circulación, siendo estos un total de siete casos de uso arquitectónicamente significativos, en el subsistema de Catalogación se encuentran los casos de uso: *Gestionar ejemplares*, *Gestionar registros bibliográficos*, *Realizar búsqueda en catálogos* que tiene como caso de uso extendido *Realizar búsqueda por clasificación DEWEY*. En el módulo de Circulación están los casos de uso: *Registrar devoluciones* y *Realizar préstamo externo* que incluye el caso de uso *Negar préstamo*.

Después de un análisis previo, se puede decir que los casos de uso *Gestionar ejemplares*, *Gestionar registros bibliográficos* y *Negar préstamo* son actividades o funciones que debe desarrollar solamente una persona autorizada, dígase un bibliotecario que debe tener acceso a la interfaz administrativa, de modo que la utilización de estos en alguno de los demás sistemas de la solución no cumpliría objetivo dado que solo los bibliotecólogos pueden realizar esta tarea y basta con que se realicen en el SIGB, de ahí que no serían útiles para la integración que se pretende lograr; pero aún se debe valorar si el resto de los casos de uso considerados arquitectónicamente significativos constituyen una necesidad actual para la integración.

Dado que el subsistema o módulo para el catálogo en línea (OPAC) es accedido vía web desde el Portal de la dirección de Información, y su función principal es precisamente la búsqueda de materiales o ejemplares en el catálogo, los casos de uso *Realizar búsqueda en catálogos* y *Realizar búsqueda por clasificación DEWEY*, no constituyen una necesidad para el mismo, sino que su utilización supone una duplicidad de funcionalidades que resultaría tediosa e innecesaria; de igual manera el Repositorio cuenta con su propia interfaz de usuario relacionada con las búsquedas de los materiales que este almacena, los que a excepción de las tesis no han de estar recogidos en el catálogo, y teniendo en cuenta que en el momento actual no es de interés para el mismo la búsqueda o exposición de materiales que no estén almacenados en su base de datos se entiende que estos casos de uso tampoco le son necesarios.

Asimismo los casos de uso *Registrar devoluciones* y *Realizar préstamo externo* no constituyen una necesidad actual ni para el Sistema de Gestión de la Dirección de Información ni para el Repositorio Institucional; sino que las principales necesidades de estos sistemas van enfocadas al acceso a la información que les pueda ofrecer el SIGB.

## **2.2 Funcionalidades necesarias para la integración.**

Haciendo un análisis de los casos de uso arquitectónicamente significativos, como punto de partida para la integración funcional entre los sistemas que forman la actual Solución para Bibliotecas, se pudo concluir que estos no constituyen una necesidad actual para la integración, o sea, que el Portal de la Dirección de Información y/o el Repositorio Institucional no tienen necesidad de estas funcionalidades.

El portal busca información relacionada con los reportes y estadísticas que este publica, así como de las nuevas adquisiciones de materiales registradas mediante la intranet del SIGB, o sea, los materiales que están en proceso de catalogación que luego serán publicados en el catálogo. El Repositorio institucional por su parte utilizaría la información de las nuevas adquisiciones para obtener las que se corresponden con Tesis de Diploma, Maestrías o Doctorado, pues serían sus futuras adquisiciones y publicaciones. En función de un sistema de autenticación única, también les es necesaria la información asociada al usuario autenticado en el mismo.

Se puede decir entonces, que las principales necesidades para la integración en este momento, se corresponden con la información a la que solo se tiene acceso desde la Intranet de Administración del SIGB en forma de reportes, esta será útil en la publicación de las estadísticas del Portal de la Dirección de Información, futuras adquisiciones del Repositorio Institucional y/o publicaciones en el catálogo de la biblioteca. Esta información hasta ahora solo está en conocimiento de los trabajadores de la biblioteca con acceso a la interfaz administrativa del SIGB, su publicación como servicio web hará que esta sea accesible al resto de la comunidad universitaria y permitirá reportes más completos en cuanto a las estadísticas de la Dirección de

Información.

## **2.3 Descripción del módulo de Servicios Web**

El módulo de Servicios Web, para lograr una integración funcional entre los sistemas que forman el Sistema de Gestión Bibliotecaria de la universidad, toma como punto de partida el Sistema Integrado de Gestión Bibliotecaria como parte fundamental de la solución, y de este las funcionalidades implementadas en la intranet administrativa, con el fin de ofrecer la información que hasta este momento solo está en conocimiento de los administradores del sistema o bibliotecarios con acceso al mismo. Aún así, todos los servicios no se enfocan solo a esta área, sino que también se ofrecen algunos datos de usuario que se encuentran almacenados en las tablas correspondientes al catálogo de acceso en línea.

Este módulo se nutre de la información contenida en la base de datos y cada servicio es ofrecido de manera independiente, la información que se ofrece se puede dividir en información de estadísticas, información de usuarios e información de últimas adquisiciones.

### **2.3.1 Concepción inicial del módulo de Servicios Web**

El módulo de servicios web se concibe con la idea de contar con funcionalidades e información común entre los sistemas de la actual Solución para Bibliotecas, no afectar el diseño de los sistemas que la forman y evitar la duplicidad de información, respetando la independencia de cada uno de los sistemas en cuanto a sus datos y teniendo en cuenta que están desarrollados todos en tecnologías diferentes.

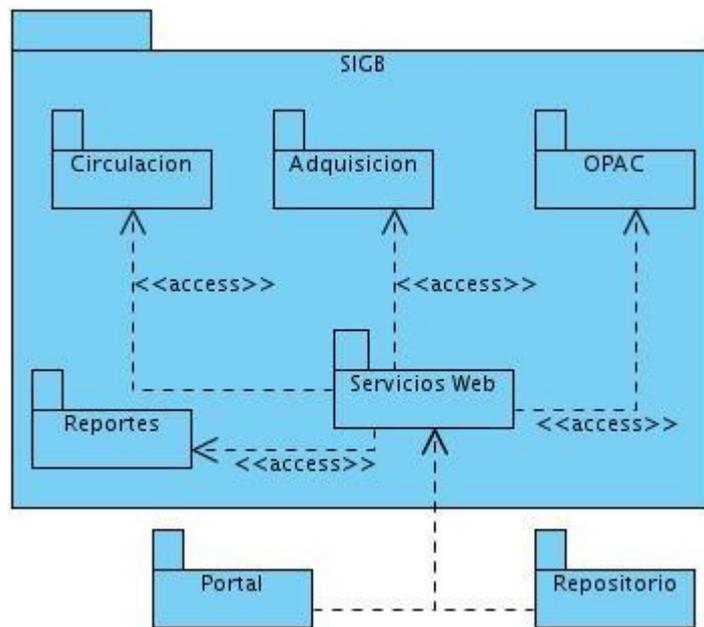
La idea inicial suponía un trabajo sobre los tres sistemas publicando funcionalidades y servicios de todos ellos a la vez, ya sean aquellos que solo se realiza en uno de estos sistemas -o sea, los que los hacían diferentes entre sí -de manera que se pudiera realizar cualquier función desde cualquiera de las componentes de la solución e incluso desde cualquier sistema del entorno (intranet, portal del graduado, sitio de la FEU) o determinando las funcionalidades comunes a los tres pero que cada uno realiza de manera particular y haciéndolas generales en todos –un ejemplo de estos son las búsquedas, se pretendía que desde el repositorio se pudiera realizar búsquedas en el catálogo del SIGB y viceversa, y desde el portal o cualquier otro sistema igualmente se hicieran búsquedas generales - . Estas son ideas válidas pero sugieren un trabajo bastante profundo que requiere esfuerzo y recursos, esencialmente recursos humanos y un estudio profundo y detallado de los sistemas en cuestión, además de requerir cambios en el diseño de las interfaces y/o arquitectura de dichos sistemas. De manera que se redujo el alcance del trabajo a tomar uno de los componentes de la solución que tuviese funciones que los demás podrían necesitar y partiendo de ese desarrollar la integración, esto podría servir a su vez como base para alcanzar en el futuro lo que se pensó inicialmente.

Actualmente el SIGB no cuenta con un módulo que ofrezca servicios web que faciliten la información de este a otros sistemas. La intranet para administración de los servicios que ofrece la biblioteca se encuentra aislada del resto de los sistemas que forman la solución, solo manteniendo relación con el catálogo en línea (OPAC), de quien recibe peticiones y le envía respuestas, aún así no le brinda a este ninguna información que no sea las respuestas a las peticiones recibidas por los usuarios del mismo. Más del 60% de la información que se tiene almacenada desde dicha intranet administrativa solo puede ser accedida desde su propia interfaz y no la comparte con otro sistema. El módulo para servicios web es una opción para facilitar el compartir parte de esta información con el resto de los sistemas de la biblioteca, facilitando la comunicación de estos con la intranet y rompiendo un tanto las barreras que hasta aquí existen. Los servicios web pondrán al alcance del Portal y el Repositorio, la información que han tenido vedada, y les permitirá a los mismos la publicación de información de interés para sus usuarios, asimismo hará más eficientes los reportes que emite el portal sobre los servicios que ofrece la Dirección de Información.

### **2.3.2 Dependencias y Relaciones con otros módulos**

La estructura modular del SIGB permite que sea bastante sencilla la incorporación de un nuevo módulo. Una vez desarrollados los servicios web para la integración, estos se acoplarán bajo la estructura modular del SIGB a fin de aumentar la interoperabilidad entre este y otros sistemas, esencialmente aquellos que forman el actual Sistema de Gestión Bibliotecaria.

