

**Universidad de las Ciencias Informáticas
Facultad 10**



**Título: Servicio Web para la gestión de Préstamos en el Centro Rector de
Universidad para Todos.**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores: Yanni Rodríguez Romero.
Daniel Hernández Velázquez.

Tutor: Lic. Daynel Marmol Lacal.

Co-tutor: Ing. Misael Fonseca Mata.

Ciudad de La Habana, junio 2008.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 _____.

Autor: Yanni Rodríguez Romero.

Autor: Daniel Hernández Velázquez.

Firma del Autor

Firma del Autor

Tutor: Lic. Daynel Marmol Lacal.

Co-tutor: Ing. Misael Fonseca Mata.

Firma del Tutor

Firma del Co-tutor

DATOS DE CONTACTO

Tutor: Lic. Daynel Marmol Lacal (dmarmol@uci.cu)

Graduado de Licenciatura en Ciencias de la Computación en la Universidad Central “Martha Abreu” de Las Villas en julio del 2005. Desde su graduación está trabajando en la UCI, desempeñándose como profesor de las asignaturas Programación II (Estructura de Datos) y Programación IV (Compiladores), desde su incorporación al centro es líder del proyecto para informatizar el Centro Rector de Universidad para Todos.

DEDICATORIA

Yanni

A mis padres que con todo su amor me han dado siempre su apoyo, confianza y comprensión, dedico este trabajo, mi carrera y todo lo que soy. A mi maestro Jorge por ayudarme a dar los primeros pasos en mi vida de estudiante y además por todo su cariño. A mis hermanos, mis amistades y demás familiares que me apoyaron en todos estos años.

Daniel

A mi hermano. A todos mis familiares en especial a mi papá y mamá, mi hermano Rey y mi hermana Mayita y mis tres sobrinos. A mi novia por quererme tanto y saberme esperar siempre. A todos mis amigos en especial Miguelito y Carlitos. A mis compañeros de proyecto, a mis compañeros de cuarto en especial a los Trovo-Sapiens. A todos los que de una forma u otra han tenido que ver con la realización de este largo sueño que hoy se hace realidad.

RESUMEN

El Centro Rector de Universidad para Todos, es una institución que tiene la misión de prestar servicios a profesores de Universidad para Todos y Canales Educativos, con este objetivo, la Revolución ha puesto en sus manos los medios necesarios para su desempeño.

En este trabajo se presenta el desarrollo de un Servicio Web para la gestión de préstamo de servicios en el Centro Rector del Claustro de Universidad para Todos. Para una mejor comprensión del problema se debe partir de la definición de elementos fundamentales como: gestión de préstamo de servicios, el análisis de la importancia de la elaboración de dicho sistema y los beneficios que trae consigo, para ello se ha realizado un estudio de los sistemas de este tipo desarrollados hasta el momento, tomando los más usados como punto de partida para este trabajo. El objetivo fundamental que se persigue con esta investigación consiste en desarrollar un servicio web que sea capaz de gestionar el préstamo de los servicios que brinda esta entidad.

Para la creación de este Sistema de Gestión de Préstamo de Servicios basado en un Servicio Web se trabaja con el lenguaje de programación PHP, como servidor de base de datos MySQL, servidor web Apache y paquete de librerías ADOdb para la conexión abstracta a bases de datos. Se utiliza la metodología RUP para llevar a cabo el proceso de desarrollo y como lenguaje de modelado UML a través de la herramienta CASE Visual Paradigm.

PALABRAS CLAVE

Préstamo, servicios, Servicio Web.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Definiciones y tipologías.....	6
1.2 Sistemas automatizados de gestión de préstamo de servicios.	6
1.3 Estado del arte de los sistemas de gestión de préstamo de servicios.	7
1.4 Necesidad de los Sistemas de Gestión de Servicios.	10
1.5 ¿Qué ventajas posibilitaría un Sistema de Gestión de Préstamo de Servicios en el Centro Rector de Universidad para Todos?.....	11
1.6 Análisis de las tecnologías y software.....	11
1.6.1 ¿Por qué un Servicio Web?.....	11
1.6.1.1 XML.....	12
1.6.1.2 SOAP	12
1.6.2 ¿Por qué PHP?	13
1.6.3 Sistema Gestor de base de datos (SGBD).....	13
1.7 Fundamentación de la metodología a utilizar.....	14
1.7.1 Desarrollo basado en RUP.....	14
1.7.2 UML.....	15
1.8 Herramientas a utilizar.	15
1.8.1 Visual Paradigm.	15
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	17
2.1 Objetivos estratégicos de la organización	17
2.2 Descripción de los procesos que serán objeto de automatización.	17
2.3 Información que se maneja.	18
2.4 Propuesta del sistema.....	19
2.5 Modelo de dominio	19
2.6 Modelo del sistema.	22
2.6.1. Requerimientos funcionales.	23
2.6.2. Requerimientos no funcionales.	25
2.7 Definición de los casos de uso.....	26
2.7.1 Actores del sistema.	26
2.7.2 Diagrama de casos de uso del sistema.....	26
2.7.3 Casos de uso del sistema. (CUS)	28
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA	37
3.1 Diseño	37
3.1.1 Representación UML del Diagrama de clases del diseño.	37
3.1.2 Representación UML de los diagramas de interacción.	40
3.1.3 Descripción de las clases del diseño.....	44

3.2 Diseño de la base de datos.....	72
3.2.1 Representación UML del Diagrama de Entidad - Relación.	73
3.2.2 Descripción de las tablas de la Base de Datos.	73
CAPÍTULO 4: IMPLEMENTACIÓN.....	74
4.1 Diagrama de despliegue.	74
4.2 Representación UML del Diagrama de Despliegue.....	74
4.3 Modelo Implementación	74
4.3.2 Descripción de cada uno de los Paquetes	79
CONCLUSIONES	80
RECOMENDACIONES.....	81
REFERENCIAS BIBLIOGRÁFICAS.	82
BIBLIOGRAFÍA.....	83
ANEXOS.....	84
GLOSARIO.....	90

INTRODUCCIÓN

El vertiginoso desarrollo de la tecnología en nuestros días, ha contribuido en favor de que exista un mayor intercambio de información a nivel mundial. Tal situación presupone que para que se logre el éxito, es decir, que la información garantice la gestión eficaz de los recursos de las organizaciones, se ha hecho inevitable el surgimiento de los sistemas de información.

El incremento de la competencia en los mercados de bienes y servicios, la rapidez en el desarrollo de las Tecnologías de Información y las Comunicaciones (TIC), el aumento de la incertidumbre en el entorno y la reducción de los ciclos de vida de los productos, entre otros factores, hacen que la información se convierta en un elemento clave para la gestión organizacional, así como para su supervivencia y crecimiento.

El disponer de información completa, exacta y confiable en el momento oportuno, constituye un elemento esencial para garantizar una gestión eficaz de los recursos en las organizaciones, así como mejorar la calidad de los servicios que prestan y adecuarse a su entorno.

Actualmente las empresas invierten importantes sumas en recursos en tecnologías de la información (TI) con el fin de apoyar los procesos del negocio. El valor que adquiere el uso la misma para las organizaciones, establece que todos los procesos de producción, administración y uso de servicios deben ser gestionados y controlados de manera eficiente para asegurar la calidad de esta y el cumplimiento de los objetivos del negocio.

El procesamiento de datos e información referente a las operaciones del negocio, necesitan de la aplicación de medidas de control, dentro de un sistema de gestión que permita garantizar la prestación de los servicios y la reducción de vulnerabilidades que pueden poner en peligro la estabilidad del sistema operativo, organizacional y de la compañía en general.

En las organizaciones de todo nivel existe un área que genera y provee los servicios TI, y un grupo usuarios (internos y/o externos) que demandan dichos servicios y esperan su prestación oportuna y, con calidad.

Con el objetivo de crear una infraestructura productiva para garantizar la calidad y actualización de la información surge el Centro Rector de Universidad para Todos, el cual provee servicios y productos de información especializada, primordialmente dirigidos a cubrir las necesidades y demandas de esa índole de los usuarios, en especial el sector académico. No obstante, de acuerdo con el carácter y los objetivos de la Universidad para Todos, merced a su infraestructura y capacidad técnica y profesional, el centro

también ofrece sus servicios y productos a los profesionales que no sean usuarios habituales que requieran de este tipo de recursos. Por esta razón se necesita una buena gestión de los préstamos que se realizan, para garantizar la integridad física y una mejor organización de los medios.

Uno de los principales aportes que se esperan de este sistema es lograr una mayor eficiencia y rapidez en el control y aprovechamiento racional de los recursos de la entidad, así como registrar los servicios prestados a los diversos usuarios de la institución, posibilitando cuantificar el tiempo y los recursos empleados para cada servicio.

En esta institución no ha existido ninguna experiencia previa en cuanto a este tipo de trabajo, ya que el Centro Rector de Universidad para Todos (UpT) es un proyecto nuevo creado con el objetivo de mantener unido al claustro de UpT y los Canales Educativos.

La misma se está iniciando en la informatización de sus procesos, es ahí donde juega un papel fundamental la Universidad de las Ciencias Informáticas (UCI), como retaguardia tecnológica, con el objetivo de ayudar a crear una infraestructura productiva para garantizar la calidad y actualización de la información y el conocimiento que necesitan sus usuarios.

Situación problemática

En estos momentos en la institución no existe un sistema automatizado para gestionar y controlar el préstamo de servicios a usuarios. Realizar este trabajo de forma manual, traería como consecuencia que el servicio se torne lento, además puede ocurrir pérdida de datos, control ineficiente de la información, lo cual causa molestias a los usuarios y mal funcionamiento en el área de préstamos de la entidad.

El **problema científico** de la investigación es:

¿Cómo resolver de forma automatizada el proceso de gestión de préstamo de servicios en el Centro Rector del Claustro de Universidad para Todos?

Considerando como **objeto de estudio** el proceso de gestión y control de servicios, se toma como **campo de acción** el proceso de gestión y control de préstamo de servicios en el Centro Rector del Claustro de Universidad para Todos y los canales educativos.

Objetivo general

Diseñar e implementar un servicio web para gestionar el préstamo de servicios, de forma tal que reúna las características y expectativas del control de los medios de la institución.

Objetivos específicos de la investigación

- 1 Determinar las necesidades de diagrama de flujos de información referente a las solicitudes de los servicios en la institución.
- 2 Analizar los aspectos conceptuales relacionados con la información obtenida sobre los materiales para el préstamo.
- 3 Realizar el diseño de un servicio web para gestionar el préstamo de servicios, que pueda integrarse con el resto de los módulos del Centro Rector del claustro de Universidad para todos.
- 4 Implementar un Servicio Web que permita controlar el préstamo de servicios en la institución.
- 5 Depurar el Servicio Web por medio de pruebas efectuadas con datos reales obtenidos de los encargados de controlar el préstamo de servicios.

Las **preguntas científicas** de esta investigación son las siguientes:

1. ¿Cómo mejorar el flujo de trabajo en la gestión de préstamo de servicios?
2. ¿Cuáles son las principales necesidades que lograría satisfacer el módulo a automatizar?
3. ¿Cuán beneficioso sería la implementación de un sistema de gestión de préstamo de servicios?
4. ¿Cómo se lograría la seguridad de los datos administrados por los trabajadores?

Como vía para cumplir los objetivos propuestos se definieron las siguientes **tareas de investigación**:

1. Investigar sobre las diferentes tendencias y tecnologías más usadas y actuales para el desarrollo de Servicios Web.
2. Desarrollar un estudio de las herramientas de gestión de préstamo de servicios existentes en diferentes entidades.
3. Reconocer los requerimientos que debe cumplir el Módulo de Préstamos.
4. Realizar la modelación de diagramas para la fácil comprensión del diseño y la implementación del sistema.
5. Desarrollar un Servicio Web que facilite la gestión de préstamo de servicios del Centro Rector de Universidad para Todos.

La realización de las tareas de la investigación y con ello el logro de los objetivos planteados trae consigo un mejor desarrollo del servicio que necesita el Centro Rector de Universidad para Todos, el cual brindaría rapidez y seguridad en el trabajo con los datos, además disminuye la posibilidad de que ocurran pérdidas en la institución.

Se utilizaron varios métodos con vista a dar cumplimiento a las tareas planteadas, como fueron:

Métodos teóricos:

- ✓ Analítico-Sintético

Este método ha servido para extraer los elementos más importantes y relacionados con el objeto de estudio, centrándose en el análisis de documentos, teorías y otros materiales como manuales que permitan elaborar conclusiones.

- ✓ Análisis Histórico-Lógico

El Análisis Histórico-Lógico se utilizará para el estudio de la evolución y desarrollo del objeto de estudio de la investigación.

Métodos empíricos:

- ✓ Observación.

Se ha observado la situación real que se está investigando, para poder conocer los procesos del negocio relacionado con el préstamo de servicios en la institución, permitiendo un mayor acercamiento al objetivo final.

- ✓ Entrevistas

Como técnicas para la recopilación de la información del negocio, en este caso el mayor aporte se centra en entrevistas planificadas y una comunicación fluida entre ambas partes. Cabe destacar que estas entrevistas son muy útiles ya que tienen una gran flexibilidad al estructurar y realizar las preguntas, y permiten llegar a una mayor profundidad y amplitud sobre el tema.

Estructuración del contenido

Este documento está estructurado en 4 capítulos además de las conclusiones, recomendaciones, referencia bibliográfica, bibliografía, anexos y glosario de términos.

Capítulo 1: Fundamentación del tema. Se describen algunos conceptos y características de sistemas de gestión y control de préstamos que se han estudiado para poder desarrollar el trabajo y se realiza un análisis de las tecnologías de software empleadas para desarrollar el sistema.

Capítulo 2: Características del Sistema. Se abordan las descripciones iniciales del sistema (servicio web), se hace un estudio del Centro Rector de Universidad para Todos y Canales Educativos y un análisis crítico de los procesos involucrados en la gestión de servicios, se modela el dominio, se presenta la propuesta del sistema, se definen los requerimientos funcionales y no funcionales y los casos de uso.

Capítulo 3: Diseño del sistema propuesto. Se describe el diseño del sistema que se propone. Se representan los diagramas de interacción, los diagramas de clases y el diseño de la base de datos.

Capítulo 4: Implementación del sistema. Se describe el sistema que se propone a través de una óptica de programación, basado en diagramas de componentes y diagrama de despliegue.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se abordarán características y conceptos esenciales con relación a los sistemas de gestión y control de préstamo de servicios, así como las tendencias actuales de los servicios de este tipo, además se fundamenta el uso de cada metodología, herramienta, tecnología o software en el desarrollo de nuestro proyecto teniendo en cuenta los intereses del usuario.

1.1 Definiciones y tipologías.

Teniendo en cuenta que un préstamo es la acción y efecto de prestar un medio material a una persona determinada para que lo utilice durante algún tiempo y después lo restituya o devuelva [1].

Se puede decir que el servicio de préstamo permite al usuario hacer uso de los recursos que se encuentran disponibles en el fondo de información de la institución por un período de tiempo específico.