El módulo de publicación de servicios web comparte información del SIGB al resto de los sistemas que forman la actual Solución para Bibliotecas, en función de hacer esto de la mejor manera posible este interactúa con otros de los módulos de sistema en cuestión. Una vez obtenida la información es ofrecida a las interfaces web del Portal de la Dirección de Información y al Repositorio Institucional. Estas relaciones quedan descritas a continuación:

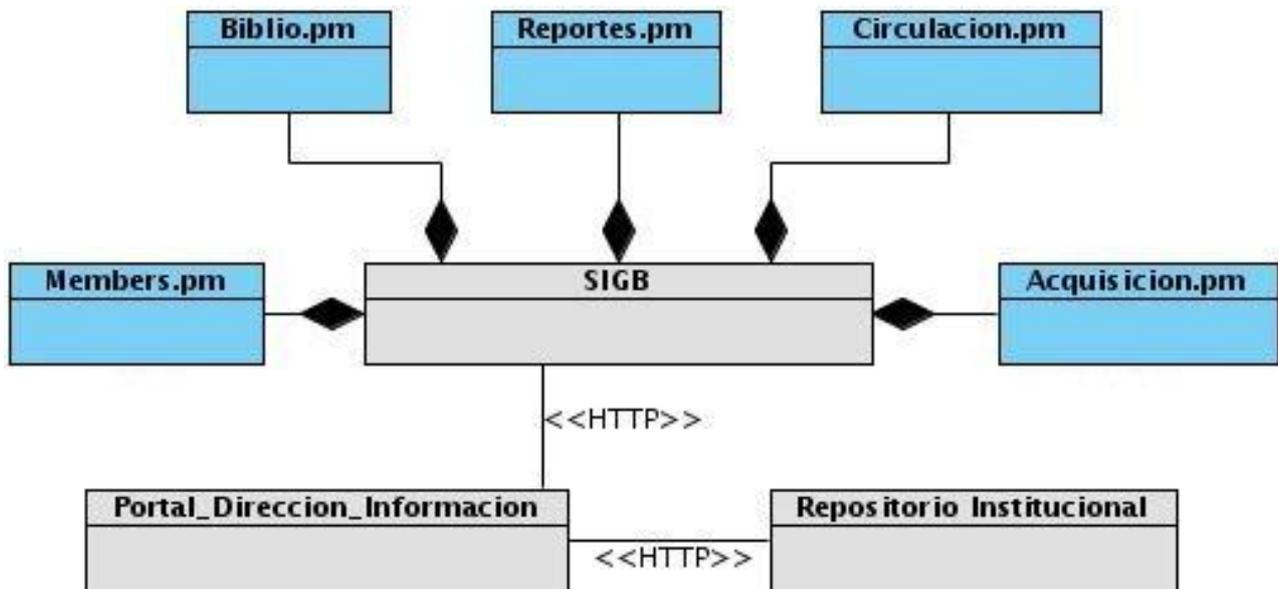


**Figura 1. Diagrama de paquetes con dependencias del módulo de servicios web**

Los módulos de Reportes, Catalogación, Circulación y Opac, son los que ofrecen la información que se pretende publicar, de ahí que exista una relación entre estos y el módulo para servicios web; aún así, este funciona de manera independiente como una capa abstracta, sin ejecutar métodos o algoritmos de los demás módulos sino que accede directamente a la información guardada en la base de datos por mediación de dichos módulos, o sea, la información que desde la interfaz de estos se almacena en la base de datos; y aunque la información que ofrecen los servicios web es también brindada por ellos, la manera en que lo hace y la manera en que se tendrá acceso a la misma es diferente. Entonces la relación descrita no implica una dependencia funcional de los servicios web ya que no es relevante que el resto de los módulos funcionen o no, sino que se tenga la información necesaria en la base de datos para que este pueda acceder a ella y ofrecerla a quien pueda interesarle, sin utilizar funcionalidades de ningún otro.

### 2.3.3 Modelo de Dominio

En la actualidad no existe una relación funcional entre los sistemas que forman la Solución para la Biblioteca UCI, sino que estos sistemas están vinculados desde el portal de la Dirección de Información mediante una conexión HTTP pero sus funciones están completamente aisladas entre sí, esta relación se define conceptualmente en el siguiente modelo de dominio.



**Figura 2. Modelo de Dominio del Sistema de Gestión Bibliotecaria**

El modelo de dominio permite de manera visual mostrar los principales conceptos que se manejan en el dominio del problema. Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes de software. El modelo desarrollado no se trata de un conjunto de diagramas que describen clases de software u objetos de software con responsabilidades, sino que pueden considerarse como un diccionario visual de las abstracciones relevantes, vocabulario e información del dominio. Aprovechando las posibilidades que brinda UML para representar conceptos, el modelo de dominio se representa en forma de diagramas de clases donde figuran los principales conceptos y roles del sistema en cuestión.

Para una mejor comprensión del diagrama de clases del modelo de dominio se proporciona una descripción de las clases que se representan en el mismo.

**Repositorio Institucional, Portal\_Direccion\_Informacion y SIGB:** son los componentes del Sistema de Gestión Bibliotecaria empleado en la UCI, estos sistemas no tienen relación funcional sino que el Repositorio y el SIGB son accedidos vía HTTP desde el Sistema de Gestión de la Dirección de Información o Portal de la Dirección de Información, de manera que su única relación es un vínculo que tienen en dicho portal. El **SIGB** (Sistema Integrado de Gestión Bibliotecaria) es el componente que más funcionalidades y servicios ofrece y el de mayor responsabilidad dentro de la Solución, está compuesto por seis módulos que representan procesos dentro de la biblioteca, de ellos se representan cuatro en este modelo (Opac, Reportes, Circulación y Adquisición) mediante las clases Members.pm, Biblio.pm, Reportes.pm, Circulacion.pm y Adquisicion.pm.

**Members.pm:** es una componente del catálogo en línea (Opac) que representa a los usuarios registrados en el sistema; o sea, los miembros asociados a la biblioteca.

**Biblio.pm:** Esta clase es parte esencial del SIGB y representa los registros bibliográficos, manejando el acceso y control sobre los mismos.

**Reportes.pm:** esta es una componente del módulo de Reportes que es donde se controlan las estadísticas del SIGB y otros informes.

**Circulacion.pm:** indicador del módulo de circulación que funciona como controlador de los procesos relacionados con los movimientos de los materiales dentro y fuera de la biblioteca.

**Adquisicion.pm:** representación del módulo de adquisición que controla los nuevos materiales adquiridos por la biblioteca mientras esperan ser catalogados.

### 2.3.4 Levantamiento de requisitos

La ingeniería de requisitos comprende todas las tareas relacionadas con la determinación de las necesidades o de las condiciones a satisfacer para un software nuevo o modificado, tomando en cuenta los diversos requisitos de los inversores o clientes que puedan entrar en conflicto entre ellos; el propósito es hacer que los requisitos alcancen un estado óptimo antes de alcanzar la fase de diseño en el proyecto. Los buenos requisitos deben ser medibles, comprobables, sin ambigüedades ni contradicciones.

Desde el punto de vista conceptual las actividades a realizar con estos requisitos son cinco: obtener los requisitos para conocer cuáles son los deseos del cliente, analizar los requisitos para determinar y corregir los errores comunicativos durante la captura de los mismos y transformarlos en condiciones apropiadas para ser tratadas en el diseño, documentar los requisitos, verificarlos o comprobar el correcto funcionamiento de estos en la aplicación y validar los requisitos a fin de que una vez implementados se correspondan con lo que inicialmente se pretendía.

Partiendo de las necesidades del Sistema de Gestión de la Dirección de Información y el Repositorio Institucional en relación con las funcionalidades ofrecidas por el SIGB se pudo recoger la siguiente lista de requisitos:

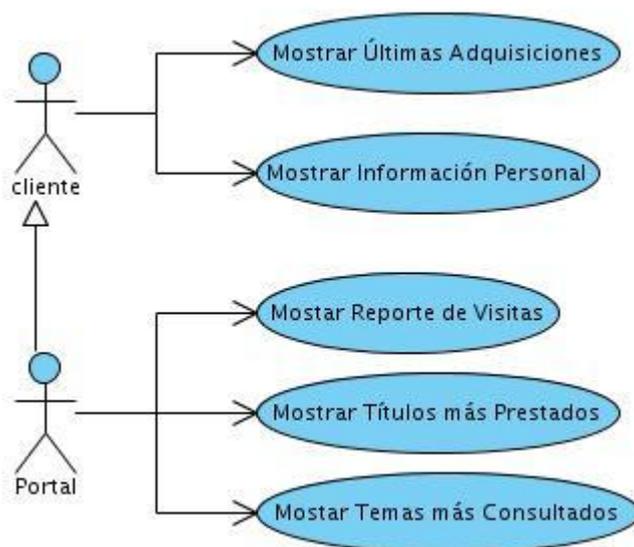
1. **Mostrar ejemplares adquiridos recientemente:** pretende mostrar la información referente a los materiales adquiridos por la biblioteca que aún no se encuentran en el catálogo dado que están pendientes de catalogar para ser accedidos por el público.
2. **Visualizar resumen personal dado un usuario:** dado un usuario determinado, devolver los materiales que tiene prestado el mismo.
3. **Visualizar sanciones asociadas a un usuario:** mostrar las sanciones referentes a un determinado usuario.
4. **Mostrar cantidad de visitas realizadas en el día:** se debe devolver cuántos usuarios han accedido al SIGB en las últimas 24 horas.
5. **Mostrar cantidad de visitas realizadas en la semana:** devolver cuántos usuarios han accedido al SIGB en la última semana.

6. **Mostrar cantidad de visitas realizadas en el mes:** mostrar cuántos usuarios han accedido al SIGB en el transcurso correspondiente a un mes de anterioridad.
7. **Mostrar cantidad de visitas realizadas en el año:** cuántas visitas ha recibido el SIGB en los últimos 365 días o 366 en caso de haber sido un año bisiesto.
8. **Listar títulos más prestados:** brindará el conjunto de los títulos que más han sido solicitados por los usuarios y más se han prestado en la biblioteca sin distinción entre tipos de préstamos (internos o externos).
9. **Mostrar temas más consultados:** muestra el conjunto de temas relacionados a los préstamos y consultas de los usuarios que más se han solicitado.