La gestión de servicios tecnológicos se encuentra definida por las mejores prácticas de la Biblioteca de Infraestructura de la Tecnología de la Información (ITIL), como “la disciplina que se enfoca en la gestión del grupo de personas, procesos y tecnología que cooperan para asegurar la calidad de los servicios de tecnologías de la información (TI), en concordancia a unos niveles de servicio pactados previamente con el cliente”. [2]

La gestión de la información se vincula con la generación y la aplicación de estrategias, el establecimiento de políticas, así como con el desarrollo de una cultura organizacional y social dirigida al uso racional, efectivo y eficiente de la información en función de los objetivos y metas de las organizaciones en materia de desempeño y de calidad.

1.2 Sistemas automatizados de gestión de préstamo de servicios.

Con el paso del tiempo los ordenadores han contribuido a incrementar la fiabilidad de los archivos de préstamos, se han ido experimentando y aplicando varios métodos automatizados de préstamo y control de los inventarios. Se crean sistemas con el objetivo de realizar las actividades de manera más organizada y efectiva.

En la actualidad la creciente dependencia de las empresas a la tecnología ha aumentado la dificultad para realizar una gestión eficiente de los servicios. Por esta razón ahora es necesaria la utilización de

soluciones y herramientas que permitan garantizar que las empresas brinden servicios tecnológicos fiables, implementando procesos de gestión de servicios integrados y que además, consideren los componentes tecnológicos como parte articulada de los servicios que suministra la empresa, para así lograr una eficiente gestión de los mismos.

La gestión de servicios de TI, es la disciplina utilizada en las empresas que prestan servicios o una combinación de bienes y servicios. Aunque ampliamente utilizados en la industria de TI, específicamente en las TIC, la gestión de los servicios pueden ser integrados en muchas otras industrias.

Los sistemas que utilizan la gestión de los servicios pueden incluir la gestión de pedidos, gestión de inventario, la activación, mantenimiento, el rendimiento diagnóstico y varios otros tipos de sistemas de apoyo para asegurarse de que estos sistemas se ejecuten libres de errores.

Hoy en día, se pueden encontrar muchos recursos y proveedores de servicios que ayudan a las pequeñas y grandes empresas a incorporar los principios de gestión de los servicios en su organización.

1.3 Estado del arte de los sistemas de gestión de préstamo de servicios.

Se ha realizado una amplia investigación acerca de los Sistemas de Gestión de Préstamo de Servicios y se han encontrado varios ejemplos de aplicaciones web vinculados fundamentalmente a la gestión de préstamos bibliotecarios y de servicios informáticos.

✓ **En el mundo.**

Red Municipal de Bibliotecas de Murcia (RMBM)

La Red Municipal de Bibliotecas de Murcia (RMBM) cuenta con un sistema que tiene como objetivos principales: cubrir las necesidades de información y de lectura de los ciudadanos del municipio de Murcia, además pretende ser un lugar en el que ocupen su tiempo libre, ser un espacio en el que accedan a las tecnologías de la información, y ser una zona para la participación en actividades culturales y de promoción de la lectura.

AYUNTAMIENTO DE MURCIA www.murcia.es

RED MUNICIPAL DE BIBLIOTECAS DE MURCIA




[Información/Contacto](#) | [Utilidades](#) | [RMBM:net](#) | [Mapa Web](#)

Estás en RMBM > Catálogo y servicios web

Catálogo y servicios web

El Catálogo colectivo de la RMBM, ccRMBM, contiene información de los libros, discos, revistas, mapas, etc. de los que puedes disponer, es decir, te dice qué tenemos, dónde está y si se encuentra prestado o no.

Catálogo:

- [Acceso al ccRMBM](#)
- [Envío de desideratas](#)

Servicios web ligados al carnéRMBM:

- [Consulta de préstamos](#)

Enlaces relacionados

- [Catálogo Red de bibliotecas Región de Murcia](#)

Catálogos colectivos:

- [Bibliotecas públicas del Estado](#)
- [Patrimonio bibliográfico español](#)
- [Publicaciones periódicas](#)
- [Andalucía](#)
- [Aragón](#)
- [Asturias](#)




Aspiran llegar a ser el soporte cultural e informativo básico del municipio, siempre al servicio de los ciudadanos, las instituciones y las empresas. Este sistema permite la solicitud de diferentes servicios como: gestión de información, servicio de préstamos, siempre portando el carné que lo identifica como miembro del centro, programar actividades y además atender las quejas, sugerencias y reclamaciones de los usuarios.

✓ **En Cuba.**

Portal de la Salud en Cuba

Infomed es la red de personas e instituciones que trabajan y colaboran para facilitar el acceso a la información y el conocimiento para mejorar la salud de los cubanos y de otros pueblos del mundo, mediante el uso intensivo y creativo de las tecnologías de la información y la comunicación.

Próximos eventos
H T A IV Simposio Internacional sobre Hipertensión Arterial del 26-29 de Mayo 2008

Eventos Cursos

VIDEO SALUD 2008 29 SEPT. - 3 OCT. LA HABANA CUBA

Especialidades

Cirugía pediátrica, CIRURED, Embriología, Estadísticas, Farmacia, Gastroenterología,

Mensaje de Fidel

 A mis entrañables compatriotas, que me hicieron el inmenso honor de elegirme en días recientes como miembro del Parlamento, en cuyo seno se deben adoptar acuerdos importantes para el destino de nuestra Revolución, les comunico que no aspiraré ni aceptaré- repito- no aspiraré ni aceptaré, el cargo de Presidente del Consejo de Estado y Comandante en Jefe, [Texto completo](#)

Feria del Libro

En las sesiones de la XVII Feria Internacional del Libro, la Dirección Nacional de Publicaciones


Al Día

Muestra de cabello permite diagnosticar cáncer de mama

Prueba sanguínea puede detectar cáncer de ovarios

Relacionan uso del teléfono móvil al desarrollo de tumores

El portal de la salud en Cuba también posee un sistema que gestiona diversos servicios útiles para los médicos y trabajadores de la salud cubanos, brinda acceso a documentos, libros, revistas, bases de datos, además da la posibilidad de informarse sobre enfoques médicos novedosos, tiene previsto brindarles temas periódicamente en los que participe como un actor más en una sala de debate, compartiendo con nosotros este espacio de intercambio de conocimientos y creación. Otra de las funcionalidades que ofrece el sistema es la de facilitar a los usuarios la búsqueda y divulgación de cursos de salud, tanto nacionales como internacionales, virtuales y presenciales, todos estos servicios se pueden solicitar mediante la web.

✓ **En la Universidad de la Ciencias Informáticas.**

En nuestra universidad en la url: <http://pase.uci.cu> se encuentra el sitio para gestionar el servicio de pase para los estudiantes, a través del cual se controlan los datos de los usuarios que solicitan este servicio. Este sitio Web se desarrolló con el lenguaje de programación script Php y como gestor de base de datos MySQL.

<input type="radio"/> Reserva de ida <input checked="" type="radio"/> No reservar <input type="radio"/> Reservar	<input type="radio"/> Reserva de regreso <input checked="" type="radio"/> No reservar <input type="radio"/> Reservar
<input type="radio"/> Datos de la persona que va a visitar	
<input checked="" type="radio"/> Familiar: Alberto Hernandez Parentesco: Primo <input type="radio"/> Calle: Playa No. 23 Apto: 456 <input type="radio"/> Entre: Calle y calle Localidad: Playa Teléfono:	
<input type="radio"/> Familiar: <input type="text"/>	<input type="radio"/> Parentesco: <input type="text"/>
<input type="radio"/> Calle: <input type="text"/>	<input type="radio"/> No: <input type="text"/> Apto: <input type="text"/>
<input type="radio"/> Entre: <input type="text"/>	<input type="radio"/> Localidad: <input type="text"/>
<input type="radio"/> Teléfono: <input type="text"/>	
<input type="button" value="Reservar"/> <input type="button" value="Limpiar"/> <input type="button" value="Salir"/>	

1.4 Necesidad de los Sistemas de Gestión de Servicios.

El desarrollo creciente de las Tecnologías de Información y las Comunicaciones en la actualidad ha traído como consecuencia que la mayoría de las organizaciones y empresas hagan uso de las mismas para el desarrollo de sus actividades cotidianas.

Cuando se trata de grandes organizaciones que prestan varios servicios la gestión de información requiere el tratamiento, almacenamiento y difusión de grandes volúmenes de esta.

En estos casos los métodos manuales para gestionar adecuadamente la información referente a los servicios y responder con rapidez y agilidad a las demandas así como optimizar la comunicación con los demás sectores de la institución, no son viables, por lo que surge la necesidad de automatizar los procesos de gestión de préstamo de servicios.

1.5 ¿Qué ventajas posibilitaría un Sistema de Gestión de Préstamo de Servicios en el Centro Rector de Universidad para Todos?

Para el Centro Rector del Claustro de Universidad para Todos contar con un sistema de gestión de los servicios que se prestan significaría:

1. Control, planeamiento y administración de los servicios prestados a usuarios de diversas categorías, integrando informaciones de las diferentes áreas de la institución.
2. Incremento de la eficiencia de los servicios que se prestarán.
3. Permitir registrar los servicios prestados a los diversos usuarios de la institución, posibilitando cuantificar el tiempo empleado para cada servicio.

1.6 Análisis de las tecnologías y software.

Para un mejor desarrollo del sistema de gestión de préstamo de servicios que se desea automatizar en el Centro Rector de Universidad para Todos y los Canales Educativos, se ha realizado un profundo análisis de las tecnologías y herramientas más usadas en la elaboración de servicios web. A continuación se presentan las tecnologías y software empleados y varias definiciones esenciales en este trabajo.

1.6.1 ¿Por qué un Servicio Web?

Un servicio web es una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones. Es un componente de software que se comunica con otras aplicaciones codificando los mensajes en XML y enviando estos mensajes a través de protocolos estándares de Internet tales como el Hypertext Transfer Protocol (HTTP). Los servicios web exponen funcionalidad útil a los usuarios web mediante un protocolo web estándar. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar estos servicios para intercambiar datos en redes de ordenadores.

Razones para crear servicios web:

Se decidió crear un servicio web ya que aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen, permite que

servicios y software de diferentes compañías geográficamente distantes puedan ser combinados fácilmente para proveer servicios integrados, además de que fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.

La principal razón para usar servicios Web es que se basan en HTTP sobre TCP (Transmission Control Protocol) en el puerto 80, por la simple razón de que este puerto no se encuentra bloqueado por las organizaciones que protegen sus redes mediante firewalls que cierran casi todos los puertos salvo el 80 que es el que usan los navegadores. Otra razón para usar estos servicios es que son muy prácticos ya que pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De esta forma, los cambios a lo largo del tiempo en uno no deben afectar al otro.

1.6.1.1 XML.

Es un lenguaje desarrollado por el W3 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 [6]. Sirve para que muchos programas interpreten bien cualquier tipo de dato y se comuniquen entre ellos sin intervención humana. Constituye el lenguaje base de los Servicios Web.

Este es un lenguaje extensible de etiquetas estandarizado para describir datos. Es un metalenguaje que permite diseñar un lenguaje propio de etiquetas para múltiples clases de documentos. Los documentos XML están compuestos por unidades de almacenamiento llamadas entidades, que contienen datos analizados o sin analizar.

1.6.1.2 SOAP

(Simple Object Access Protocol, Protocolo Simple de Acceso a Objetos). Especificación XML para la formación de los mensajes intercambiados entre los sistemas distribuidos y la red.

Es un protocolo que permite la comunicación entre aplicaciones a través de mensajes por medio de Internet. Es independiente de la plataforma, y del lenguaje. Está basado en XML y es la base principal de los servicios web [3].

Brinda la posibilidad de transportar un mensaje, un documento XML, utilizando cualquier protocolo capaz de transmitir texto, ya que no se encuentra fuertemente asociado a ningún protocolo de transporte, además permite la interoperabilidad entre múltiples entornos.

1.6.2 ¿Por qué PHP?

Se seleccionó PHP como lenguaje de programación a utilizar ya que es uno de los más extendidos en la red de redes y ha sido aceptado precisamente por la simplicidad y potencia que lo caracteriza, ofrece gran variedad de funciones para la explotación de bases de datos sin grandes complicaciones. El mismo se ejecuta del lado del servidor, pues reúne las mejores capacidades funcionales para la creación del Servicio Web, además existe mucha información, documentación y perfecta integración del Apache-PHP-MySQL y cualquiera con experiencia en lenguajes del estilo C podrá entenderlo rápidamente porque posee una sintaxis clara y bien definida similar a la de este lenguaje.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como GNU/Linux), Windows y Mac OS X, y puede interactuar con los servidores web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI. Su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU. [4]

En resumen es un lenguaje fácil de aprender y de aplicar, consume pocos recursos y con gran rapidez de ejecución, contiene funciones para trabajar virtualmente con todas las tecnologías para la web existentes hoy, por lo que es muy empleado para el desarrollo de servicios web.

1.6.3 Sistema Gestor de base de datos (SGBD)

¿Por qué MySQL?

Los sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, con el propósito de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información.

Se escogió este Sistema Gestor de Base de Datos porque es muy potente, es un sistema multihilo y multiusuario con más de seis millones de instalaciones. Posee una gran escalabilidad por lo que es posible manipular bases de datos enormes, está escrito y desarrollado en C y C++ y probado con multitud de compiladores.

1.7 Fundamentación de la metodología a utilizar.

La metodología en el software, monitorea y permite el proceso de estandarizar los métodos y las técnicas necesarios para llevar a cabo el desarrollo de productos software de forma segura y guiada hacia un objetivo. Es una guía que muestra la organización y rol que se debe adoptar por cada una de las personas que componen un equipo de desarrollo mientras se balancean los requerimientos del negocio, el tiempo al mercado y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura.

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se lleva una metodología de por medio, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. A nivel mundial actualmente en dependencia del tiempo de vida y la complejidad del proyecto que se vaya a desarrollar se proponen diferentes metodologías, pudiendo ser tradicionales o ágiles, dentro de las tradicionales se encuentra RUP, que es la que se pretende utilizar en el presente trabajo.

1.7.1 Desarrollo basado en RUP.

Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP), constituye la metodología tradicional estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos, por su eficiencia y calidad obtenida a lo largo del ciclo de vida del software, ya que traza una mejor y completa línea de trabajo, basándose en un conjunto de metodologías adaptables al contexto y necesidades de cada organización, por lo que se puede decir que no es un sistema con pasos firmemente establecidos[5].

Es una metodología que permite una interacción continua y clara con el cliente, evitando construir Sistemas de Información que no están acorde a las expectativas finales. Unifica los mejores elementos creados por algunas metodologías existentes en el desarrollo del software, preparados para desarrollar grandes y complejos proyectos de envergadura, orientado a objetos, utiliza el UML como lenguaje de representación visual, se caracteriza por ser guiado por casos de uso, iterativo e incremental, y centrada en la arquitectura.

Divide el proceso de desarrollo en ciclos de vida, obteniendo un producto al final de cada ciclo, los cuales se dividen en fases que deben de terminar con un hito, dentro de estas fases se encuentran: Inicio, Elaboración, Construcción y Transición.

En resumen proporciona una guía para ordenar las actividades del equipo de trabajo, dirige las tareas de cada desarrollador por separado y del equipo como un todo, especifica los artefactos que deben desarrollarse y ofrece criterios para el control y la medición de los productos y actividades de proyectos.