### 2.3.5 Definición de los casos de uso

Una de las técnicas para documentar posibles requisitos son los casos de uso, donde se grafican las relaciones del sistema con los usuarios u otros sistemas; en otras palabras, es la técnica más efectiva y a la vez la más sencilla para modelar los requisitos del sistema desde el punto de vista del usuario. El objetivo es mejorar la comunicación entre usuarios y desarrolladores mediante las pruebas tempranas de prototipos, para minimizar cambios hacia el final del proyecto y reducir los costes finales. Los casos de uso se utilizan para correlacionar cada escenario con los requisitos que completa.

La lista de requisitos funcionales obtenida anteriormente permite modelar el sistema en términos de casos de uso, los mismos agrupándose en cinco casos de uso de la manera siguiente:



**Figura 3. Casos de Uso del módulo de Servicios Web.**

1. **Caso de Uso Mostrar Últimas Adquisiciones:** Recoge el requisito funcional de mostrar ejemplares adquiridos recientemente.

2. **Caso de Uso Mostrar Información Personal:** Recoge los requisitos funcionales de visualizar resumen personal y visualizar sanciones dado un usuario.
3. **Caso de Uso Mostrar Reporte de visitas:** Recoge los requisitos funcionales de mostrar cantidad de visitas realizadas en el día, semana, mes y año respectivamente.
4. **Caso de Uso Mostrar Títulos más prestados:** Recoge el requisito funcional de mostrar títulos más prestados.
5. **Caso de uso Mostrar Temas más consultados:** Recoge el requisito funcional de mostrar temas más consultados.

Los casos de uso sirven para describir como el sistema funciona actualmente o como se desea que funcione el mismo; estos no son una aproximación a la orientación a objetos sino una forma de modelar procesos. Cada caso de uso se documenta con una descripción del escenario.

**Tabla 1. Descripción del CU Mostrar Últimas Adquisiciones.**

Sección del caso de Uso	Descripción
Nombre	Mostrar últimas adquisiciones.
Breve Descripción	Con el objetivo de que se puedan publicar las futuras publicaciones en el catálogo de la biblioteca de la universidad y/o el Repositorio Institucional, el servicio devolverá las últimas adquisiciones de materiales de la biblioteca UCI.
Precondiciones	Existen adquisiciones en la base de datos.
Post-Condiciones	Se devuelven las últimas adquisiciones de la biblioteca UCI.

**Tabla 2. Descripción del CU Mostrar Información Personal**

Sección del caso de Uso	Descripción
Nombre	Mostrar Información Personal
Breve Descripción	Tiene como objetivo darle a conocer a los usuarios autenticados información de su perfil mediante una petición al sistema acerca de los títulos que tiene prestados y las sanciones de las que pueda ser objeto.
Precondiciones	El usuario está autenticado.
Post-Condiciones	La información debe estar actualizada y en correspondencia con el

	usuario en cuestión.
--	----------------------

**Tabla 3. Descripción del CU Mostrar Reporte de Visitas**

Sección del caso de Uso	Descripción
Nombre	Mostrar Reporte de Visitas
Breve Descripción	El Caso de uso permite conocer las estadísticas de las visitas al SIGB en un número correspondiente a un día, semana, mes o año desde la fecha actual hacia atrás.
Precondiciones	No requiere precondiciones.
Post-Condiciones	Se devuelven las visitas divididas por cantidad: semanal, mensual y anual.

**Tabla 4. Descripción del CU Mostrar Títulos más prestados.**

Sección del caso de Uso	Descripción
Nombre	Mostrar Títulos más prestados.
Breve Descripción	Se pretende obtener estadísticas sobre títulos más prestados devolviendo los títulos de materiales y/o ejemplares más prestados por la biblioteca de la universidad.
Precondiciones	Existen préstamos registrados en la base de datos.
Post-Condiciones	Se obtiene un conjunto de títulos de materiales existente en la biblioteca.

**Tabla 5. Descripción del CU Mostrar Temas más consultados**

Sección del caso de Uso	Descripción
Nombre	Mostrar Temas más consultados
Breve Descripción	El sistema devolverá los temas correspondientes a los materiales más consultados por los usuarios.
Precondiciones	Existen préstamos registrados en la base de datos.
Post-Condiciones	Se obtienen temas relacionados con los materiales más

	consultados.
--	--------------

### 2.3.6 Actores del Sistema

Un actor es cualquiera que interactúe con el sistema siendo externo a él con el fin de utilizar los servicios que este ofrece; estos son representaciones de los usuarios que intercambian información con el sistema y suelen corresponderse con trabajadores o actores del negocio. En otros casos, un actor sobre los servicios web es cualquier sistema o aplicación que acceda a la interfaz de servicios web para hacerle una petición determinada; o sea, todo aquel sistema que utilice la información que este módulo ofrece.

Dado que este módulo está enfocado principalmente a la integración en la Solución Integral para Bibliotecas se conciben inicialmente como actores de este sistema al Sistema de Gestión de la Dirección de Información o Portal de la Dirección de Información y el Repositorio Institucional, que son las componentes que utilizarán la información que se publica en el mismo y son especificados a continuación:

**Tabla 6. Especificación de Actores del Sistema.**

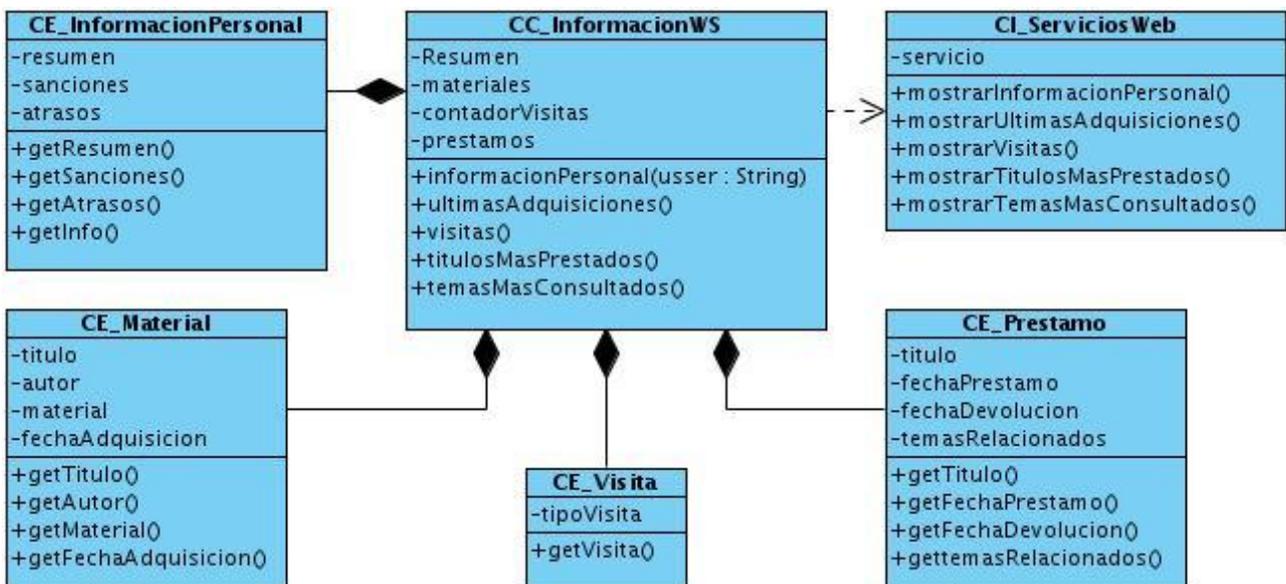
<b>Actor</b>	<b>Justificación</b>
Sistema de Gestión de la Dirección de Información.	Sistema que consulta toda la información que se publica con el fin de generar reportes de estadísticas y publicar información novedosa. Tiene acceso a todas las funcionalidades del sistema y las utiliza de manera continuada.
Repositorio Institucional.	Sistema que consulta información relacionada con los préstamos y sanciones asociados a un usuario determinado y a las últimas adquisiciones que hacen referencias a trabajos de tesis para publicar información novedosa en su interfaz.
Cliente	Es una generalización de los sistemas que pueden acceder a los servicios web y pueden realizar las operaciones de consultar últimas adquisiciones y la información asociada a un usuario determinado.

### 2.3.7 Diagrama de Clases

Durante la fase de elaboración, se realiza el diseño del módulo para servicios web del Sistema de

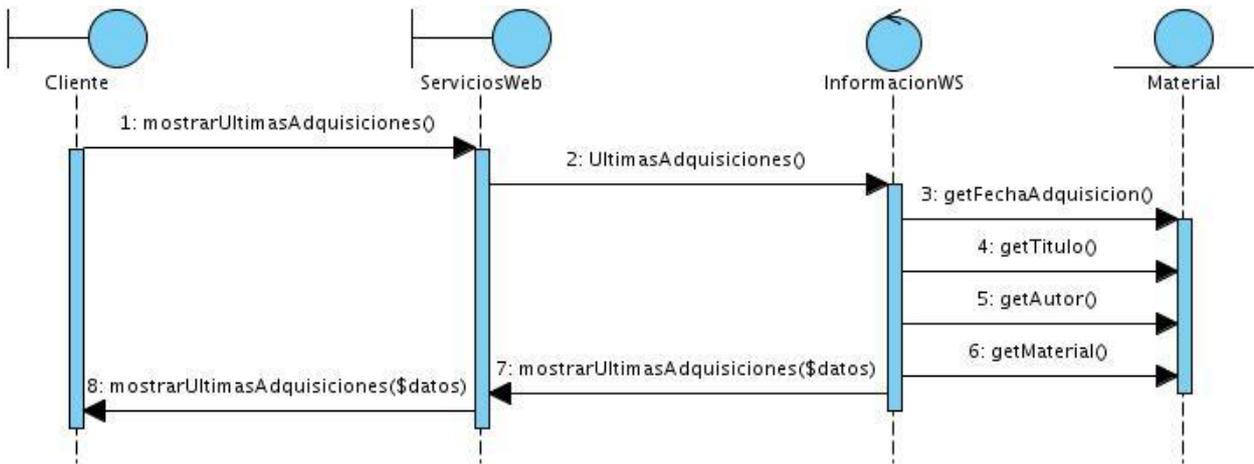
Gestión Bibliotecaria, en dicho módulo se conciben seis clases distribuidas en entidades, controladoras e interfaces y teniendo en cuenta el patrón de arquitectura Modelo-Vista-Controladora. Este patrón es capaz de separar los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El modelo es quien modela los datos y el procesamiento detrás de los procesos de negocios, se encapsulan los datos y el comportamiento que son independientes de la presentación; el controlador actúa como intermediario entre el modelo, la vista, y cualquier otro recurso necesario para generar una página, y en la vista se despliega la información en dependencia del tipo de cliente. Para procesar los datos, el controlador recibe una petición y decide quién la lleva a cabo en el modelo, una vez que el modelo termina sus operaciones devuelve el flujo al controlador y este envía el resultado a la vista.

Se utiliza una interfaz para publicar los servicios, una controladora para centralizar la información que se ofrece y controlar el acceso a la base de datos y las entidades para el modelo de dichos datos. De esta manera el Modelo de Clases del diseño queda como se muestra a continuación:

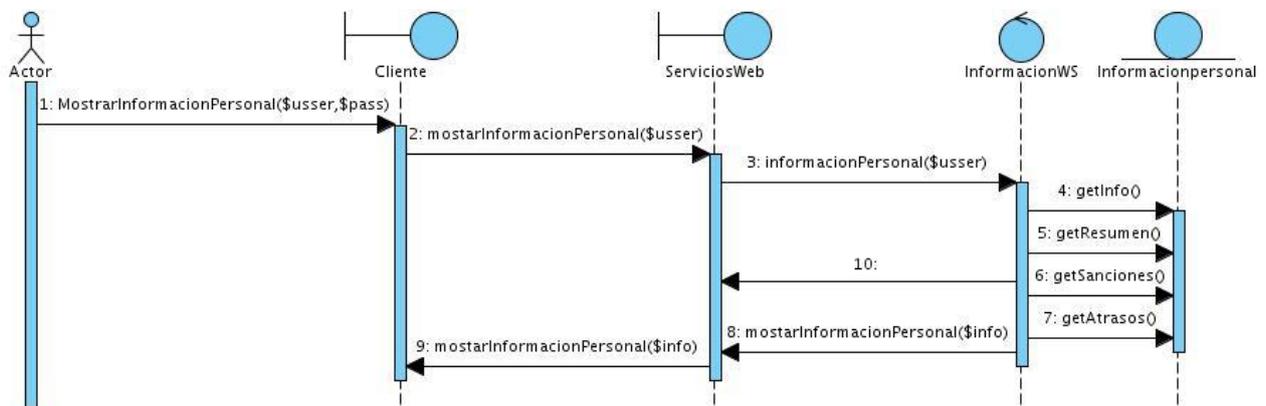


**Figura 4. Modelo de Clases del módulo de Servicios Web.**

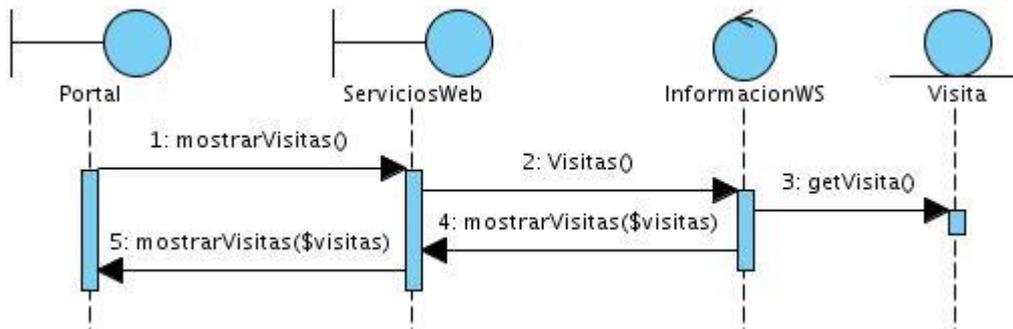
Estas clases interactúan en dependencia de las peticiones realizadas con las secuencias que se muestran a continuación:



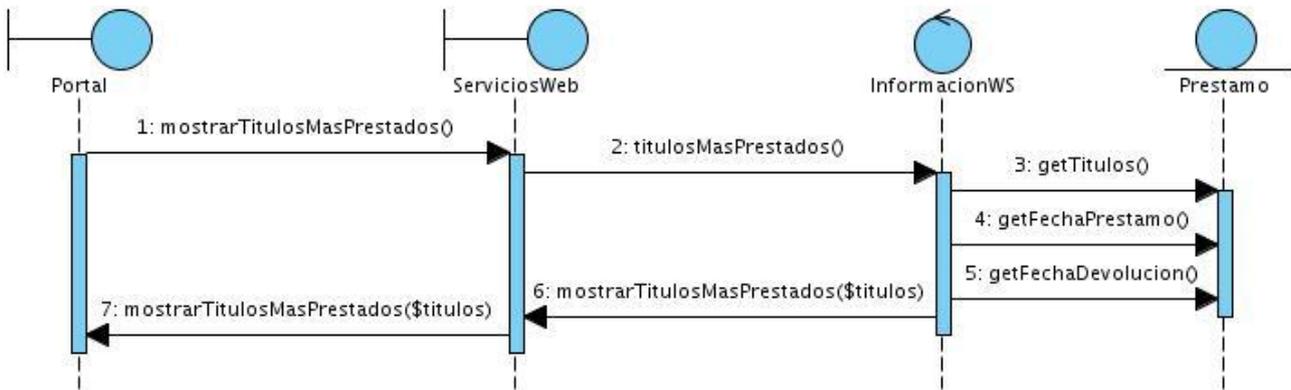
**Figura 5. Diagrama de Secuencia del CU\_Mostrar últimas adquisiciones.**



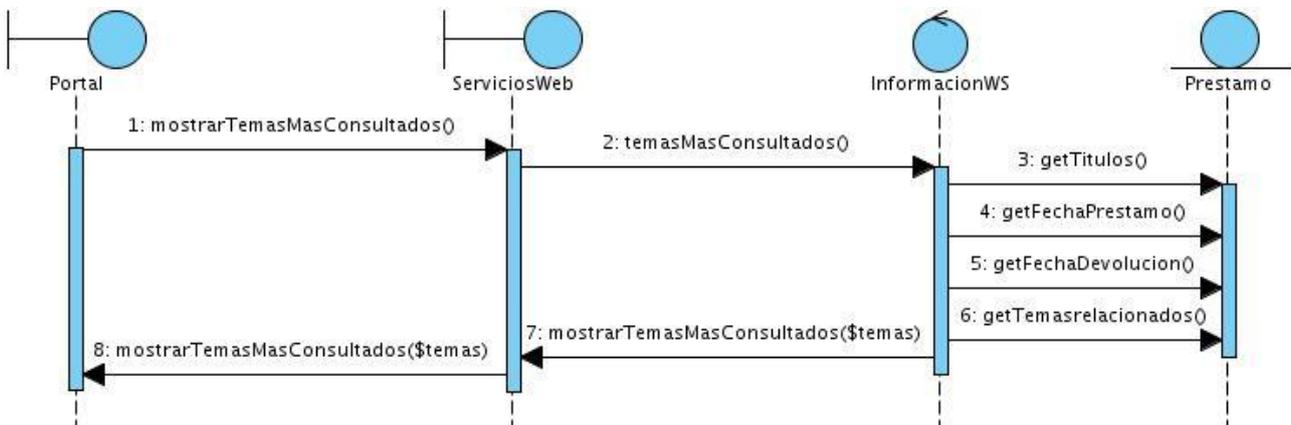
**Figura 6. Diagrama de Secuencia del CU\_ Información Personal.**



**Figura 7. Diagrama de Secuencia del CU\_Mostrar Reporte de Visitas.**



**Figura 8. Diagrama de Secuencia del CU\_ Mostrar Títulos más prestados.**



**Figura 9. Diagrama de Secuencia del CU\_ Mostrar Temas más consultados.**

### 2.3.7.2 Módulos y Clases utilizadas

El lenguaje de programación Perl incluye módulos que se encuentran ya desarrollados que ayudan en la implementación y publicación de servicios web, a continuación se muestran los que son utilizados en la elaboración del presente módulo de Servicios web:

- **WSDL::Generator:** Es un módulo Perl desarrollado por Pierre Denis para crear archivos de descripción de servicios web (wsdl) automáticamente a partir de módulos Perl expuestos como servicios web. A continuación se muestran las dependencias del mismo.

Module		Pass rate	Pass	Unknown	Fail	NA
- WSDL::Generator	Q 兼 ✓ ⇄ ☒		20	0	0	0
- Class::Hook	Q 兼 ✓ ⇄ ☒		21	0	0	0
Test::Simple	Q 兼 ✓ ⇄					Core module
Time::HiRes	Q 兼 ✓ ⇄					Core module
<b>Chance of all tests passing</b>			100% (note this doesn't mean much)			

The chance of success is calculated by multiplying the probabilities of the module and all its dependencies passing their tests.