1.7.2 UML.

Desde hace mucho tiempo la representación de los diseños de forma gráfica ha sido muy útil en cualquier rama de ingeniería o arquitectura, por lo que se necesita un lenguaje no sólo para comunicar las ideas a otros desarrolladores sino también para servir de apoyo en los procesos de análisis de un problema, en la informática con este objetivo se creó el Lenguaje Unificado de Modelado (Unified Modeling Language), que es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, establece un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

Se seleccionó este lenguaje porque permite modelar elementos conceptuales como lo son procesos de negocio y funciones de sistema, elementos concretos como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. Es un lenguaje consolidado que permite una comunicación fluida entre los diversos actores acerca del modelo.

1.8 Herramientas a utilizar.

Para llevar a cabo este trabajo han sido de gran utilidad herramientas como las que se fundamentan a continuación.

1.8.1 Visual Paradigm.

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por una computadora) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas. Dentro de estas herramientas una de la que más se destaca y se utiliza hoy en día es: Visual Paradigm.

Se decidió trabajar con esta herramienta porque está especializada en la ingeniería del software de bases de datos. Permite gestionar proyectos muy complejos con gran sencillez. Es una herramienta que da soporte al modelado visual con UML 2.0, presentando un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad, usa un lenguaje estándar común a todo el equipo de desarrollo y facilita la comunicación, tiene capacidades de ingeniería directa (versión profesional) e inversa, además de modelos y códigos que permanecen sincronizados en todo el ciclo de desarrollo [6].

Para concluir este capítulo se puede decir que con la realización de una profunda investigación acerca de los sistemas de gestión de préstamos y el análisis de la situación actual del Centro Rector de Universidad para Todos se pudo comprender los beneficios que trae consigo la automatización de este módulo en esta institución. El estudio de las tendencias actuales de servicios de este tipo ayudó a emprender el desarrollo de la propuesta de solución.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Una de las actividades fundamentales que se desarrolla en este capítulo es la descripción detallada del proceso que se lleva a cabo en el área de la gestión de préstamo de servicios en el Centro Rector de Universidad para Todos, esto permitirá un mejor entendimiento del entorno que se presenta lo cual brindará las bases necesarias para una mejor comprensión de la propuesta de sistema.

En este capítulo se realiza una propuesta del sistema que se quiere implementar, luego de haber analizado otros sistemas, pero demostrando lo necesario que se hace implementar el nuestro. Se lleva a cabo la captura de los requisitos funcionales y no funcionales, dando el punto de partida para el desarrollo del sistema con las características y funcionalidades requeridas, por otra parte se realiza la descripción y representación de los casos de uso del negocio y del sistema.

2.1 Objetivos estratégicos de la organización

Dentro de los objetivos estratégicos que se persiguen con la automatización de la gestión de los servicios que se prestan en la Casona se encuentran:

- ✓ Analizar, recopilar, difundir y potenciar la información científica en todas las áreas del conocimiento.
- ✓ Garantizar la seguridad y protección de los medios con que cuenta la institución.

2.2 Descripción de los procesos que serán objeto de automatización.

Solicitar servicio.

Permitirá gestionar la solicitud de cualquiera de los servicios que presta la institución, tanto para usuarios habituales de la entidad como para invitados, además se logrará una mayor rapidez y eficiencia en este proceso, se pueden solicitar servicios como:

- ✓ Servicio de búsqueda y recuperación de información.
- ✓ Servicio de reproducción de documentos.
- ✓ Servicio de referencia bibliográfica.
- ✓ Servicio de hemeroteca.
- ✓ Servicio de formación de usuarios.
- ✓ Servicio de diseminación selectiva de la información.
- ✓ Servicio de TV.

- ✓ Servicios telemáticos.
- ✓ Servicios de reservación de locales.
- ✓ Servicio de diseño.
- ✓ Servicio de editorial.

Gestionar solicitud.

Brinda la posibilidad de aprobar o denegar las solicitudes pendientes al coordinador u otra persona autorizada de la entidad.

Mostrar reporte.

Posibilita la creación de un comprobante que contiene los datos de los servicios prestados por la entidad, para un mejor control de los medios de la entidad y de la distribución de los servicios.

Mostrar Estado de la Solicitud.

Le permite al usuario conocer si su solicitud ha sido aprobada o no.

Ejecutar Servicio.

Brinda la posibilidad de almacenar los datos del servicio una vez que este sea ejecutado por el cliente, además se puede aplazar o anular completamente la solicitud de un servicio determinado.

Actualizar Listado.

Facilita al cliente actualizar su lista de solicitudes, eliminando las que han sido denegadas.

2.3 Información que se maneja.

En el Centro Rector de Universidad para Todos para el proceso de realizar la solicitud de algún servicio el usuario debe autenticarse previamente, por lo que se registran datos personales como: carné de Identidad, nombre completo, dirección particular, número de teléfono y correo electrónico de los usuarios los cuales se encuentran almacenados en la base de datos de gestión de usuarios, pero además se guardarán los datos del servicio prestado, es decir, la fecha, el tiempo de duración de servicio, una breve descripción y un nombre del servicio. No constituye una gran cantidad de información pero hay que tener en cuenta que son muchos los usuarios de la entidad por lo que esta información se multiplica y su manejo puede llegar a ser tedioso y complejo.

2.4 Propuesta del sistema.

Con el objetivo de darle solución al problema planteado al inicio de esta investigación y luego de los estudios realizados, se propone dentro de este módulo el desarrollo de un servicio web con el objetivo de gestionar el préstamo de servicios en la Casona de 23 y B.

Se tuvo en cuenta esta solución partiendo en primer lugar de las ventajas que presentan los servicios web y considerando la situación actual de la institución, teniendo en cuenta el desarrollo de la intranet, a la cual se pretenden integrar cada uno de los módulos desarrollados bajo el lenguaje de programación PHP, en este caso se encuentra el préstamo de servicios.

Este sistema (Servicio Web) contendrá funciones que permitirán gestionar la solicitud de cualquier servicio de los que brinda la Casona de 23 y B, garantizando el control adecuado sobre los medios y los datos pertenecientes al servicio, además los usuarios podrán conocer el estado de sus solicitudes y actualizar el listado de estas. La entidad podrá conocer un reporte de los servicios prestados en cualquier momento, con sus datos principales.

Para llevar a cabo la realización de esta propuesta se desarrolló un estudio de los sistemas de préstamo de servicios en Cuba y en el mundo, los cuales quedaron reflejados en el capítulo 1, se debe destacar que no se detectaron servicios similares al que se desea implementar. Después de realizar un profundo análisis de cada uno de los sistemas relacionados en el estado del arte de nuestra investigación, se decide llevar a cabo la implementación de un nuevo servicio que cumpla con las expectativas de los clientes, ya que por ejemplo, en el caso de Red Municipal de Bibliotecas de Murcia (RMBM) solo se prestan servicios bibliotecarios, y en la Casona 23 y B además de esos servicios se prestan equipos, recursos y servicios de hospedaje entre otros. En el caso de Infomed, se prestan servicios de correo electrónico y de biblioteca por lo que tampoco cumple las características deseadas y lo mismo pasa con el sistema de pase de la UCI, ya que solo prestan servicios de reservación de transporte.

En aras de satisfacer los requerimientos planteados se propone un módulo que brinde la opción a los clientes de reservar cualquier tipo de servicio de los prestados en la Casona 23 y B, y que también permita a los usuarios pertinentes ver la información requerida por ellos a modo de reportes.

2.5 Modelo de dominio

Como consecuencia del bajo nivel de estructuración que posee el negocio que se está estudiando, se propone la realización de un modelo de dominio, ya que permite de manera visual mostrar al usuario los

principales conceptos que se manejan en el dominio del sistema en desarrollo. Además brinda a los usuarios, clientes, desarrolladores e interesados, un vocabulario común para poder entender el contexto en que se emplaza el sistema. Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

Se realiza la descripción del modelo de dominio mediante un diagrama de clases UML donde se especifican las principales clases conceptuales que pueden intervenir en el sistema, estos representarán los objetos que existen o eventos que suceden en el entorno en el que trabajará el mismo. A continuación se procederá a explicar los principales conceptos que se identifican en este modelo.

Casona 23 y B: Centro Rector del Claustro de Universidad Para Todos.

Intranet: Portal Web que contiene las publicaciones del Centro del Claustro de Universidad Para Todos.

Web Service: Servicios telemáticos brindados en el portal de la Intranet.

Usuario: es la persona que visita la entidad y se beneficia con sus servicios.

Control de Acceso: módulo que facilita el control de las entradas y salidas del personal usuario de los servicios de la entidad.

Gestión de Usuario: módulo que gestiona toda la información de los usuarios y el sistema de servicios de la entidad.

Control de Medios Básicos: módulo que controla los Activos Fijos Tangibles con que cuenta la entidad.

Préstamo de servicios: módulo que gestiona los servicios de la entidad y la información referente a estos.

Reproducción de documentos: Impresión y fotocopiado de documentos.

Diseño: servicio que se encarga del diseño de interfaces, publicaciones en todos los soportes y composición de dichas publicaciones.

Hemeroteca: Biblioteca en que principalmente se guardan y sirven al público diarios y otras publicaciones periódicas, digitales o no.

Editorial: servicio que se encarga tanto de la revisión, corrección y composición como de la impresión de libros y otros documentos.

Televisión: Provee canales, paquetes, programas y banco de imágenes a pedido, así como salones de visionaje.

Telemáticos: Correo, internet y servicios conexos.

Formación de usuarios: Entrenamientos integrales y especializados para servicios de la Intranet de la casona.

Diseminación Selectiva de la información: servicio que se encarga de la emisión de información, sobre una temática determinada por el usuario, personalizada, a pedido y en su destino.

Referencia bibliográfica: servicios de cita e indexación científica.

Búsqueda y recuperación de la información: búsqueda de información de acuerdo a las temáticas de interés del usuario.

Servicio de reservación: consiste en la reservación de algunos de los locales de trabajo con que cuenta la entidad.

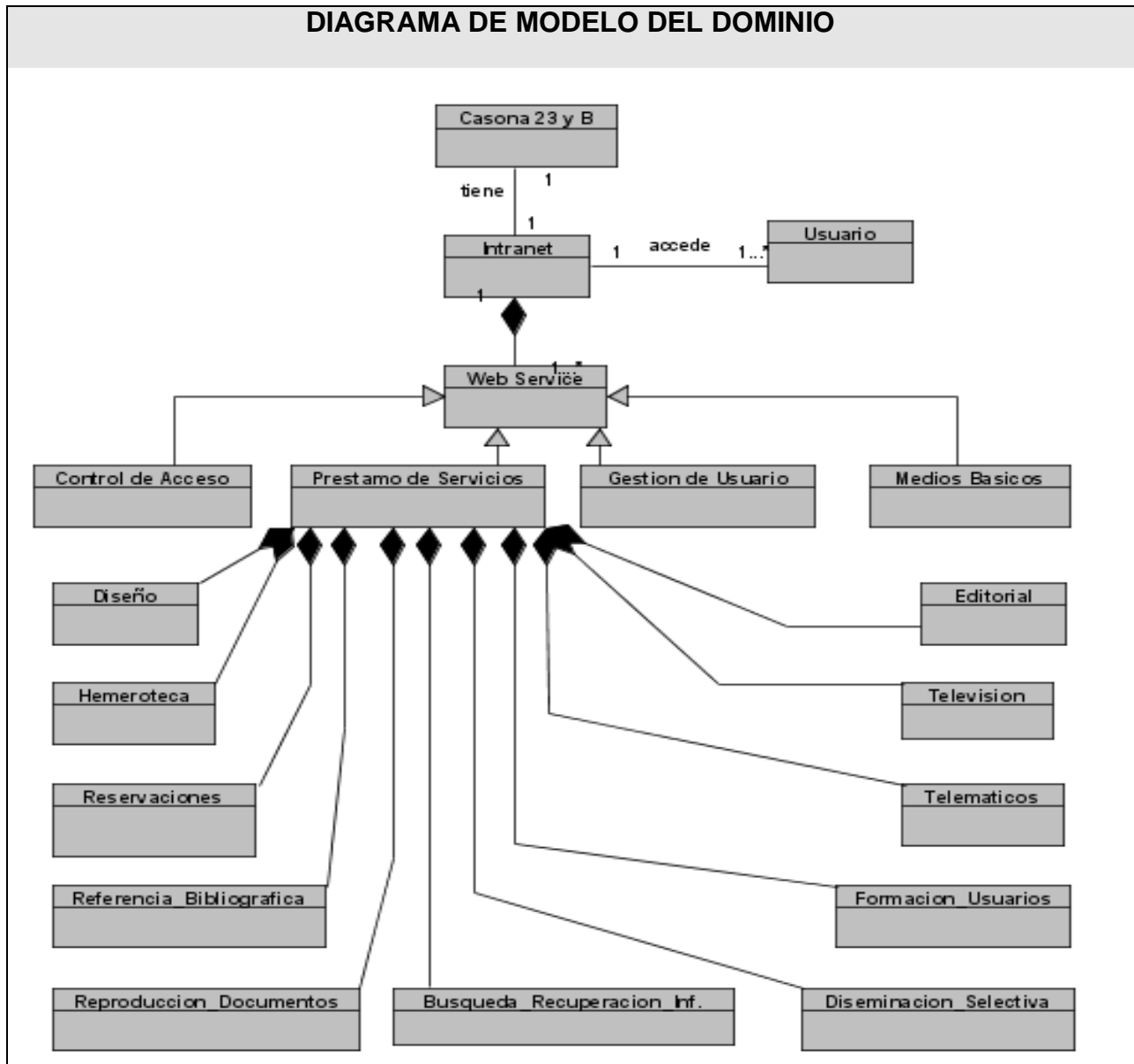


Figura 2.1 Modelo de dominio.

2.6 Modelo del sistema.

Para una mejor comprensión se enumeran a continuación los requerimientos funcionales los cuales se modelan en término de casos de uso del sistema, así como los no funcionales. Además, se desarrolla un análisis de los principales casos de uso del sistema.

2.6.1. Requerimientos funcionales.

RF-1 Solicitar servicio de búsqueda y recuperación de información.

- RF1.1- Solicitar servicio de consulta de recursos.
- RF1.2- Solicitar servicio de préstamo de documentos.
- RF1.3- Solicitar servicio de monitoreo.
- RF1.4- Solicitar servicio de observatorio.
- RF1.5- Crear solicitud.

RF-2 Solicitar servicio de reproducción de documentos.

- RF2.1- Solicitar servicio de impresión.
- RF2.2- Solicitar servicio de fotocopias.
- RF2.3- Solicitar servicio de digitalización de documentos.
- RF2.4- Solicitar servicio de transferencia de formato.
- RF2.5- Crear solicitud.

RF-3 Solicitar servicio de referencia.

- RF3.1- Solicitar servicio de referencia en línea.
- RF3.2- Solicitar servicio de referencia en sala.
- RF3.3- Crear solicitud.

RF-4 Solicitar servicio de hemeroteca.

- RF4.1- Solicitar servicio de hemeroteca en línea.
- RF4.2- Solicitar servicio de hemeroteca en sala.
- RF4.3- Crear solicitud.

RF-5 Solicitar servicio de formación de usuarios.

- RF5.1- Solicitar servicio de cursos.
- RF5.2- Solicitar servicio de talleres.
- RF5.3- Solicitar servicio de asesoramiento en sala.
- RF5.4- Crear solicitud.

RF-6 Solicitar servicio de diseminación selectiva.

- RF6.1- Solicitar servicio de edición de boletines.
- RF6.2- Solicitar servicio de paquetes de información.
- RF6.3- Crear solicitud.

RF-7 Solicitar servicio de televisión.

- RF7.1- Solicitar servicio de publicación de carteleras de canales de TV.
- RF7.2- Solicitar servicio de transmisión de programas en vivo.
- RF7.3- Solicitar servicio de grabación de programas.
- RF7.4- Solicitar servicio de digitalización de programas.
- RF7.5- Solicitar servicio de reproducción de programas.
- RF7.6- Crear solicitud.

RF-8 Solicitar servicios telemáticos.

- RF8.1- Solicitar servicio de creación de usuarios.
- RF8.2- Solicitar servicio de correo electrónico.
- RF8.3- Solicitar servicio de acceso a navegación.
- RF8.4- Solicitar servicio de conexión de red inalámbrica en sala de trabajo.
- RF8.5- Crear solicitud.

RF-9 Solicitar servicios de reservación.

- RF9.1- Solicitar servicio de tiempo de máquina.
- RF9.2- Solicitar servicio de PC portátil fuera de la entidad.
- RF9.3- Solicitar servicio de PC portátil en sala.
- RF9.4- Solicitar servicio de locales de trabajo.
- RF9.5- Crear solicitud.

RF-10 Solicitar servicios de diseño.

- RF10.1- Solicitar servicio de diseño de plantillas.
- RF10.2- Solicitar servicio de diseño de interfaz (web, multimedia, DVD, etc....).
- RF10.3- Solicitar servicio de diseño editorial (portada de libros, tabloides, DVD).
- RF10.4- Solicitar servicio de Identidad visual.
- RF10.5- Crear solicitud.

RF-11 Solicitar servicios de editorial.

- RF11.1- Solicitar servicio de redacción.
- RF11.2- Solicitar servicio de corrección.
- RF11.3- Crear solicitud.

RF-12 Gestionar solicitud.

- RF-12.1 Aprobar la solicitud.

RF-12.2 Denegar la solicitud.

RF-12.3 Modificar estado del servicio.

RF-13 Mostrar estado de solicitud.

RF-13.1 Devolver un listado mostrando el estado de las solicitudes hechas por el usuario.

RF-14 Mostrar reporte.

RF-15 Ejecutar servicio.

RF-15.1 Prestar servicio.

RF-15.2 Aplazar solicitud de servicio.

RF-15.2 Anular solicitud de servicio.

RF-16 Actualizar Listado de solicitudes.

RF-16.1 Eliminar solicitudes.

2.6.2. Requerimientos no funcionales.

1. Software

- ✓ El Servicio Web requiere tener instalado el lenguaje de programación PHP 5.0
- ✓ El Servicio Web utilizará como base de datos MySQL.
- ✓ Se utilizará como servidor web Apache 2.0.

2. Seguridad y Privacidad

- ✓ Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.

3. Portabilidad

- ✓ Se podrá acceder a los servicios desde cualquier protocolo de transporte que sea capaz de transportar texto ejemplo (http, https, SMTP, etc.)
- ✓ El diseño del servicio web permite que pueda funcionar tanto en sistemas operativos libres como propietarios.

4. Hardware

- ✓ Requiere como mínimo de RAM 512 MB.

5. Restricciones en el diseño y la implementación

- ✓ Lenguaje de programación PHP.

6. Usabilidad

- ✓ El Servicio Web podrá ser consumido desde diferentes plataformas de desarrollo pues su protocolo de comunicación está basado en XML compatible para todas las plataformas.

7. Legales

- ✓ El Servicio Web y toda la documentación generada pertenecen al proyecto del Centro Rector del Claustro de Universidad para Todos y a la Universidad de las Ciencias Informáticas.

8. Confiabilidad

- ✓ El Servicio Web debe estar disponible las 24 horas de forma tal que se pueda acceder a todas sus funcionalidades.

2.7 Definición de los casos de uso

2.7.1 Actores del sistema.

El actor del sistema es la intranet, la cual es la que interactúa con nuestro servicio, haciendo las peticiones de los servicios que se brindan. En la tabla siguiente se muestra su justificación.

Tabla 2-1. Actor del sistema.

Actores del sistema	Justificación
Intranet	La Intranet es el actor que inicia los casos de uso invocando los servicios que permiten ejecutarlos beneficiándose de ellos.

2.7.2 Diagrama de casos de uso del sistema.

Los casos de uso son fragmentos de funcionalidad del sistema. En ellos se describe la secuencia determinada de eventos que realiza un actor en interacción con la aplicación, en nuestro caso con el Servicio Web. En el diagrama general de casos de uso de este sistema (Ver Fig.2-1.Diagrama General de **CUS**), el actor intranet se relaciona con el paquete Servicio Web “Préstamo de servicios”, el cual encapsula los casos de usos de sistema.

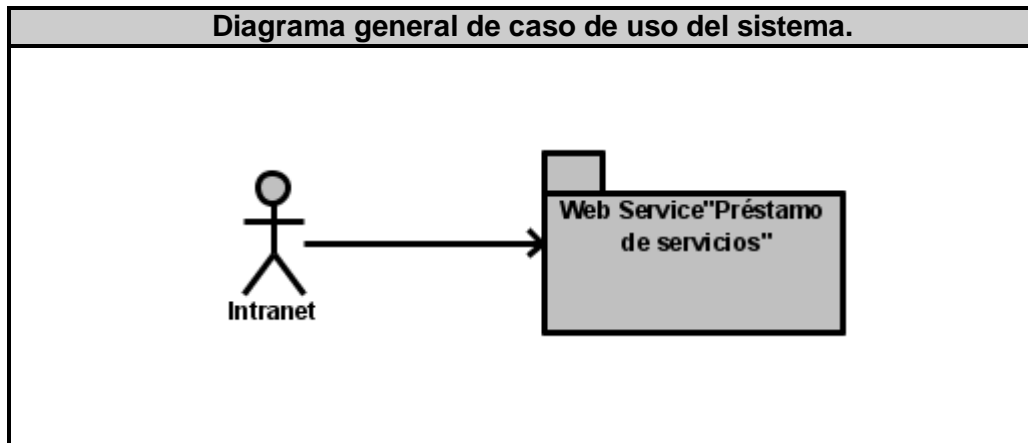


Fig.2-2. Diagrama General de CUS



Fig. 2-3. Diagrama de CUS

2.7.3 Casos de uso del sistema. (CUS)

Tabla 2-2. Breve descripción del CUS. Solicitar servicio.

CU-1	Solicitar servicio.
Actor(s)	Intranet
Descripción	El mismo se inicia cuando el actor invoca el vínculo que permite solicitar algún servicio, este posibilita al usuario hacer uso de los medios con que cuenta la institución, permite la solicitud de servicios de búsqueda y recuperación de información, de reproducción de documentos, de referencia, de hemeroteca, de formación de usuarios, de diseminación selectiva, de televisión, servicios telemáticos, de diseño y además de editorial.
Referencia	RF1 RF2, RF3, RF4, RF5, RF6, RF7, RF8, RF9, RF10, RF11.

Tabla 2-3. Breve descripción del CUS. Mostrar estado de solicitud.

CU-2	Mostrar estado de solicitud.
Actor(s)	Intranet
Descripción	El mismo se inicia cuando el actor invoca el vínculo de mostrar estado, con el objetivo de conocer si su solicitud ha sido aprobada o no por parte de los directivos de la entidad. De esta forma sabrá si puede hacer uso del servicio solicitado.
Referencia	RF-13.

Tabla 2-4. Breve descripción del CUS. Gestionar solicitud.

CU-3	Gestionar solicitud.
Actor(s)	Intranet
Descripción	El mismo se inicia cuando el actor invoca el vínculo que permite aprobar o no alguna solicitud pendiente, le brinda la posibilidad al directivo de la entidad que este encargado de prestar o no un servicio a los usuarios, teniendo en cuenta varios criterios se aprueba o no la solicitud y para finalizar el caso de uso se modifica el estado del servicio.
Referencia	RF-12, RF-12.1, RF-12.2, RF-12.3.

Tabla 2-5. Breve descripción del CUS. Mostrar reporte.

CU-4	Mostrar reporte.
Actor(s)	Intranet
Descripción	El mismo se inicia cuando el actor desea conocer los diferentes servicios prestados e invoca el vínculo de mostrar reporte que le permite llevar un control de los servicios prestados y con ello controla además los medios con que cuenta.
Referencia	RF-14.

Tabla 2-6. Breve descripción del CUS. Ejecutar Servicio.

CU-5	Ejecutar Servicio.
Actor(s)	Intranet
Descripción	El mismo se inicia cuando el usuario ejecute algún servicio autorizado previamente, el actor invoca el vínculo de ejecutar servicio que le permite almacenar en la base de datos del sistema los datos de la solicitud, una vez que el usuario ya hizo uso del servicio solicitado.
Referencia	RF-15.

Tabla 2-7. Breve descripción del CUS. Actualizar listado.

CU-6	Actualizar listado.
Actor(s)	Intranet
Descripción	El mismo se inicia cuando el usuario desee actualizar su listado de solicitudes, permitiendo que elimine las solicitudes antiguas y que hayan sido denegadas.
Referencia	RF-16, RF-16.1.

Tabla 2-8. Casos de uso por ciclo.

Ciclo de desarrollo primario.			
Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-1	Solicitar servicio.	Servicio Web "Préstamo de servicios".	Servicios prioritarios que constituyen el mecanismo básico de acceso a los datos fundamentales del sistema.
CU-2	Mostrar estado de solicitud.		
CU-3	Gestionar solicitud.		
CU-4	Mostrar reporte		
CU-5	Ejecutar Servicio		
CU-6	Actualizar listado		

Tabla 2-9. Descripción expandida CUS. Solicitar servicio.

CU-1	Solicitar servicio.
Propósito	Brinda las funciones que permiten solicitar cualquier servicio de los que brinda la institución siempre que sea un usuario real de la misma.
Actores:	Intranet (inicia)
Resumen:	El mismo se inicia cuando el actor invoca el vínculo que permite solicitar algún servicio, este posibilita al usuario hacer uso de los medios con que cuenta la institución, permite la solicitud de servicios de búsqueda y recuperación de información, de reproducción de documentos, de referencia, de hemeroteca, de formación de usuarios, de diseminación selectiva, de televisión, servicios telemáticos, de diseño y además de editorial.
Referencias	RF1 RF2, RF3, RF4, RF5, RF6, RF7, RF8, RF9, RF10, RF11.
Acción del actor	
Respuesta del sistema	

<p>1. La intranet puede realizar la solicitud de cualquier servicio de los que brinda la entidad, enviando parámetros como:</p> <p>a) Fecha.</p> <p>b) Duración.</p> <p>c) Descripción.</p>	<p>2. Verifica que los datos obligatorios hayan sido enviados.</p> <p>3. La solicitud se almacena en la base de datos.</p> <p>4. Envía un mensaje indicando que la solicitud fue almacenada correctamente. Finaliza el CU.</p>
Curso alternativo.	
Acción del actor	Respuesta del sistema.
	<p>2. Si existe algún dato obligatorio que no haya sido llenado, se retorna un mensaje indicándolo.</p>

Tabla 2-10.Descripción expandida CUS. Mostrar estado de solicitud.

CU-2	Mostrar estado de solicitud.
Propósito	Permite que el usuario conozca si su solicitud ha sido aprobada o no.
Actores:	Intranet (inicia)
Resumen:	El mismo se inicia cuando el actor invoca el vínculo de mostrar estado, con el objetivo de conocer si su solicitud ha sido aprobada o no por parte de los directivos de la entidad. De esta forma sabrá si puede utilizar el servicio solicitado.
Referencias	RF-13.
Acción del actor	Respuesta del sistema

1. La intranet invoca el vínculo que le permite ver el estado de la solicitud, para esto envía el identificador del usuario, es decir su número de carné de identidad.	2. Verifica que el identificador es correcto. 3. Verifica que el campo de búsqueda no esté vacío. 4. Busca en la base de datos las solicitudes realizadas por el usuario. 5. Devuelve un listado con las solicitudes realizadas por el usuario con sus estados correspondientes. Finaliza el CU.
Curso alternativo.	
Acción del actor	Respuesta del sistema.
	2. Se retorna un mensaje indicando que el identificador del usuario es incorrecto. 3. Si existe algún dato obligatorio que no haya sido llenado, se retorna un mensaje indicándolo. 4. Si no existen solicitudes hechas por ese usuario se retorna un mensaje indicándolo y finaliza el caso de uso.

Tabla 2-11.Descripción expandida CUS. Gestionar solicitud.

CU-3	Gestionar solicitud.
Propósito	Permite que se apruebe o no la solicitud de servicio hecha por algún usuario.
Actores:	Intranet (inicia)
Resumen:	El mismo se inicia cuando el actor invoca el vínculo que permite aprobar o no alguna solicitud pendiente, le brinda la posibilidad al directivo de la entidad que esté encargado de prestar o no un servicio a los usuarios, teniendo en cuenta varios criterios se aprueba o no la solicitud y para finalizar el caso de uso se modifica el estado del servicio.
Referencias	RF-12.

Acción del actor	Respuesta del sistema
<p>1. La intranet invoca cualquiera de los servicios que le permite chequear las solicitudes:</p> <p>a) Si desea aprobar la solicitud ir a la sección "Aprobar solicitud".</p> <p>b) Si desea denegar la solicitud ir a la sección "Denegar solicitud".</p>	
Sección "Aprobar solicitud".	
Acción del actor.	Respuesta del sistema.
1.a) La intranet invoca el servicio de aprobar solicitud.	2. Actualiza en la base de datos el estado de la solicitud. Finaliza el caso de uso.
Curso alternativo.	
Acción del actor.	Respuesta del sistema.
Sección "Denegar solicitud".	
Acción del actor.	Respuesta del sistema.
1.b) La intranet invoca el servicio de denegar solicitud.	2. Actualiza en la base de datos el estado de la solicitud. Finaliza el caso de uso.
Curso alternativo.	
Acción del actor.	Respuesta del sistema.

Tabla 2-12.Descripción expandida CUS. Mostrar reporte.

CU-4	Mostrar reporte.	
Propósito	Permite que se lleve un control estadístico de los servicios que presta la entidad.	
Actores:	Intranet (inicia)	
Resumen:	El mismo se inicia cuando el actor desea conocer los diferentes servicios prestados e invoca el vínculo de mostrar reporte que le permite llevar un control de los servicios prestados y con ello controla además los medios con que cuenta.	
Referencias	RF-14.	
Acción del actor		Respuesta del sistema
1. La intranet invoca el vínculo que le permite Mostrar reporte, enviando la fecha de la que desea conocer el reporte.		2. Busca en la base de datos la información referente a cualquier servicio prestado. 3. Devuelve un reporte con los datos del servicio prestado. Finaliza el CU.
Curso alternativo.		
Acción del actor		Respuesta del sistema.