Key : Q = Search 兼 = Bug reports ✓ = Test results ⇄ = Reverse dependencies ☒ = OS/Perl results matrix

**Figura 10. Dependencias del modulo perl WSDL::Generator.**

- **SOAP::Lite**: es un conjunto de módulos con una interfaz limpia y liviana para el protocolo SOAP tanto en el cliente como en el servidor, actualmente es el kit de desarrollo de servicios web más difundido de perl y el más utilizado; provee clases para implementar funcionalidades de un cliente SOAP, varios servidores, soporte a datos y muchas otras tareas. Este tiene dependencias de otros módulos y clases como muestra la imagen siguiente:

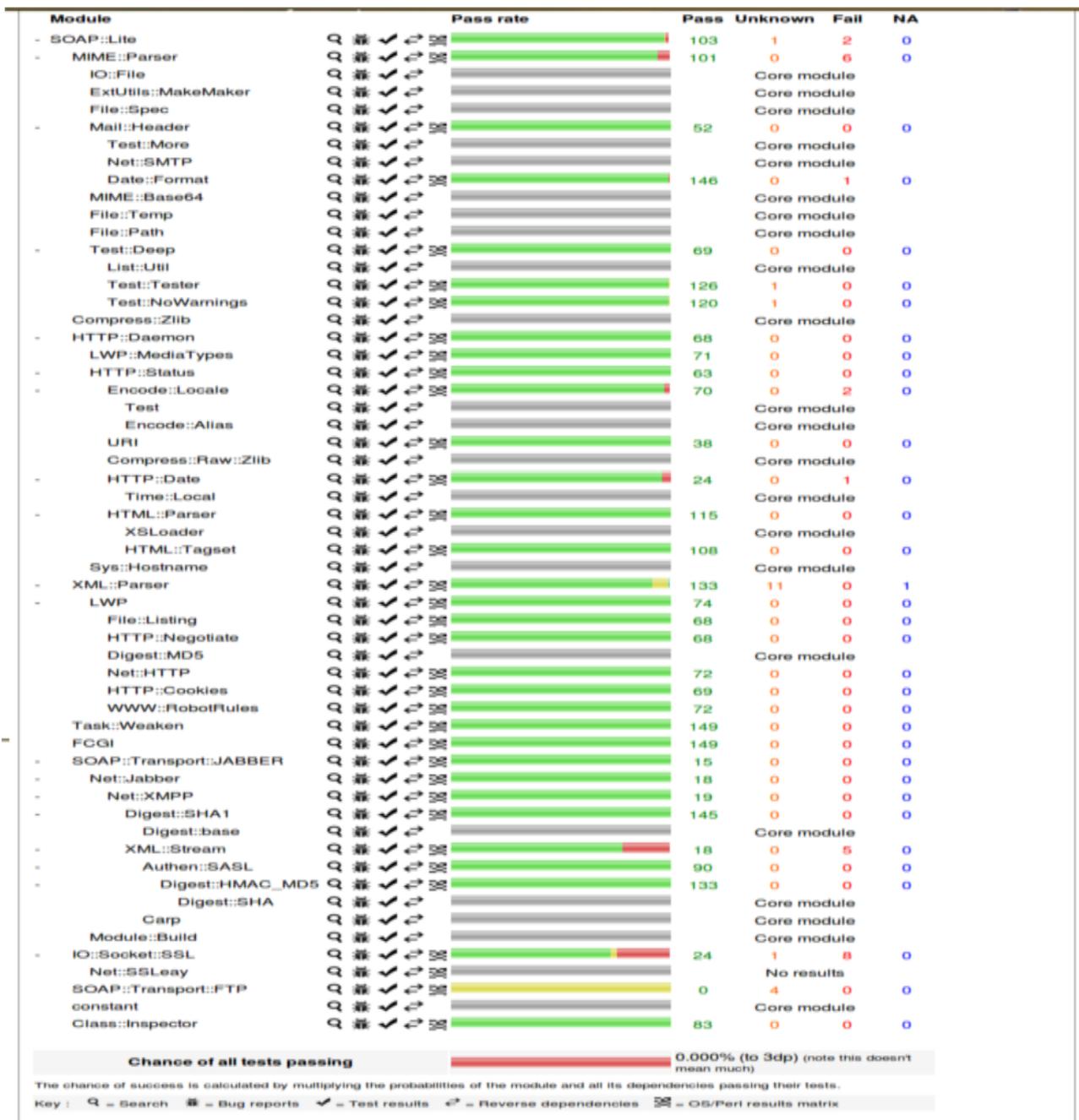


Figura 11. Dependencias del modulo perl SOAP::Lite.

### 2.3.8 Modelo de Datos

Para que sean cumplidos satisfactoriamente los objetivos propuestos y dar cumplimiento a los requisitos funcionales es necesario un correcto manejo de los datos; de ahí la importancia que se le concede el modelo de datos, pues este muestra cómo se organizan los datos y cuál es la información que se muestra indicando si es o no la correcta.

Las tablas expuestas a continuación forman parte de la base de datos del SIGB incluidas en su modelo de datos, solo se seleccionan aquellas que contienen los datos requeridos por el módulo de servicios web.



de servicios web y publicación de los mismos, así como con los estándares para la creación de un fichero wsdl para la descripción de los servicios. El estilo arquitectónico Modelo-Vista-Controladora proporcionará robustez al módulo garantizando que posibles cambios afecten la menor cantidad de componentes posibles. De manera que los modelos hasta aquí obtenidos responden a las necesidades de integración entre los sistemas que forman parte del Sistema de Gestión Bibliotecaria y supone un marco conceptual mediante el cual se puede simplificar la creación y mantenimiento de aplicaciones; en otros términos, se alcanzará una integración bastante eficiente al SIGB si se siguen las pautas que se plantean en este capítulo y por consiguiente se obtendrá una integración entre los subsistemas de la Solución para Bibliotecas sentando las bases para la incorporación de futuros servicios.

## Capítulo 3 Implementación y Prueba

Durante el proceso de desarrollo de software los flujos de trabajo de implementación y prueba son determinantes a fin de lograr un producto bien acabado y de calidad, ambos tienen su mayor peso en la fase de construcción. El primero se ve más enfocado al código de la aplicación además de hacer una representación de la implementación de las clases del diseño en términos de componentes y como estos se organizan de acuerdo con los nodos específicos del modelo de despliegue. El segundo tiene su enfoque en la calidad, determinando los errores que pueda tener el sistema para corregirlos y lograr un funcionamiento óptimo.

Para dar cumplimiento al objetivo general de la investigación, la misma se realiza basada en los servicios web como una solución universal, usando el módulo SOAP::Lite que permite implementar servicios web en Perl y el protocolo SOAP.

El presente capítulo hace una descripción de las funcionalidades implementadas en el módulo de Servicios web, y su organización una vez desplegado el mismo, se realiza un análisis de los casos de prueba validando la aceptación de la solución propuesta para la integración funcional entre los sistemas de la actual Solución para Bibliotecas.

### 3.1 Diagrama de despliegue

El módulo de Servicios web expone una interfaz de comunicación para que sistemas externos al SIGB accedan a la información contenida en el mismo, esta se basa en el paradigma cliente servidor con el protocolo SOAP. A continuación se mostrará el despliegue en nodos físicos identificando los diferentes nodos y la forma de conexión entre ellos. Esta implementación logra la exposición de funcionalidades del SIGB en forma de servicios.

*El diagrama de despliegue se utiliza para modelar la configuración de los elementos de procesado en tiempo de ejecución y de los componentes, procesos y objetos de software que viven en ellos. Se modelan los nodos físicos y las asociaciones de comunicación que existe entre ellos.* (Laurencio Pérez 2009)

El diagrama se utiliza para capturar los elementos de configuración del procesamiento de la información y las conexiones que existen entre estos elementos. Está formado por nodos, dispositivos y conectores. Los nodos son elementos de procesamiento con al menos un procesador, memoria y posiblemente otros dispositivos; los dispositivos a su vez son nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela, y los conectores expresan el protocolo o conector utilizado entre los elementos del diagrama.



**Figura 13. Diagrama de Despliegue.**

El diagrama de despliegue muestra la topología de hardware sobre la cual se ejecuta el sistema y muestra la configuración de los componentes de hardware, los procesos, los elementos de procesamiento y los objetos que existen en tiempo de ejecución.

### **Nodo PC\_Cliente**

Representa las computadoras que utilizarán los usuarios finales para interactuar con las aplicaciones del Sistema de Gestión Bibliotecaria. Establece relación con el servidor de aplicaciones a través del protocolo HTTP.

### **Nodo Servidor de Aplicaciones**

En este nodo estarán los script de la aplicación.

### **Nodo Servidor de Base de Datos**

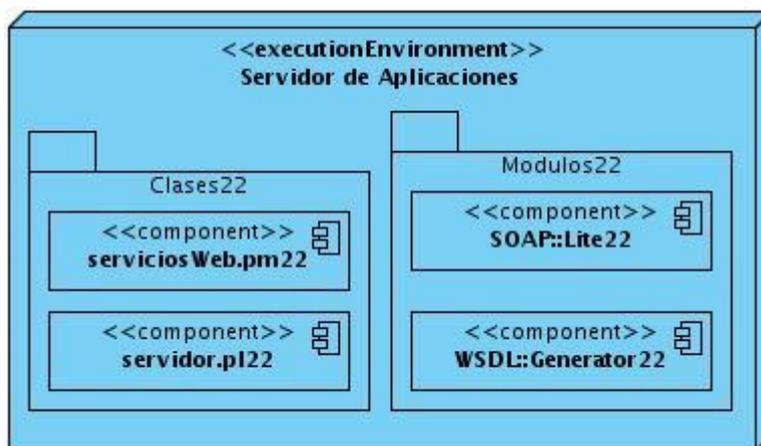
En este nodo se encuentra la base de datos que contiene toda la información del SIGB de la solución.

## **3.2 Diagrama de componentes**

Los modelos de implementación describen los componentes y la organización de acuerdo a los nodos así como las dependencias de compilación entre ellos.