Tabla 2-13.Descripción expandida CUS. Ejecutar Servicio.

CU-5	Ejecutar Servicio.	
Propósito	Permite que se lleve un control estadístico de los servicios que presta la entidad.	
Actores:	Intranet (inicia).	
Resumen:	El mismo se inicia cuando el usuario ejecute algún servicio autorizado previamente, el actor invoca el vínculo de ejecutar servicio que le permite almacenar en la base de datos del sistema los datos de la solicitud, una vez que el usuario ya hizo uso del servicio solicitado.	
Referencias	RF-15.	
Acción del actor		Respuesta del sistema
1. La intranet invoca el vínculo que le permite ejecutar Servicio, enviando el identificador del servicio en cuestión.		2. Verifica la existencia de dicho servicio. 3. Busca en la base de datos la información referente al servicio especificado. 3. Actualiza el estado del servicio en la base de datos. Finaliza el CU.
Curso alternativo.		
Acción del actor		Respuesta del sistema.
		2. Envía un mensaje de error en caso de que no exista el servicio especificado. Finaliza el CU.

Tabla 2-14.Descripción expandida CUS. Actualizar Listado.

CU-6	Actualizar Listado.	
Propósito	Permite que el usuario actualice sus solicitudes de servicios.	
Actores:	Intranet (inicia)	
Resumen:	El mismo se inicia cuando el usuario desee actualizar su listado de solicitudes, permitiendo que elimine las solicitudes antiguas y que hayan sido denegadas.	
Referencias	RF-16, RF-16.1.	
	Acción del actor	Respuesta del sistema
	<p>1. La intranet invoca el vínculo que le permite actualizar listado de solicitudes, enviando el identificador del usuario que desea actualizar su listado.</p> <p>4. Elimina aquella solicitud que fue denegada y aún esta en su lista de solicitudes.</p>	<p>2. Verifica la validez del identificador de usuario.</p> <p>3. Busca en la base de datos la información referente a dicho usuario.</p> <p>3. Devuelve un listado de las solicitudes hechas por este usuario.</p> <p>5. Actualiza en la base de datos el listado de solicitudes del usuario. Finaliza el CU.</p>
	Curso alternativo.	
	Acción del actor	Respuesta del sistema.
		<p>2. Envía un mensaje de error en caso de que no en la base de datos el usuario especificado. Finaliza el CU.</p>

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

El presente capítulo se centra fundamentalmente en el diseño del sistema propuesto, se detallan los casos de uso identificados y se determinan las clases necesarias para llevar a cabo las funcionalidades del sistema. Además se representan los diagramas de interacción por escenario de cada caso de usos. Se brinda una breve descripción de las clases entidades y controladoras usadas, así como de las tablas de la base de datos, además se expone el diagrama de Entidad-Relación.

3.1 Diseño

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales, en esta etapa se desarrolla el modelo de diseño que no es más que un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos, junto con otras restricciones relacionadas con el entorno de implementación, influyen en el sistema a considerar.

3.1.1 Representación UML del Diagrama de clases del diseño.

Diagrama de Clases del Diseño del Caso de uso: Solicitar Servicio.

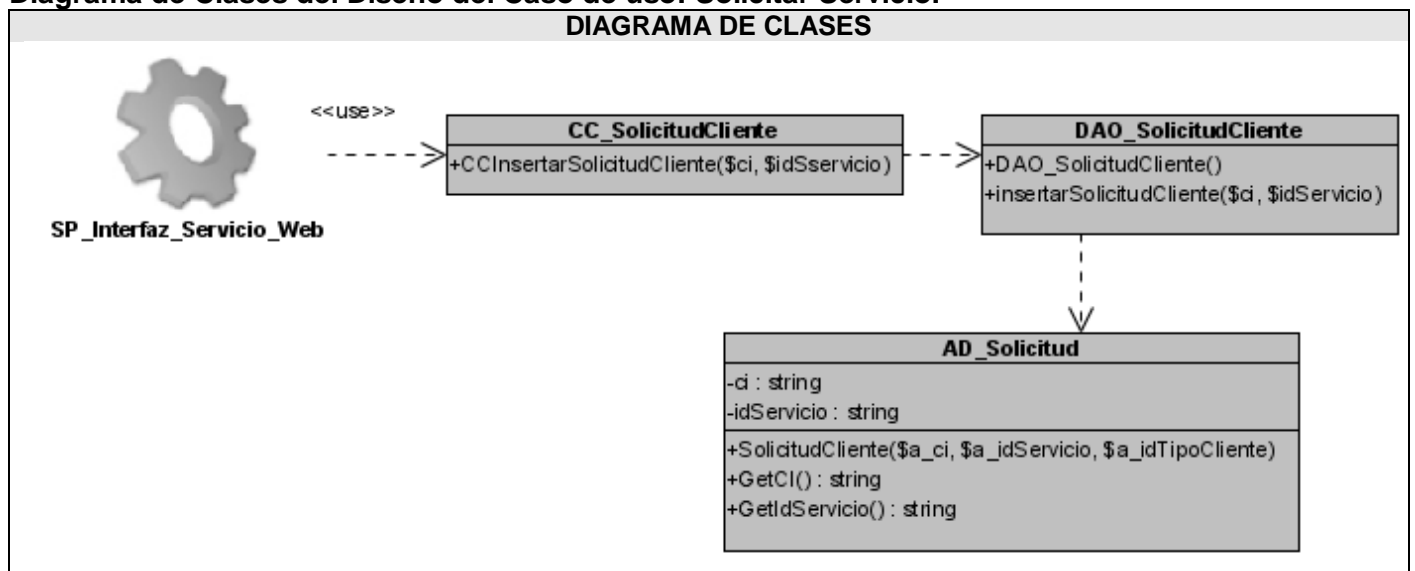


Fig.3-1.DCD. CU_ Solicitar_Servicio.

Diagrama de Clases del Diseño del Caso de uso: Mostrar estado de solicitud.

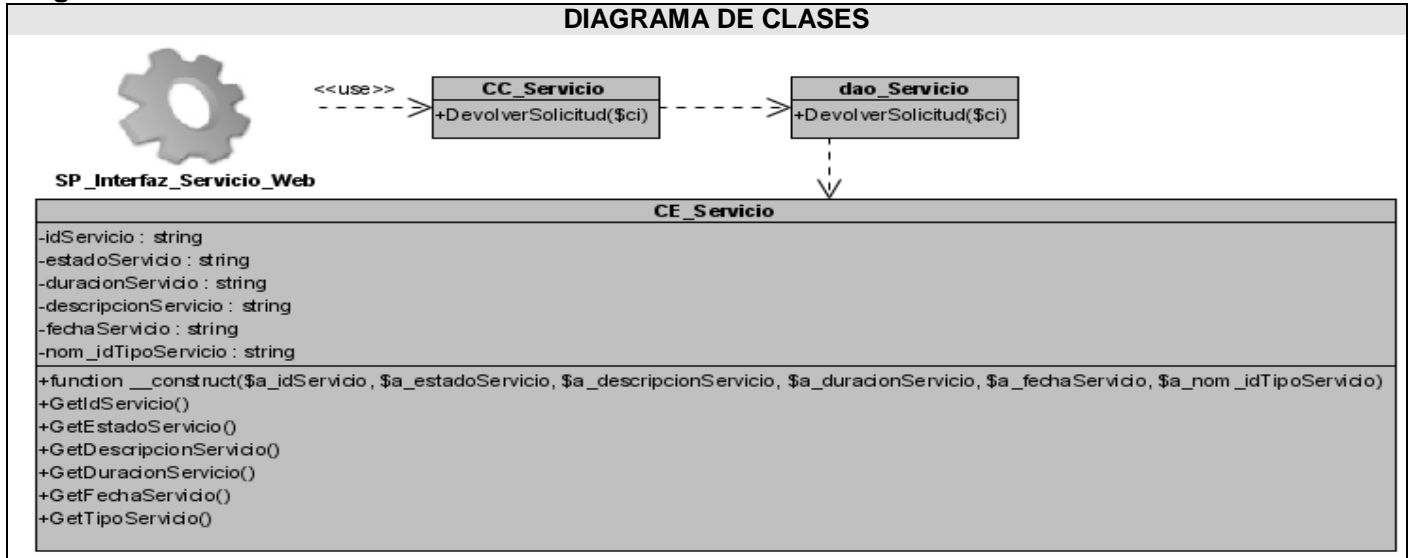


Fig.3-2.DCD. CU_Mostrar_Estado.

Diagrama de Clases del Diseño del Caso de uso: Gestionar Solicitud

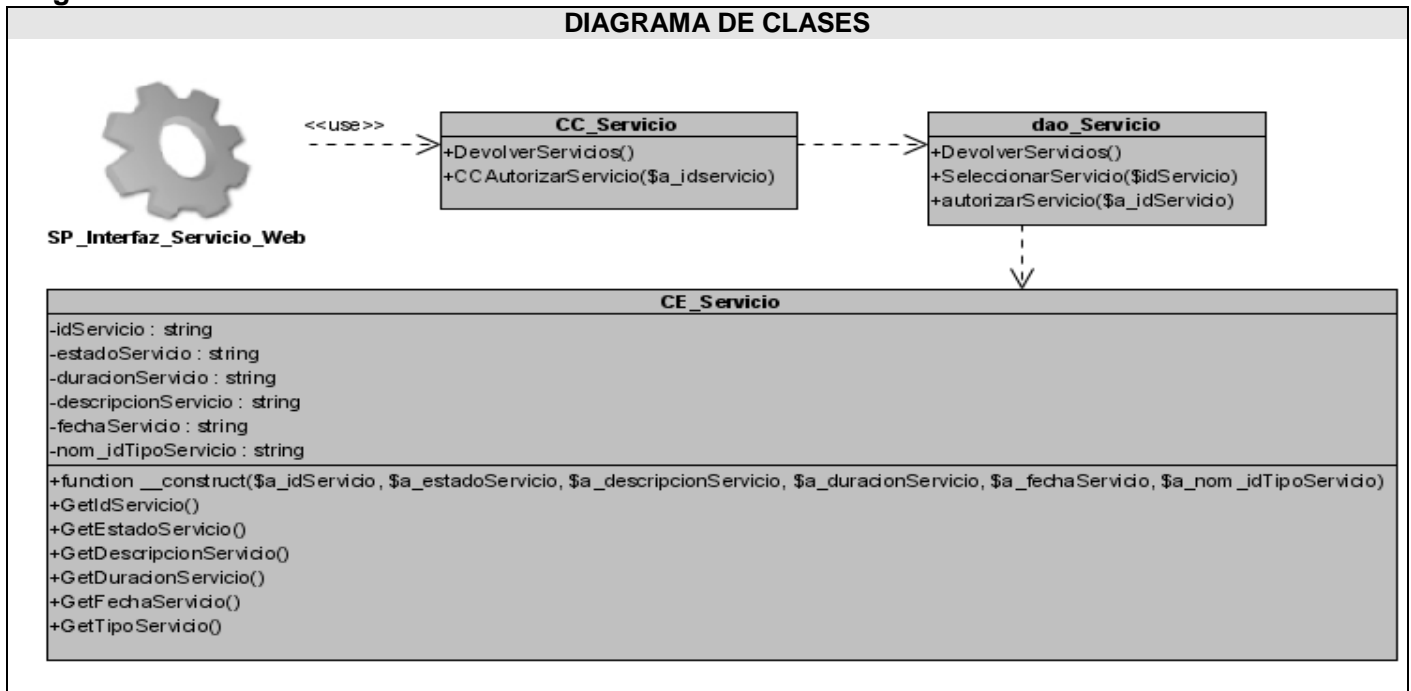


Fig.3-3.DCD. CU_Gestionar_Solicitud.

Diagrama de Clases del Diseño del Caso de uso: Mostrar Reporte.

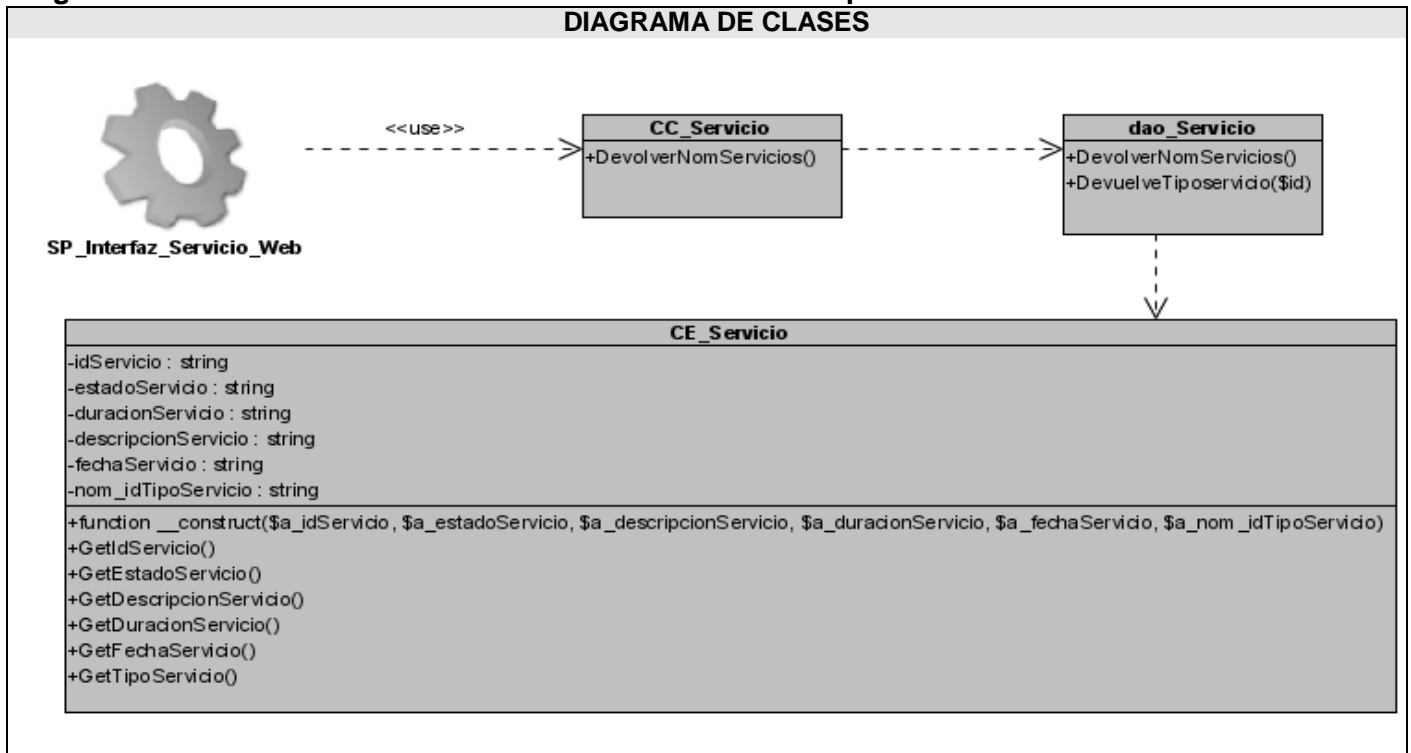


Fig.3-4.DCD. CU_Mostrar_Reporte.

Diagrama de Clases del Diseño del Caso de uso: Ejecutar Servicio.

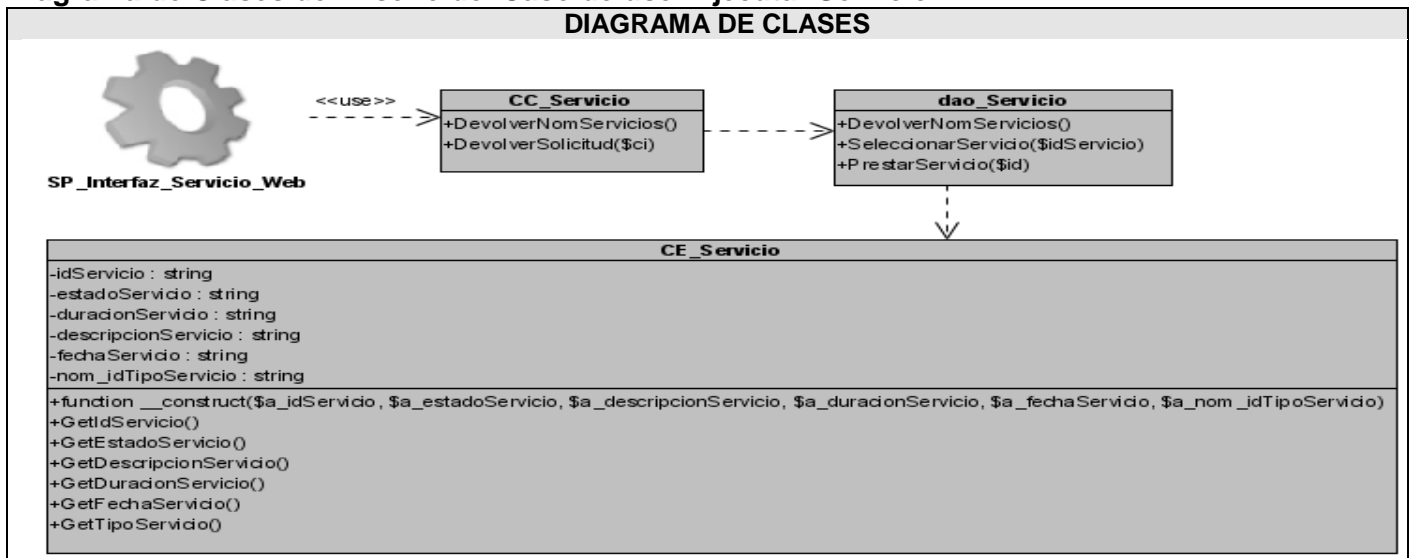


Fig.3-5.DCD. CU_Ejecutar_Servicio.

Diagrama de Clases del Diseño del Caso de uso: Actualizar Listado.

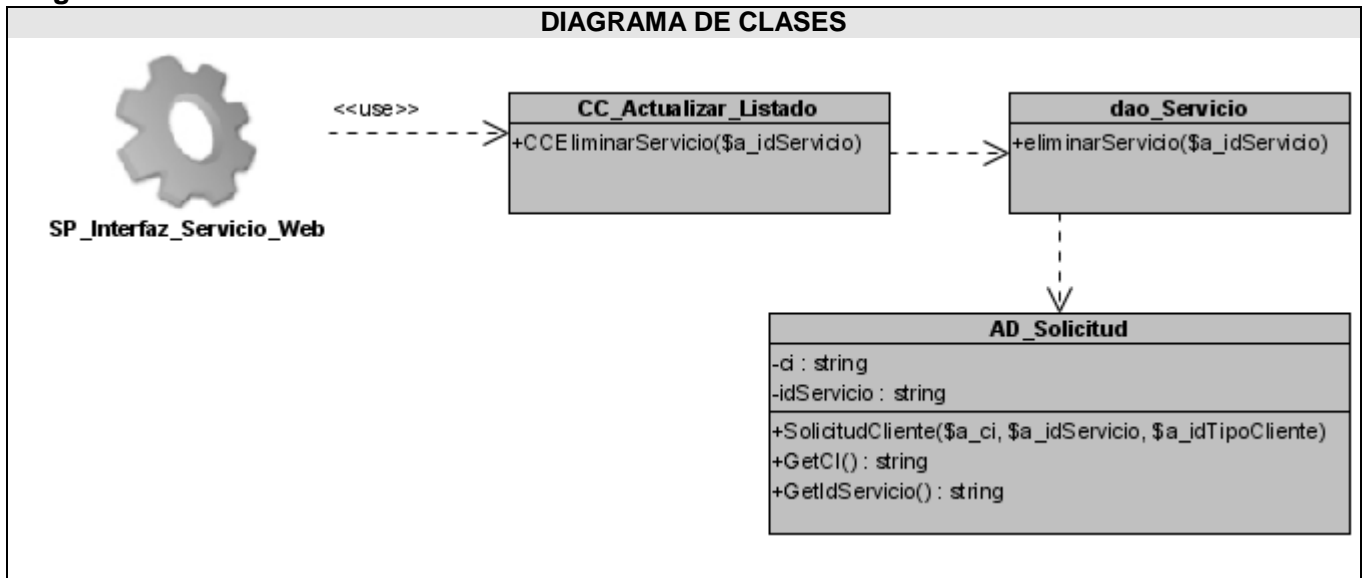


Fig.3-6.DCD. CU_ Actualizar_Listado.

3.1.2 Representación UML de los diagramas de interacción.

Un diagrama de interacción explica gráficamente las interacciones existentes entre las instancias (y las clases) del modelo de éstas. El punto de partida de las interacciones es el cumplimiento de las poscondiciones de los contratos de operación.

El UML define dos tipos de estos diagramas, ambos sirven para expresar interacciones semejantes o idénticas de mensaje:

- a. Diagramas de colaboración
- b. Diagramas de secuencia

Los diagramas de secuencia describen las interacciones en una especie de formato de cerca o muro, como se muestra a continuación:

Diagrama de Secuencia del Diseño del Caso de uso: Solicitar Servicio.

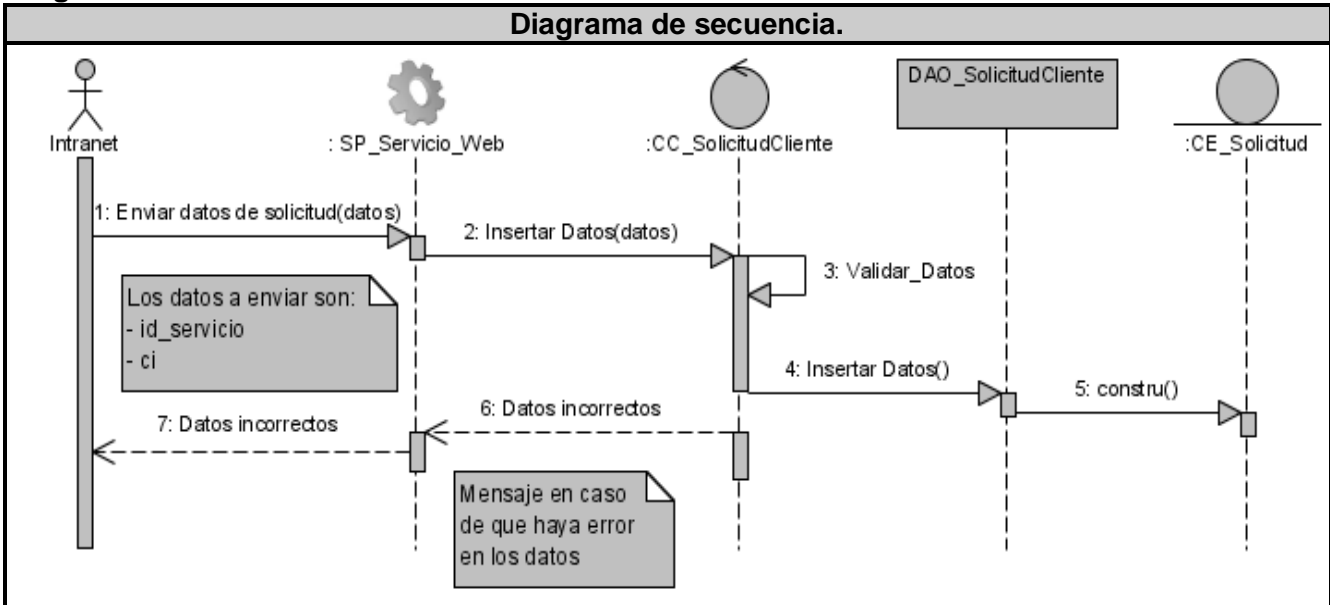


Fig.3-7. DS. CU_Solicitar_Servicio

Diagrama de Secuencia del Diseño del Caso de uso: Mostrar estado de solicitud.

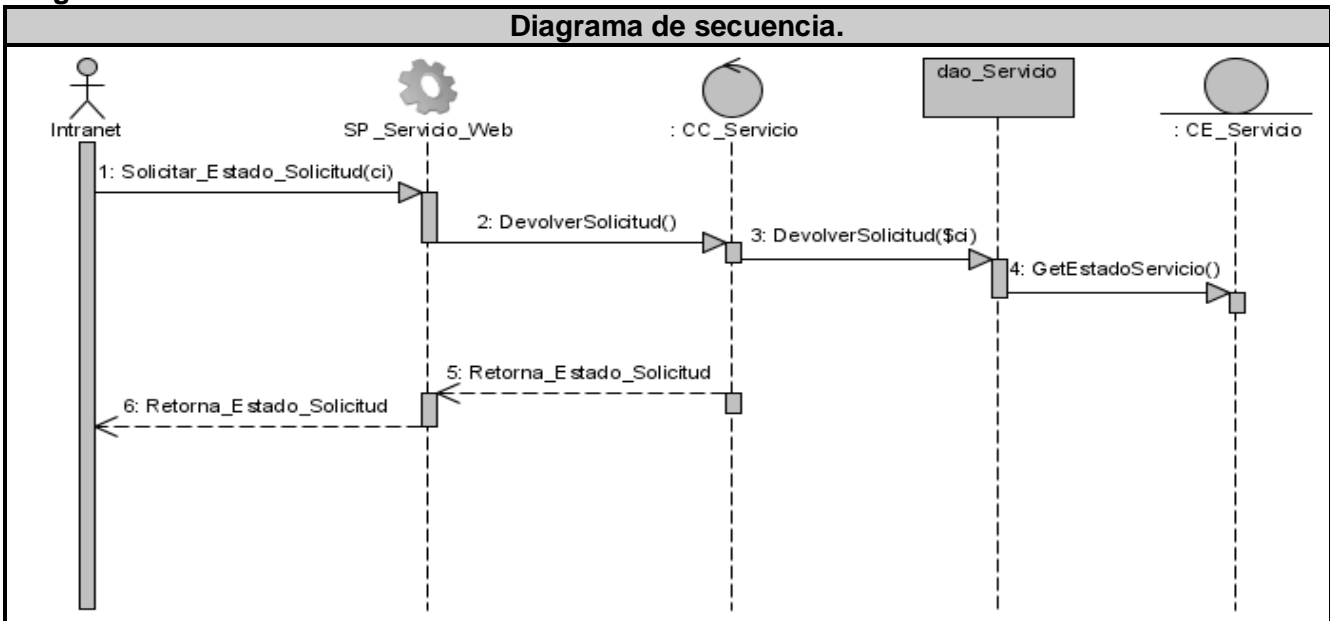


Fig.3-8.DS. CU_Mostrar_Estado_Solicitud

**Diagrama de Secuencia del Diseño del Caso de uso: Gestionar Solicitud
Escenario "Aceptar Solicitud".**

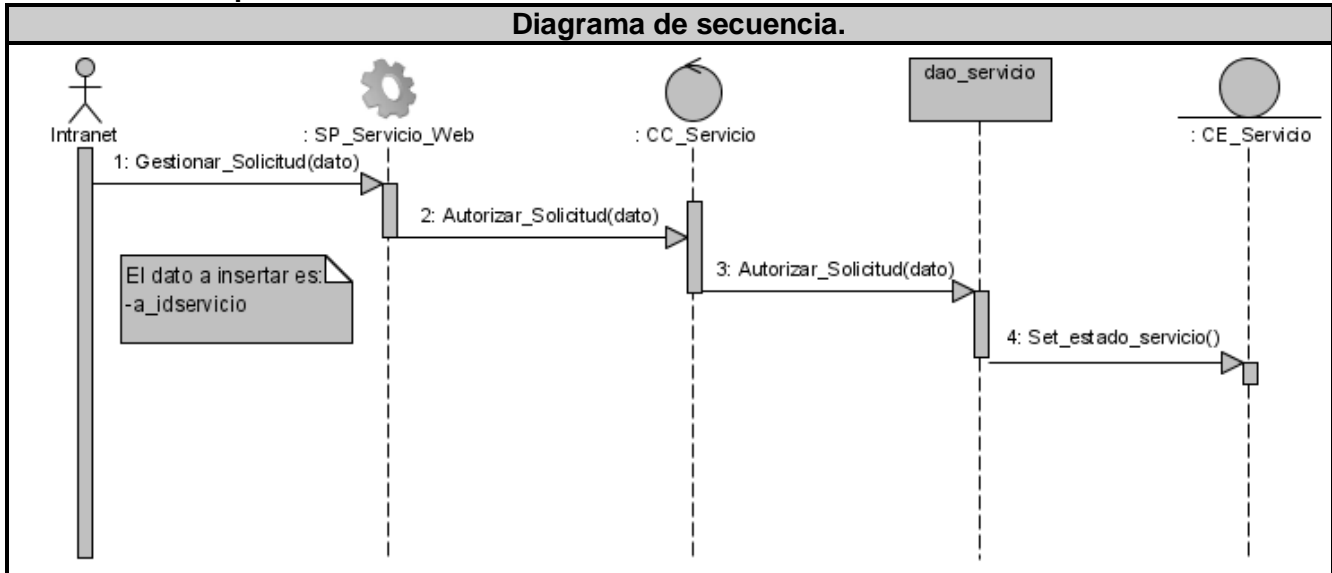


Fig.3-9.DS. CU_Gestionar_Solicitud

**Diagrama de Secuencia del Diseño del Caso de uso: Gestionar Solicitud
Escenario "Denegar Solicitud".**

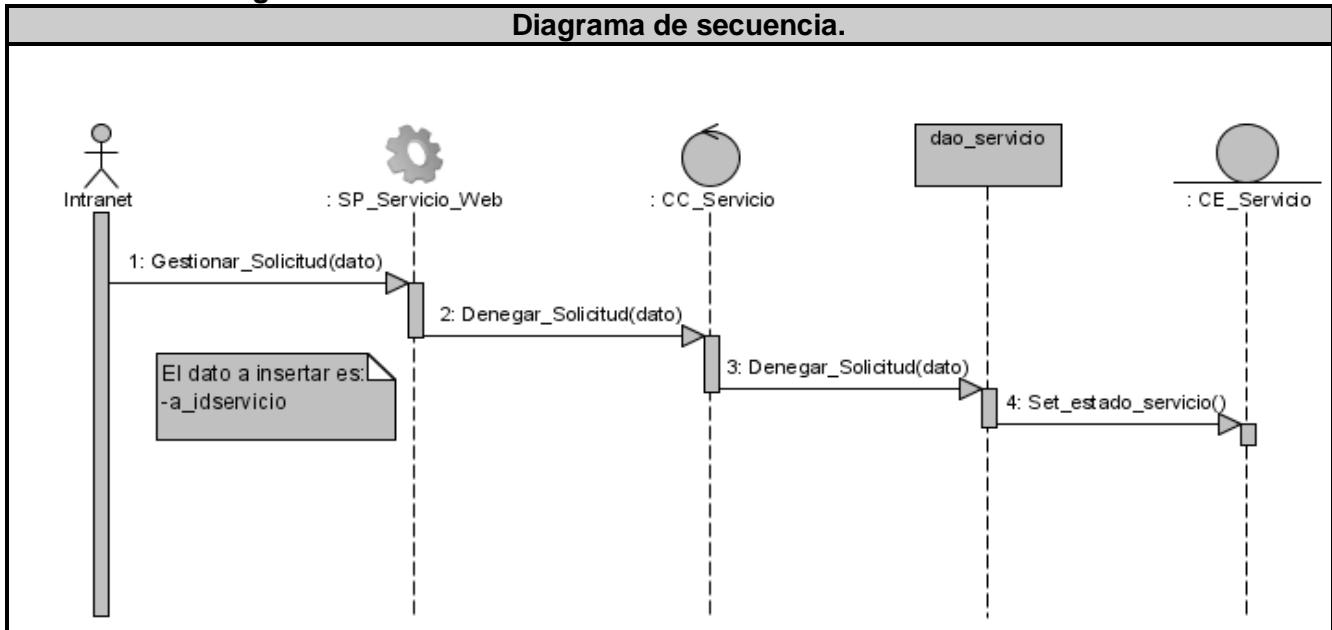


Fig.3-10.DS. CU_Gestionar_Solicitud

Diagrama de Secuencia del Diseño del Caso de uso: Mostrar Reporte.