*Los diagramas de componentes describen los elementos físicos del sistema y su relación, mostrando las dependencias lógicas entre componentes de software. El diagrama de componentes hace parte de la vista física de un sistema, la cual modela la estructura de implementación de la aplicación por sí misma, su organización en componentes y su despliegue en nodos de ejecución. La vista de implementación se representa con los diagramas de componentes. (Laurencio Pérez 2009)*

El modelo de implementación tiene trazas con las clases del diseño que implementa de manera que se representa en capas compuestas por subsistemas y componentes agrupados en paquetes para una mejor organización. El siguiente apartado describe los componentes más relevantes que implementan las clases relacionadas con los servicios web que ofrece el módulo con este fin en el SIGB.



**Figura 14. Diagrama de componentes.**

Los diagramas de componentes se utilizan para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los mismos. Su principal objetivo es mostrar la estructura de alto nivel del modelo de implementación, mostrando la organización y las dependencias lógicas entre un conjunto de componentes de software que pueden ser por ejemplo: librerías, código fuente, ficheros binarios o ejecutables.

### **Subsistema Modulos**

Contenedor físico de los módulos necesarios para el correcto funcionamiento de la aplicación y contiene los módulos siguientes:

#### **SOAP::Lite**

Módulo que permite la creación, publicación y consumo de servicios web en perl.

#### **WSDL::Generator**

Módulo para generar automáticamente el archivo wsdl para la descripción de servicios web.

### **Subsistema Clases**

Contenedor físico del compilado de cada una de las clases de la aplicación y contiene las clases siguientes:

#### **serviciosWeb.pm**

Clase controladora encargada de recoger toda la información del modelo que esté relacionada con los servicios web y hacerla llegar a la interfaz de publicación de los mismos.

#### **Servidor.pl**

Clase que funciona como interfaz para el acceso a los servicios web propuestos.

La clase serviciosWeb.pm que también es conocida por perl como un módulo de acceso a datos es la clase principal del subsistema o módulo para servicios web. Esta es la clase que permite implementar cada uno de los métodos que son expuestos en términos de servicios web, de

manera que invocando correctamente los métodos que esta implementa se logra tener acceso a toda la información publicada. Los métodos fueron desarrollados en correspondencia con los casos de uso diseñados para el sistema en función de dar cumplimiento a los requisitos web capturados en el flujo de trabajo de requisitos y se exponen a continuación:

- **ultimasAdquisiciones**: accediendo a las tablas `aqorders` y `biblio` permite conocer cuales son las últimas adquisiciones o los materiales que aún no se han catalogado, obteniendo de ellos el título y la fecha de entrada para con esta última determinar cuales de estas adquisiciones son más recientes.
- **prestamosUsuario**: recibiendo como parámetro de entrada el identificador de un determinado usuario, devuelve los títulos que tiene prestado el mismo así como las fechas de préstamos y devolución de los mismos.
- **sancionesUsuario**: con el identificador de un determinado usuario como variable de entrada devuelve las fechas de inicio y fin así como las causas de las sanciones de las que dicho usuario ha sido objeto.
- **visitas**: una llamada a esta función devuelve el número de visitas efectuadas al SIGB dividida en frecuencias correspondientes a la fecha actual, un rango de siete días, un mes, o un año cumplidos hasta hoy (fecha actual).
- **titulosMasPrestados**: teniendo en cuenta todos los préstamos realizados y registrados en el SIGB, determina cuantas veces se ha prestado un material con un título determinado y devuelve una lista de los títulos que han sido prestados y el número de veces que se ha prestado en orden descendente según la cantidad de préstamos efectuados de cada uno de ellos.
- **temasMasConsultados**: esta función hace un análisis de los títulos prestados y teniendo en cuenta los temas a los que estos reponen, devuelve los temas más consultados en función de dichos préstamos.

Una vez concluida la implementación de estas funciones, el próximo paso para la publicación de los servicios web es la creación de una interfaz mediante la utilización del módulo `SOAP::Lite` para exponer los mismos usando el servidor tipo `SOAP::Transpot::HTTP::Daemon`, y esto es precisamente lo que hace la clase `servidor.pl` (véase el código en el **Anexo 2**), a partir de este momento cualquier aplicación puede consumir el servicio si sabe como invocarlo. La llamada desde una consola con un cliente `perl` a la función **ultimasAdquisiciones** da como resultado la siguiente pantalla.

```
geany_run_script.sh
1: cuba
2: marti
3: opup
4: hjhj
5: jjj
6: aa
7: jkj
8: sugerencia de compra
9: 3433
10: marti
11: 1111
12: sds
13: sdfsd
14: adad
15: asd
16: dfs
17: q
18: titulo de pedido
19: pedido a cia
20: fgdg
21: pedido a cia
22: pedido a cia
23: pcs del futuro
24: pedido a ciaupd

Cantidad de Visitas realizadas: 71

Titulos mas Prestados:
FIN

-----
(program exited with code: 0)
Press return to continue

alberto@holiness: ~/tesis
Archivo Editar Ver Buscar Terminal Ayuda
alberto@holiness:~/tesis$ ./server.pl
La url del daemonio es:http://holiness:8079/
Se realizo la consulta y se devolvieron 25 elementos
1: sugerencia de compra
2: cuba
3: marti
4: opup
5: hjhj
6: jjj
7: aa
8: jkj
9: sugerencia de compra
10: 3433
11: marti
12: 1111
13: sds
14: sdfsd
15: adad
16: asd
17: dfs
18: q
19: titulo de pedido
20: pedido a cia
21: fgdg
22: pedido a cia
23: pedido a cia
24: pcs del futuro
25: pedido a ciaupd

Se realizo la consulta y se devolvieron 1 elementos
1: 71
```

**Figura 15. Pantalla del servicio web mostrarUltimasAdquiciociones.**

### 3.3 Casos de Pruebas

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. Las pruebas de software constituyen un apartado importante de cualquier desarrollo de software garantizando la calidad del producto final desarrollado, validando que el sistema cumpla con el funcionamiento esperado desde el punto de vista de su funcionalidad y rendimiento.

Existen varios métodos para realizar las pruebas, de ellos las pruebas de caja negra y pruebas de caja blanca son los fundamentales y más utilizados actualmente. Los métodos de caja blanca están orientados a probar el código y su funcionamiento siendo necesario que los probadores conozcan el código y la lógica interna del mismo, las pruebas de caja negra por su parte evalúan las funcionalidades a nivel de interfaz y los resultados de las entradas y salidas de los datos al sistema mediante casos de prueba que pretenden demostrar que las funciones del mismo son operativas, que la entrada se acepta adecuadamente y que la salida es correcta, de manera que se mantenga la integridad de la información externa.

Las pruebas de caja negra examinan aspectos del modelo fundamentalmente sin tener muy en cuenta la estructura interna del software. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un

programa, ignorando la estructura de control y centrándose en los requisitos antes mencionados. La presente investigación ha seleccionado para realizar las pruebas el método de Caja Negra, diseñando casos de pruebas para identificar las condiciones que se llevarán a cabo en la prueba y verificar la aplicación exitosa y aceptable de los requisitos del producto. A continuación se describen los casos de prueba asociados a cada caso de uso especificando las variables de entrada y los resultados obtenidos una vez ejecutado el caso de prueba.

**Tabla 7. Caso de prueba para el CU Mostrar últimas Adquisiciones**

Nombre del Caso de Uso	Mostrar últimas Adquisiciones
Entrada	Resultados
Se invoca el servicio con valores de entrada innecesarios.	Se muestra un listado con las últimas adquisiciones del SIGB.
Se llama el servicio y no se especifican valores de entrada.	

**Tabla 8. Caso de Prueba para el CU Mostrar Información Personal**

Nombre del Caso de Uso	Mostrar Información Personal
Entrada	Resultados
<b>Escenario 1: Mostrar Prestamos Asociados al Usuario</b>	
Se llama la función sin especificar usuario.	El sistema devuelve un mensaje indicando que el usuario no es correcto.
Se llama la función enviando datos de un usuario que no existe en la base de datos.	
Se invoca el servicio con un usuario correcto que no tiene préstamos asociados.	Se devuelve una lista vacía.
Se invoca el servicio con un usuario correcto que si tiene préstamos asociados.	Se devuelve una lista con los préstamos asociados al usuario.
<b>Escenario 2: Mostrar Sanciones Asociadas al Usuario</b>	
Se llama la función sin especificar usuario.	El sistema devuelve un mensaje indicando que el usuario no es correcto.
Se llama la función enviando datos de un usuario que no existe en la base de datos.	
Se invoca el servicio con un usuario correcto que no ha sido sancionado.	Se devuelve una lista vacía.
Se invoca el servicio con un usuario correcto que si ha sido sancionado.	Se devuelve una lista con las sanciones asociadas al usuario.

**Tabla 9. Caso de prueba para el CU Mostrar Reporte de Visitas**

Nombre del Caso de Uso	Mostrar Reporte de Visitas
Entrada	Resultados
Se invoca el servicio con valores de entrada innecesarios.	Se muestra un listado con las visitas realizadas al SIGB divididas en correspondencia al término de un día (24 horas), una semana, un mes y un año hasta la hora y fecha actual.
Se llama el servicio y no se especifican valores de entrada.	

**Tabla 10. Caso de prueba para el CU Mostrar Títulos más prestados**

Nombre del Caso de Uso	Mostrar Títulos más prestados
Entrada	Resultados
Se invoca el servicio con valores de entrada innecesarios.	Se muestra un listado con los títulos que ha prestado el SIGB en orden descendente según cantidad de préstamos.
Se llama el servicio y no se especifican valores de entrada.	

**Tabla 11. Caso de prueba para el CU Mostrar Temas más consultados**

Nombre del Caso de Uso	Mostrar Temas más consultados
Entrada	Resultados
Se invoca el servicio con valores de entrada innecesarios.	Se muestra un listado con los temas correspondientes a los títulos más prestados del SIGB en orden descendente según número de coincidencia en los préstamos asociados.
Se llama el servicio y no se especifican valores de entrada.	

### 3.4 Resultados Obtenidos

Cada uno de los casos de prueba anteriormente expuesto arrojó resultados satisfactorios, de manera que la información que se obtuvo fue la que se esperaba. Queda así disponible la primera versión del módulo de servicios web para el SIGB que forma parte del Sistema de Gestión Bibliotecaria utilizado en la Universidad de las Ciencias Informáticas.

Se logró un módulo que cumple con las especificaciones capturadas como requisitos, y que desde el punto de vista del diseño no afecta estructuralmente a los componentes de la Solución, se

puede incorporar fácilmente al SIGB desde donde se pueden consumir cada uno de los servicios expuestos.

El módulo para servicios web queda disponible para los sistemas que forman la actual Solución para Bibliotecas y funciona de manera correcta e independiente al resto de los módulos del SIGB. Actúa como medio de comunicación entre este y los demás componentes facilitándole a los mismos información de interés y abriendo puertas a futuras implementaciones sobre esta Solución.

Al integrar estos sistemas a partir de servicios web, se logra una estrecha vinculación entre ellos, accediendo a funcionalidades específicas del SIGB a partir de cualquier miembro de la Solución, fomentando la reutilización, flexibilidad y agilidad en los servicios que ofrece la biblioteca.

## Conclusiones

Actualmente las organizaciones demandan de propuestas de software que permitan dotarlas de los niveles deseables de agilidad y flexibilidad frente a los cambios en los requerimientos del negocio, el dinamismo del mercado y las demandas de los clientes. El adoptar SOA plantea una serie de logros en beneficio del negocio como el alinear las tecnologías de la información con los objetivos estratégicos de la organización, la mejora en la agilidad del negocio y el incremento en la reutilización de los servicios con el consiguiente ahorro de costes, mantenimiento y tiempo.

La mayoría de las aplicaciones que hoy existen en la web tienen la tendencia a implementar SOA y servicios web ya que estos prometen la integración universal de aplicaciones al punto que se dice que los servicios web son el futuro.

La incorporación de este módulo para servicios web al SIGB constituye un primer paso en la integración de los sistemas que forman la actual Solución para Bibliotecas en la Universidad de las Ciencias informáticas. El presente trabajo abre una puerta a un nuevo universo en estos sistemas donde se puedan implementar o solicitar los servicios de todos en todos.

Durante la investigación fue necesario el estudio de los componentes del Sistema de Gestión Bibliotecaria teniendo en cuenta el estado actual de las relaciones entre estos y la necesidad de integración entre los mismos. Valorar los procesos fundamentales de dichos sistemas y las necesidades de información respecto a los demás. El estudio de sistemas con Arquitectura Orientada a Servicio (SOA) y las tendencias actuales de la misma permitieron ver la posibilidad de integración mediante servicios web como una opción factible y prometedora.

Una vez diseñado el módulo de servicios web, obtenidos resultados satisfactorios en la implementación del mismo y realizadas las pruebas exitosamente, se puede incorporar al SIGB y se puede decir con certeza que la integración entre los sistemas que forman la actual Solución para Bibliotecas utilizada en la Universidad de las Ciencias Informáticas es un hecho.

## Recomendaciones

Se propone que:

- Se incorporen nuevos servicios web al módulo a fin de aumentar la utilización de los servicios del SIGB desde otra plataforma; esta vez quizás utilizando los casos de uso considerados arquitectónicamente significativos.
- Los servicios web publicados se puedan exponer para ser utilizados por otros sistemas del entorno universitario (intranet, portal del graduado, sitio de la FEU, sitios de las facultades, y otros).
- Se realice un estudio similar partiendo esta vez del Repositorio Institucional para poner a disposición de los sistemas de la Solución para Bibliotecas y otros sistemas los servicios que este ofrece.

## Referencias Bibliográficas

- [1] Álvarez Fernández, Luis Carlos. Implantación de un Repositorio Institucional en la Universidad de las Ciencias Informáticas. 2010 [Consultado: diciembre de 2011]. Disponible en: [http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/TD\\_03552\\_10/1/TD\\_03552\\_10.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_03552_10/1/TD_03552_10.pdf)
- [2] GONZÁLEZ HERRERA, G. Software libre vs. Propietario: una evaluación de sistemas Janium vs. Koha, 2010. [Consultado: 10 de septiembre de 2011]. Disponible en: <http://eprints.rclis.org/bitstream/10760/14590/1/tesis.pdf>
- [3] Porras Herrera, Lityuen Amalia y Veloz Morales, Dargel. Implantación de un Sistema Integrado de Gestión Bibliotecaria (SIGB) en la biblioteca de la UCI. 2008 [Consultado: marzo de 2011]. Disponible en: [http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/TD\\_1257\\_08/1/TD\\_1257\\_08.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_1257_08/1/TD_1257_08.pdf)
- [4] Álvarez, Miguel Angel. “Qué es Perl,”. [Consultado: 9 de octubre de 2011] Disponible en: <http://www.desarrolloweb.com/articulos/541.php>
- [5] Wall, Larry. “The Perl Programming Language - www.perl.org,” [Consultado: 9 de octubre de 2011] Disponible en: <http://www.perl.org/>
- [6] Gracia, Joaquín. “UML: Diagramas UML. ¿Qué es UML? Análisis y Diseño. Ingeniería del Software.,” [Consultado: 9 de octubre de 2011]. Disponible en: <http://www.ingenierosoftware.com/analisisydiseno/uml.php>
- [7] Laurencio Pérez, Dariuska; Diéguez García, Yunieski. “Sistema de Gestión de la Trayectoria Productiva de cada estudiante en la facultad 1,” Universidad de las Ciencias Informáticas, 2009. [Consultado: 3 de abril de 2012] Disponible en: [http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/TD\\_1903\\_09/1/TD\\_1903\\_09.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_1903_09/1/TD_1903_09.pdf)
- [8] Association of Research Libraries (2006). SPEC Kit 292: institutional repositories : executive summary [en línea]. Washington, DC: ARL. [Consultado 05 de Mayo de 2012] Disponible en: <http://www.arl.org/bm~doc/spec292web.pdf>
- [9] Bea, Vangie: Understanding Web Services - Webopedia.com. 2010 [Consultado 05 de Mayo de 2012] Disponible en: [http://www.webopedia.com/DidYouKnow/Computer\\_Science/2005/web\\_services.asp](http://www.webopedia.com/DidYouKnow/Computer_Science/2005/web_services.asp)

## Bibliografía

1. Brea, Orlando Fabián: 2011. WEB-SERVICES-CON-PHP. [Consultado: 8 de Diciembre de 2011]. Disponible en:  
<http://www.google.com/url?sa=t&rct=j&q=histioa+web+service&source=web&cd=5&ved=0CEYQFjAE&url=http%3A%2F%2Fwww.wiener.edu.pe%2Fmanuales%2F6to-ciclo%2FPROGRAMACION-WEB-4%2FWEB-SERVICES-CON-PHP.pdf&ei=VXTgTqzMNqLb0QHgrf2MBw&usq=AFQjCNExBRA1aS08eTsjDdVDMg387p72kw>
2. Brea, O. Fabián, 2005. Historia de los Web Services. desarrollowe.com. [Consultado: 8 de Diciembre de 2011]. Disponible en: <http://www.desarrollowe.com/articulos/1883.php>
3. Cabranes Pérez, Yergeni y Martínez Porra, Raly: Propuesta de Arquitectura Orientada a Servicios para el Portal de Servicios Comunitarios, 2008. [Consultado: 11 de octubre de 2011] Disponible en:  
[http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/TD\\_1676\\_08/1/TD\\_1676\\_08.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_1676_08/1/TD_1676_08.pdf)
4. Chan, Leslie: 2004. Supporting and Enhancing Scholarship in the Digital Age: The Role of Open Access Institutional Repositories. [Consultado: 13 de marzo de 2012] Disponible en: <http://eprints.rclis.org/handle/10760/5648>.
5. Ciberaula - Cursos on-line - Formación a empresas - E-learning - Consultoría de formación. [Consultado: 8 de Mayo de 2012a]. Disponible en: <http://www.linux.ciberaula.com/>
6. Cuan Céspedes, José Andrés y Manzano Chacón, Geovanis: Procedimiento para la integración de sistemas legados en organizaciones empresariales con arquitecturas orientadas a servicios, 2010 [Consultado: 7 de octubre de 2011] Disponible en:  
[http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/TD\\_02944\\_10/1/TD\\_02944\\_10.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_02944_10/1/TD_02944_10.pdf)
7. González Herrera, G.: Software libre vs. propietario: una evaluación de sistemas Janium vs. Koha, 2010. [fecha de consulta: 10 de septiembre de 2011] Disponible en: <http://eprints.rclis.org/bitstream/10760/14590/1/tesis.pdf>
8. Guía Breve de Servicios Web. World Wide Web consortium. [Consultado: 8 de Diciembre de 2011d]. Disponible en: <http://www.w3c.es/divulgacion/guiasbreves/ServiciosWeb>
9. Gutiérrez Gómez, Isaac y Otón Tortosa, Salvador: Departamento de Ciencias de la Computación de la Escuela Técnica Superior de Ingeniería Informática en la Universidad de Alcalá: Arquitecturas Orientadas a Servicios.[Consultado: 13 de octubre de 2011] Disponible en: <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-132/paper09.pdf>
10. Jacobson, I. El proceso Unificado de Desarrollo de Software, La Habana: 2004. p 438.
11. Jaime Arias, Yamile A. y Rojas Hernández, Alix E.: Semántica Web en una SOA para bibliotecas digitales [Consultado: 9 de abril de 2012].
12. Li, H., Councill, I., Lee, W., and Giles, C. L. 2006. CiteSeerx: an architecture and web service design for an academic document search engine. In Proceedings of the 15th International Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 883-884. DOI=