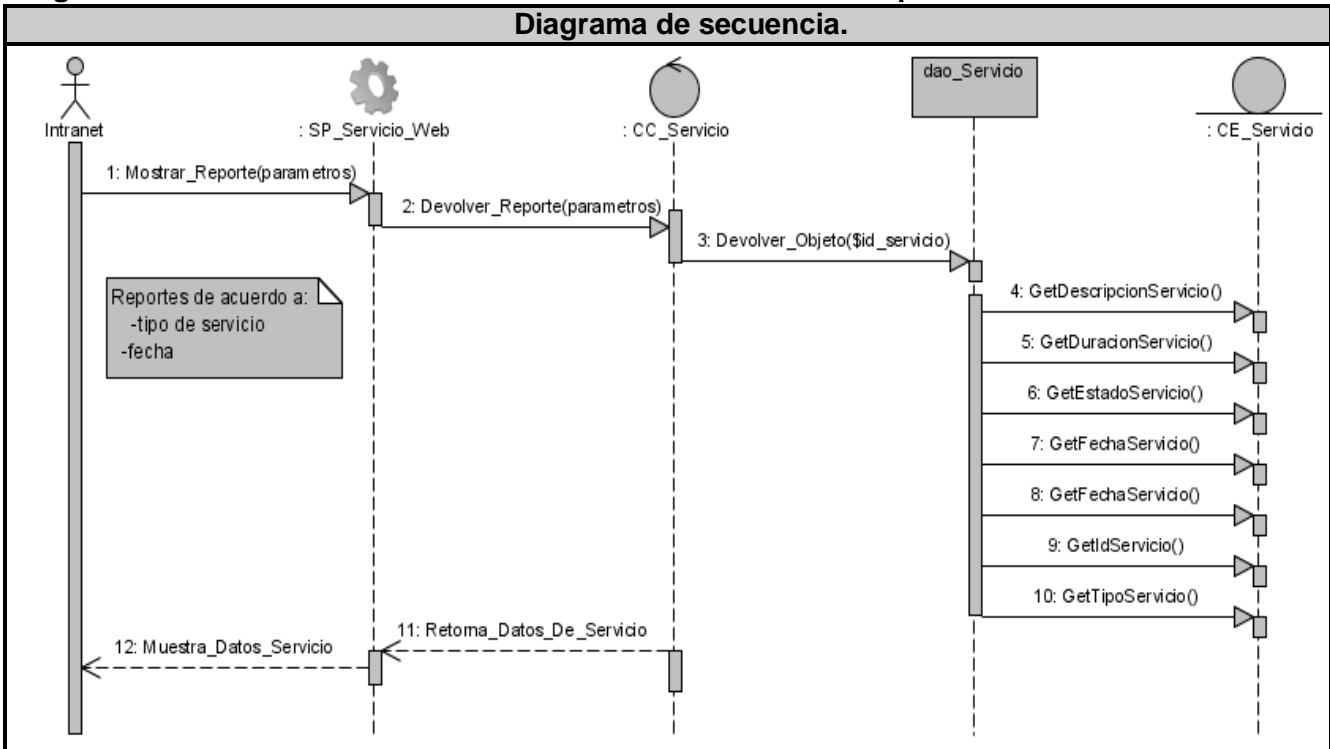


Fig.3-11.DS. CU_Mostrar_Reporte.

Diagrama de Secuencia del Diseño del Caso de uso: Ejecutar Servicio.

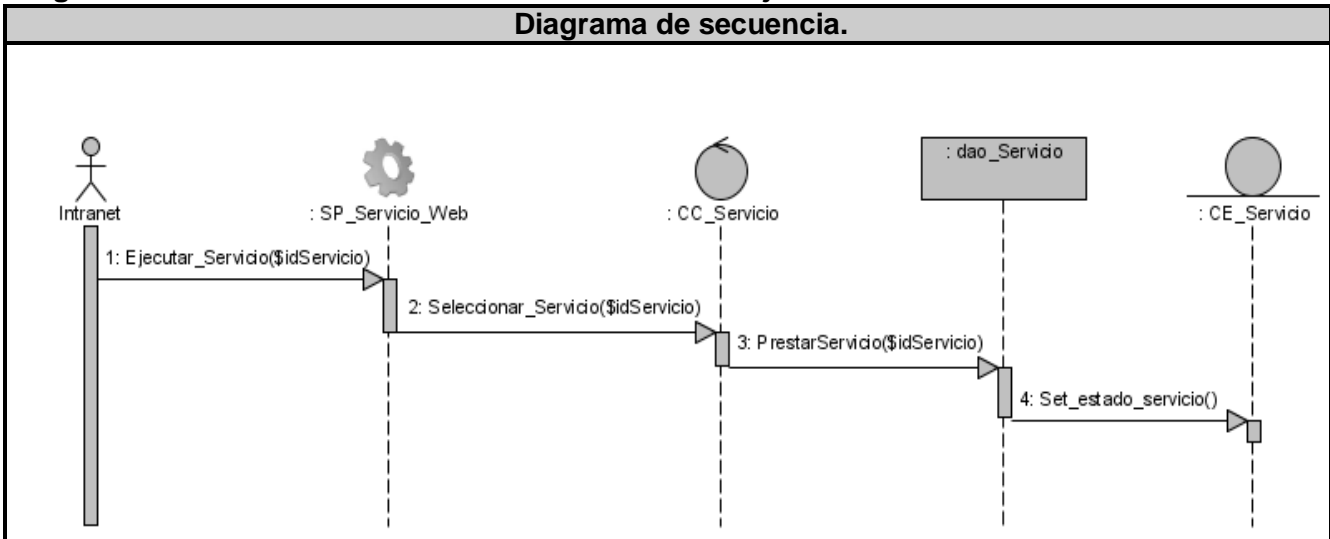


Fig.3-12.DS. CU_Ejecutar_Servicio.

Diagrama de Secuencia del Diseño del Caso de uso: Actualizar Listado.

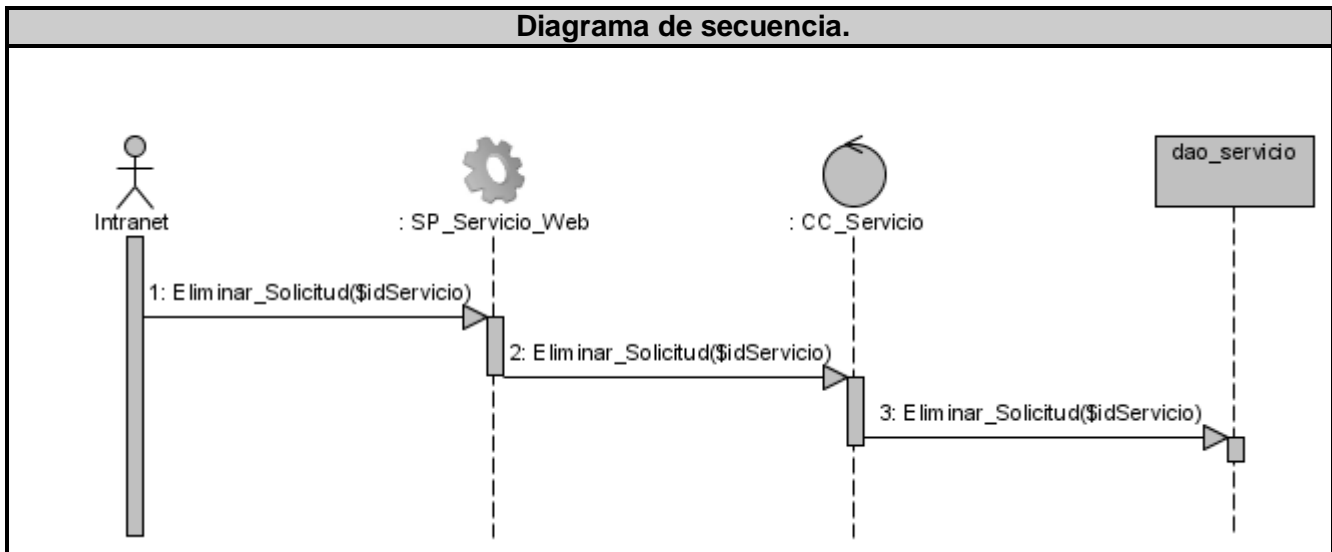


Fig.3-13.DS. CU_Actualizar_Listado.

3.1.3 Descripción de las clases del diseño.

Clase del Servicio Web.

Nombre: Web Service	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DevolverTipoServicioPorId(\$id)
Descripción:	Este método devuelve el tipo de servicio dado el identificador.
Nombre:	DevolverNombreTipoServicio(\$tipo)
Descripción:	Este método devuelve el nombre del servicio dado el tipo.
Nombre:	DevolverTipoServicios()
Descripción:	Este método devuelve el tipo de servicio en general.
Nombre:	DevolverTipoLineaSala()
Descripción:	Este método devuelve si el servicio es en línea o en sala.
Nombre:	DevolverTipoRecursos()
Descripción:	Este método devuelve el tipo de recursos de los préstamos.
Nombre:	DevolverTipoHorarios()
Descripción:	Este método devuelve el horario de los préstamos.
Nombre:	DevolverTipoWifi()
Descripción:	Este método devuelve el tipo de préstamo de red inalámbrica.
Nombre:	DevolverTipoCurso()
Descripción:	Este método devuelve el tipo de curso que se imparten en la entidad.
Nombre:	DevolverTipoFormato()
Descripción:	Este método devuelve el tipo de formato para la digitalización de programas.

Nombre:	DevolverTipoFotocopiado()
Descripción:	Este método devuelve el tipo de fotocopiado.
Nombre:	DevolverTipoBoletin()
Descripción:	Este método devuelve el tipo de boletín.
Nombre:	DevolverTipoDocumentos()
Descripción:	Este método devuelve el tipo de documentos para préstamos.
Nombre:	DevolverTipoObservatorio(\$pos)
Descripción:	Este método devuelve el tipo de observatorio que solicite el usuario.
Nombre:	DevolverTipoLugar()
Descripción:	Este método devuelve el lugar en que va a utilizar el servicio.
Nombre:	InsertarTipoObservatorio(\$obj)
Descripción:	Este método inserta el tipo de observatorio.
Nombre:	InsertarTipoBoletin(\$obj)
Descripción:	Este método inserta el tipo de boletín.
Nombre:	InsertarSolicitud(\$solicitud)
Descripción:	Este método inserta una solicitud.
Nombre:	DevolverCI(\$id)
Descripción:	Este método devuelve el carnet de identidad del usuario dado el servicio solicitado.
Nombre:	DevolverReporte(\$este)
Descripción:	Este método el reporte solicitado.
Nombre:	InsertarServicioAsesor(\$objAsesor)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioConrecurso(\$objConRecurso)
Descripción:	Este método inserta un servicio determinado.
Nombre:	DevolverServiciosEnEspera()
Descripción:	Este método devuelve los servicios que no han sido atendidos.
Nombre:	AutorizarServicio(\$id)
Descripción:	Este método autoriza el servicio dado el identificador.
Nombre:	DenegarServicio(\$id)
Descripción:	Este método deniega el servicio dado el identificador.
Nombre:	EliminarServicio(\$id)
Descripción:	Este método elimina un servicio dado el identificador.
Nombre:	PrestarServicio(\$id)
Descripción:	Este método presta un servicio dado el identificador.
Nombre:	DevolverServiciosUsuario(\$ci)
Descripción:	Este método devuelve los servicios solicitados por un determinado usuario.
Nombre:	InsertarServicioSalonTv(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioEBoletin(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioReferencia(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioCanalTv(\$obj)

Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioPresDoc(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioPcPortatil(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioObservatorio(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioMonitoreo(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioImpresion(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioHemeroteca(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioGrabacionProg(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioFotocopiado(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioDigitalProg(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioDigitaldoc(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioCurso(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioDiseno(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicio(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioWifi(\$objWifi)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioProgEnVivo(\$obj)
Descripción:	Este método inserta un servicio determinado.
Nombre:	InsertarServicioTransFormato(\$obj)
Descripción:	Este método inserta un servicio determinado.

Clases de las reglas de negocio.