<http://doi.acm.org/10.1145/1135777.1135926>

13. Lynch, Clifford. 2003. Institutional Repositories: Essential Infrastructure for Scholarship in the Digital Age. [Consultado: 13 de marzo de 2012] Disponible en: <http://www.arl.org/bmdoc/br226ir.pdf>.
14. Lynch 2003 & repositorio, [Consultado: 8 de diciembre de 2011]. Disponible en: [www.fesabid.org/zaragoza2009/actas-fesabid-2009/99-107.pdf](http://www.fesabid.org/zaragoza2009/actas-fesabid-2009/99-107.pdf)
15. MasterMagazine, Tendencias: Conceptos básicos de la Arquitectura Orientada a Servicios, 17 de Enero de 2005.[Consultado: 13 de octubre de 2011] Disponible en: <http://www.mastermagazine.info/articulo/>
16. Méndez Torres, José Alberto y Dariel Núñez Llerena: Propuesta de Modelo para la Gestión de Servicios en Arquitecturas Orientadas a Servicios, 2009. [Consultado: 5 de octubre de 2011] Disponible en: [http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/TD\\_2169\\_09/1/TD\\_2169\\_09.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_2169_09/1/TD_2169_09.pdf)
17. Oracle Corporation, Informe Ejecutivo de Oracle: Gestión de Procesos de Negocio, Arquitectura Orientada a Servicios y Web 2.0: ¿Transformaciones de Negocio o problemática Global?, agosto de 2008.[Consultado: 13 de octubre de 2011] Disponible en: <http://www.oracle.com/technetwork/es/middleware/fusion-middleware/documentation/gestion-proceso-negocio-soa-web-450487-esa.pdf>
18. Pena Fernández, Johnny Frank y Jorge Félix Portilla Castells: Propuesta de un modelo para la definición de políticas en la implantación de una Arquitectura Orientada a Servicios, 2010. [Consultado: 9 de octubre de 2011] Disponible en: [http://repositorio\\_institucional.uci.cu/jspui/bitstream/ident/TD\\_03140\\_10/1/TD\\_03140\\_10.pdf](http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_03140_10/1/TD_03140_10.pdf)
19. Porcel Iturralde ML, Rodríguez Mederos M. Software libre: una alternativa para las bibliotecas. Acimed 2005;13(6).[Consultado: 10 de septiembre de 2011] Disponible en: [http://bvs.sld.cu/revistas/aci/vol13\\_6\\_05/aci090605.htm](http://bvs.sld.cu/revistas/aci/vol13_6_05/aci090605.htm)
20. Qué es Perl. [Consultado: 8 de Mayo de 2012h]. Disponible en: <http://www.desarrolloweb.com/articulos/541.php>
21. Red de Bibliotecas. [Consultado: 18 de Mayo de 2012i]. Disponible en: <http://www.redbibliotecas.carm.es/RedBibliotecas/faces/indexp.jsp>
22. Ríos Peña. Alejandro, 004AD. Servicios Web con Perl - Grupo GNU/Linux de la Universidad del Cauca. Servicios Web con Perl. [Consultado: 16 de Mayo de 2012]. Disponible en: [http://gluc.unicauca.edu.co/wiki/index.php/Servicios\\_Web\\_con\\_Perl](http://gluc.unicauca.edu.co/wiki/index.php/Servicios_Web_con_Perl)
23. Saffirio, Mario: 2006. ¿Qué son los Web Services? [Consultado: 6 de Diciembre de 2011]. Disponible en: <http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>
24. Salamanca, Universidad de. 2009. Universidad de Salamanca. [Consultado: 8 de diciembre de 2011]. Disponible en: [http://web.usal.es/~angelpoveda/web%20biologia/tutoriales/cat%C3%A1logos.%20repositorios%20y%20bibliotecas%20virtuales1/repositorios\\_digitales.html](http://web.usal.es/~angelpoveda/web%20biologia/tutoriales/cat%C3%A1logos.%20repositorios%20y%20bibliotecas%20virtuales1/repositorios_digitales.html)
25. The Perl Programming Language - www.perl.org. [Consultado: 8 de Mayo de 2012m].

Disponible en: <http://www.perl.org/>

26. Tröger Enrico et al., 2008. Geany: Home Page. [Consultado: 8 de Mayo de 2012]. Disponible en: <http://www.geany.org/>
27. UML: Diagramas UML. ¿Qué es UML? Análisis y Diseño. Ingeniería del Software. [Consultado: 8 de Mayo de 2012n]. Disponible en: <http://www.ingenierosoftware.com/analisisydiseno/uml.php>
28. Vimal Kumar VA, Thomas. Eight Things you should Know about Open Source Integrated Library Systems. Cochinn University of Science and Technology. 2009 [Consultado: 11 de septiembre de 2011]. Disponible en: <http://eprints.rclis.org/bitstream/10760/13858/1/20.pdf>
29. w3c UDDI. WSDL and UDDI. 2009. [Consultado: 2 de diciembre de 2011]. Disponible en: [http://www.w3schools.com/WSDL/wsd\\_ uddi.asp](http://www.w3schools.com/WSDL/wsd_ uddi.asp) .
30. w3c WSDL. Web Service Definition Language (WSDL). 2009. [Consultado: 6 de diciembre de 2011]. Disponible en: <http://www.w3.org/TR/wsd/>.
31. w3c UDDI. WSDL and UDDI. 2009. [Consultado: 18 de octubre de 2011]. Disponible en: [http://www.w3schools.com/WSDL/wsd\\_ uddi.asp](http://www.w3schools.com/WSDL/wsd_ uddi.asp).
32. w3c WSDL. Web Service Definition Language (WSDL). 2009. [Consultado: 18 de octubre de 2011]. Disponible en: <http://www.w3.org/TR/wsd/>.
33. w3c XML. Extensible Markup Language (XML). 2009. [Consultado: 12 de noviembre de 2011]. Disponible en: <http://www.w3.org/XML/>.
34. w3c XML. Extensible Markup Language (XML). 2009. [Consultado: 2 de diciembre de 2011]. Disponible en: <http://www.w3.org/XML/>.
35. Web Services: XML-RPC, SOAP, sobre PHP, Perl, y otros conceptos. Programación en Castellano. [Consultado: 8 de Diciembre de 2011r]. Disponible en: [http://www.programacion.com/articulo/web\\_services:\\_xml-rpc-\\_soap-\\_sobre\\_php-\\_perl-\\_y\\_otros\\_conceptos\\_211/2](http://www.programacion.com/articulo/web_services:_xml-rpc-_soap-_sobre_php-_perl-_y_otros_conceptos_211/2)
36. XML-RPC. XML-RPC Home Page. 2008. [Consultado: 6 de diciembre de 2011]. Disponible en: <http://www.xmlrpc.com/>.
37. XML-RPC. XML-RPC Home Page. 2008. [Consultado: 2 de diciembre de 2011]. Disponible en: <http://www.xmlrpc.com/>.
38. web-services.pdf. [Consultado: 8 de Diciembre de 2011q]. Disponible en: <http://comunidades.uci.cu/blogs/echo/wp-content/uploads/2010/01/web-services.pdf>

## Glosario de Términos

**SIGB:** Sistema Integrado de Gestión Bibliotecaria, conjunto organizado de recursos humanos que utiliza dispositivos y programas informáticos, adecuados a las naturaleza de los datos que deben procesar, para realizar procesos y facilitar los servicios que permiten alcanzar los objetivos de la biblioteca: almacenar de forma organizada el conocimiento humano contenido en todo tipo de materiales bibliográficos para satisfacer las necesidades informativas, recreativas y de investigación de los usuarios.

**Servicio Web:** aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información.

**KOHA:** SIGB desarrollado en Nueva Zelanda por la empresa Katipo Communications en 1999, a solicitud de la Horowhenua Library Trust. Es un software de código abierto liberado bajo Licencia Pública General (GPL) y para su desarrollo cuenta con el esfuerzo de bibliotecarios y desarrolladores de distintos países.

**Soluciones:** aplicaciones informáticas, ya sean webs o de escritorio que interactúan con los usuarios y son las que acceden a los servicios que componen la SOA.

**Procedimiento:** Básicamente, un procedimiento consiste de una serie de pasos bien definidos que permitirán y facilitarán la realización de un trabajo de la manera más correcta y exitosa posible.

# Anexos

## Anexo 1: Código Perl para generar un archivo wsdl.

```
#!/usr/bin/perl -w
use WSDL::Generator;
my $init = {
    'schema_namesp' => 'http://10.53.9.8/tesis/Webservice.wsdl',
    'services' => 'Webservice',
    'service_name' => 'Webservice',
    'target_namesp' => 'http://10.53.9.8:8079/Webservice.wsdl',
    'documentation' => 'Servicio Web de Prueba',
    'location' => 'http://10.53.9.8/tesis/Webservice'
};
my $w = WSDL::Generator->new($init);
Webservice->nombredelaFuncion();
print $w->get(Webservice);
```

## Anexo 2 Clase utilizada para publicar los servicios web, **servidor.pl**

```
#!/usr/bin/perl -w
use strict;
# Servidor SOAP
use SOAP::Transport::HTTP;
use Webservice;
#En el parámetro dispatch_to se especifica la ruta a los módulos disponibles
my $daemonio = SOAP::Transport::HTTP::Daemon
-> new(LocalAddr => '10.53.9.8', LocalPort => 8079)
-> dispatch_to('http://10.53.9.8/tesis/', 'Webservice')
;
print "La url del daemonio es:". $daemonio->url(). " ";
$daemonio->handle();
```

### Anexo 3 Cliente Perl para consumir los servicios web, **cliente.pl**

```
#!/usr/bin/perl -w
use strict;
use SOAP::Lite;
my $soap= SOAP::Lite
-> uri('http://10.53.9.8:8079/Webservice')
-> proxy('http://10.53.9.8:8079')
;
#mostrar adquisiciones
my @adquisiciones= $soap->ultimasAdquisiciones()->paramsout();
my $i=0;
print "\n Ultimas Adquisiciones: ($cantidad)\n";
foreach my $value (@adquisiciones){
    $i++;
    print "$i: $value\n";
}
```