Nombre: CC_Asesor	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioAsesor(\$servicioAsesor)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_Boletin	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioBoletin(\$servicioBoletin)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_CanalTv	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioCanalTv(\$servicioCanalTv)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_ConRecurso	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioConRecurso(\$servicioConRecurso)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_Curso	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioCurso(\$servicioCurso)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_DigitalDoc	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioDigitalDoc(\$servicioDigitalDoc)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_Disenio

Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioDiseno(\$servicioDiseno)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_Fotocopiado	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioFotocopiado(\$servicioFotocopiado)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_GrabacionProg	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioGrabacionProg(\$servicioGrabacionProg)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_Servicio	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicio(\$servicio)
Descripción:	Este método inserta los datos de un servicio.
Nombre:	CCAutorizarServicio(\$a_idservicio)
Descripción:	Este método autoriza la solicitud del servicio seleccionado.
Nombre:	CCDenegarServicio(\$a_idServicio)
Descripción:	Este método deniega la solicitud del servicio seleccionado.
Nombre:	CCEliminarServicio(\$a_idServicio)
Descripción:	Este método elimina el servicio seleccionado.
Nombre:	ReglasNegocioInsertarServicio(\$servicio)
Descripción:	Este método verifica la validez de los datos.
Nombre:	ReglasNegocioAutorizarServicio(\$a_idservicio)
Descripción:	Este método verifica la validez de los datos.
Nombre:	ReglasNegocioDenegarServicio(\$a_idservicio)
Descripción:	Este método verifica la validez de los datos.
Nombre:	ReglasNegocioEliminarServicio(\$a_idservicio)

Descripción:	Este método verifica la validez de los datos.
--------------	---

Nombre: CC_ Hemeroteca	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioHemeroteca(\$servicioHemeroteca)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_ Impresion	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioImpresion(\$servicioImpresion)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_ Monitoreo	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioMonitoreo(\$servicioMonitoreo)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_ Observatorio	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioObservatorio(\$servicioObservatorio)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_ PcPortatil	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioPcPortatil(\$servicioPcPortatil)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_PresDoc	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioPresDoc(\$servicioPresDoc)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_ProgEnVivo	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioProgEnVivo(\$servicioProgEnVivo)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_Referencia	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioReferencia(\$servicioReferencia)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_SalonTv	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioSalonTv(\$servicioSalonTv)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_ServProg	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioServProg(\$servicioServProg)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_transFormato	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioTransFormato(\$servicioTransFormato)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.
Nombre: CC_WiFi	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarServicioWiFi(\$servicioWiFi)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_SolicitudCliente	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarSolicitudCliente(\$servicioAsesor)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoBoletin	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoBoletin(\$nomTipoBoletin)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoCurso	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoCurso(\$nomTipoCurso, \$plazas)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoDiseno	
Clase Controladora	
Atributo:	Tipo

Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoDiseno(\$nomTipoDiseno)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoDocumento	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoDocumento(\$nomTipoDocumento)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoFormato	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoFormato(\$nomTipoFormato)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoFotocopiado	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoFotocopiado(\$nomTipoFotocopiado)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoHorario	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoHorario(\$nomTipoHorario)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoLineaSala	
Clase Controladora	
Atributo:	Tipo

Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoLineaSala(\$nomTipoLineaSala)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoLugar	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoLugar(\$nomTipoLugar)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoObservatorio	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoObservatorio(\$nomTipoObservatorio)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoRecurso	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoRecurso(\$nomTipoRecurso)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Nombre: CC_TipoServicio	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	CCInsertarNomencladorTipoServicio(\$nomTipoServicio)
Descripción:	Este método inserta los datos del servicio en la tabla correspondiente.

Clases entidades del negocio.

Nombre: CE_Servicios	
Clase Entidad	
Atributo:	Tipo
idServicio	var

estadoServicio	var
descripcionServicio	var
duracionServicio	var
fechaServicio	var
nom_idTipoServicio	var
Para cada responsabilidad:	
Nombre:	Servicio(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetIdServicio()
Descripción:	Este método devuelve el identificador del servicio.
Nombre:	GetEstadoServicio()
Descripción:	Este método devuelve el estado en que se encuentra el servicio.
Nombre:	GetDescripcionServicio()
Descripción:	Este método devuelve una descripción breve del servicio.
Nombre:	GetDuracionServicio()
Descripción:	Este método devuelve el tiempo de duración del servicio
Nombre:	GetFechaServicio()
Descripción:	Este método devuelve la fecha del servicio.
Nombre:	GetTipoServicio()
Descripción:	Este método devuelve el tipo de servicio.

Nombre: CE_RegistroServicios	
Clase Entidad	
Atributo:	Tipo
idServicioRegistrado	var
nombreCliente	var
fechaServicio	var
Para cada responsabilidad:	
Nombre:	RegistroServicios(\$a_idServicioRegistrado, \$a_nombreCliente, \$a_fechaServicio)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetIdServicioRegistrado()
Descripción:	Este método devuelve el identificador del servicio registrado.
Nombre:	GetNombreCliente()
Descripción:	Este método devuelve el nombre del cliente que solicito el servicio.
Nombre:	GetFechaServicio()
Descripción:	Este método devuelve la fecha de ejecución del servicio.

Nombre: CE_SolicitudCliente
Clase Entidad

Atributo:	Tipo
ci	Var
idServicio	Var
Para cada responsabilidad:	
Nombre:	SolicitudCliente(\$a_ci, \$a_idServicio, \$a_idTipoCliente)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetCI()
Descripción:	Este método devuelve el Carne de identidad del cliente.
Nombre:	GetIdServicio()
Descripción:	Este método devuelve el identificador del servicio.

Nombre: CE_Asesor	
Clase Entidad	
Atributo:	Tipo
nom_idTipoLineaSala	var
Para cada responsabilidad:	
Nombre:	Asesor(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_nom_idTipoLineaSala)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetIdTipoAsesoria()
Descripción:	Este método devuelve el tipo de asesoría que solicite el cliente.

Nombre: CE_Boletin	
Clase Entidad	
Atributo:	Tipo
tematica	Var
email	Var
nom_idTipoBoletin	var
Para cada responsabilidad:	
Nombre:	Boletin(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_tematica, \$a_email, \$a_nom_idTipoBoletin)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetTematica()
Descripción:	Este método devuelve la tematica que abordará el boletín.
Nombre:	GetEmail()
Descripción:	Este método devuelve la dirección de correo electrónico del cliente.
Nombre:	GetTipoBoletin ()
Descripción:	Este método devuelve el tipo de boletín que desea solicitar el cliente.

Nombre: CE_CanalTv	
Clase Entidad	

Atributo:		Tipo
numeroCanal		Var
nombreCanal		var
Para cada responsabilidad:		
Nombre:	CanalTv(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_numeroCanal, \$a_nombreCanal)	
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.	
Nombre:	GetNumeroCanal()	
Descripción:	Este método devuelve el número del canal de TV que desea ver el cliente.	
Nombre:	GetNombreCanal()	
Descripción:	Este método devuelve el nombre del canal de TV que desea el cliente.	

Nombre: CE_ConRecurso		
Clase Entidad		
Atributo:		Tipo
nom_idTipoRecurso		var
Para cada responsabilidad:		
Nombre:	ConRecurso(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$nom_idTipoRecurso)	
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.	
Nombre:	GetTipoRecurso()	
Descripción:	Este método devuelve el tipo de recurso que desea consultar el cliente.	

Nombre: CE_Curso		
Clase Entidad		
Atributo:		Tipo
nom_idTipoCurso		var
Para cada responsabilidad:		
Nombre:	Curso(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_nom_idTipoCurso)	
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.	
Nombre:	GetTipoCurso()	
Descripción:	Este método devuelve el tipo de curso.	

Nombre: CE_DigitalDoc		
Clase Entidad		
Atributo:		Tipo
fechaTope		var
Para cada responsabilidad:		
Nombre:	DigitalDoc(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_fechaTope)	
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.	

Nombre:	GetFechaTope()
Descripción:	Este método devuelve la fecha en que debe terminar el servicio.

Nombre: CE_Disenio	
Clase Entidad	
Atributo:	Tipo
tituloCurso	var
fechaTope	Var
nom_idTipoDiseno	var
Para cada responsabilidad:	
Nombre:	Diseno(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_tituloCurso, \$a_fechaTope, \$a_nom_idTipoDiseno)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetTituloCurso()
Descripción:	Este método devuelve el titulo del curso que desea el diseño.
Nombre:	GetFechaTope()
Descripción:	Este método devuelve la fecha límite para el servicio.
Nombre:	GetTipoDiseno()
Descripción:	Este método devuelve el tipo de diseño que desea el cliente.

Nombre: CE_Fotocopiado	
Clase Entidad	
Atributo:	Tipo
nom_idTipoFotocopiado	var
Para cada responsabilidad:	
Nombre:	Fotocopiado(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_nom_idTipoFotocopiado)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetTipoFotocopiado()
Descripción:	Este método devuelve el tipo de fotocopiado que solicite el cliente.

Nombre: CE_GrabacionProg	
Clase Entidad	
Atributo:	Tipo
numeroCanal	Var
nombreCanal	Var
programa	var
nom_idTipoFormato	var
Para cada responsabilidad:	
Nombre:	GrabacionProg(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_numeroCanal, \$a_nombreCanal, \$a_programa, \$a_nom_idTipoFormato)

Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetNumeroCanal()
Descripción:	Este método devuelve el numero del canal de TV.
Nombre:	GetNombreCanal()
Descripción:	Este método devuelve el nombre del canal de TV.
Nombre:	GetPrograma()
Descripción:	Este método devuelve el nombre del programa de TV.
Nombre:	GetTipoFormato()
Descripción:	Este método devuelve el formato en que desea el programa.

Nombre: CE_Hemeroteca	
Clase Entidad	
Atributo:	Tipo
autor	Var
obra	Var
nom_idTipoLineaSala	var
Para cada responsabilidad:	
Nombre:	Hemeroteca(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_iautor, \$a_obra, \$a_nom_idTipoLineaSala)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetAutor()
Descripción:	Este método devuelve el autor de la obra solicitada.
Nombre:	GetObra()
Descripción:	Este método devuelve el nombre de la obra solicitada.
Nombre:	GetTipo()
Descripción:	Este método devuelve el tipo de obra.

Nombre: CE_Impresion	
Clase Entidad	
Atributo:	Tipo
cantPag	var
Para cada responsabilidad:	
Nombre:	Impresion(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_cantPag)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetCantPag()
Descripción:	Este método devuelve la cantidad de páginas necesarias para el servicio.

Nombre: CE_Monitoreo	
Clase Entidad	
Atributo:	Tipo
tematica	var

Para cada responsabilidad:	
Nombre:	Monitoreo(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_tematica)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetTematica()
Descripción:	Este método devuelve la tematica que abordará el monitoreo.

Nombre: CE_Observatorio	
Clase Entidad	
Atributo:	Tipo
nom_idTipoObservatorio	var
Para cada responsabilidad:	
Nombre:	Observatorio(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_nom_idTipoObservatorio)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetTipoObservatorio()
Descripción:	Este método devuelve el tipo de observatorio que solicite el cliente.

Nombre: CE_PcPortatil	
Clase Entidad	
Atributo:	Tipo
nom_idTipoLugar	var
Para cada responsabilidad:	
Nombre:	PcPortatil(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_nom_idTipoLugar)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetLugar()
Descripción:	Este método devuelve el lugar donde el cliente hará uso del servicio.

Nombre: CE_PresDoc	
Clase Entidad	
Atributo:	Tipo
nom_idTipoDocumento	var
Para cada responsabilidad:	
Nombre:	PresDoc(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_nom_idTipoDocumento)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetDocumento()
Descripción:	Este método devuelve el tipo de documento del servicio.

Nombre: CE_ProgEnVivo	
Clase Entidad	

Atributo:		Tipo
nombrePrograma		Var
numeroCanal		Var
nombreCanal		var
Para cada responsabilidad:		
Nombre:	ProgEnVivo(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_nombrePrograma, \$a_numeroCanal, \$a_nombreCanal)	
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.	
Nombre:	GetNombrePrograma()	
Descripción:	Este método devuelve el nombre del programa de se solicite.	
Nombre:	GetNumeroCanal()	
Descripción:	Este método devuelve el numero del canal de TV.	
Nombre:	GetNombreCanal()	
Descripción:	Este método devuelve el nombre del canal de TV.	

Nombre: CE_Referencia		
Clase Entidad		
Atributo:		Tipo
nom_idTipoLineaSala		var
Para cada responsabilidad:		
Nombre:	Referencia(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_idTipoLineaSala)	
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.	
Nombre:	GetTipoReferencia()	
Descripción:	Este método devuelve el tipo de referencia que se solicite.	

Nombre: CE_SalonTv		
Clase Entidad		
Atributo:		Tipo
programa		Var
duracionPrograma		Var
canales		var
Para cada responsabilidad:		
Nombre:	SalonTv(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_programa, \$a_duracionPrograma, \$a_canales)	
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.	
Nombre:	GetPrograma()	
Descripción:	Este método devuelve el nombre del programa que se solicite.	
Nombre:	GetDuracionPrograma()	
Descripción:	Este método devuelve la duración del programa.	
Nombre:	GetCanales()	

Descripción:	Este método devuelve el o los canales de TV que solicite el cliente.
--------------	--

Nombre: CE_ServProg	
Clase Entidad	
Atributo:	Tipo
numeroCanal	Var
nombreCanal	Var
programa	Var
nom_idTipoFormato	Var
nom_tipoFormato2	Var
nom_idTipoServPrograma	var
Para cada responsabilidad:	
Nombre:	ServProg(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_numeroCanal, \$a_nombreCanal, \$a_programa, \$a_nom_idTipoFormato, \$a_nom_tipoFormato2, \$a_nom_idTipoServPrograma)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetNumeroCanal()
Descripción:	Este método devuelve el numero del canal de TV.
Nombre:	GetNombreCanal()
Descripción:	Este método devuelve el nombre del canal de TV.
Nombre:	GetPrograma()
Descripción:	Este método devuelve el nombre del programa de TV.
Nombre:	GetTipoFormato()
Descripción:	Este método devuelve el formato que seleccione el cliente.
Nombre:	GetTipoFormato2()
Descripción:	Este método devuelve el formato que seleccione el cliente.
Nombre:	GetTipoServPrograma()
Descripción:	Este método devuelve el tipo de servicio de programación.

Nombre: CE_TransFormato	
Clase Entidad	
Atributo:	Tipo
nom_idTipoFormato	Var
nom_idTipoFormato2	var
Para cada responsabilidad:	
Nombre:	TransFormato(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_nom_idTipoFormato, \$a_nom_idTipoFormato2)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetTipoFormato()
Descripción:	Este método devuelve el tipo de formato que seleccione el cliente.
Nombre:	GetTipoFormato2()

Descripción:	Este método devuelve el tipo de formato que seleccione el cliente.
--------------	--

Nombre: CE_ Wifi	
Clase Entidad	
Atributo:	Tipo
tiempo	Var
horario	Var
nom_idTipoServiciosWifi	var
Para cada responsabilidad:	
Nombre:	Wifi(\$a_idServicio, \$a_descripcionServicio, \$a_duracionServicio, \$a_fechaServicio, \$a_nom_idTipoServicio, \$a_tiempo, \$a_horario, \$a_idTipoServiciosWifi)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	GetTiempo()
Descripción:	Este método devuelve el tiempo por el que se solicite el servicio.
Nombre:	GetHorario()
Descripción:	Este método devuelve el horario del servicio.
Nombre:	GetTipoServWifi()
Descripción:	Este método devuelve el tipo de servicio de red.

Clases de acceso a datos.

Nombre:DBConnection	
Clase Controladora.	
Atributo	Tipo
adodbObj	var
DSN	var
serverObj	var
Nombre:	DBConnection(var pDSN,)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.
Nombre:	execute(var query,)
Descripción:	Este método es el que accede a la BD y devuelve en un arreglo los resultados obtenidos.
Nombre:	execute_insert(var query,)
Descripción:	Este método devuelve el último identificador insertado en la BD dado una consulta.
Nombre:	insertSQL(var query,var array,)
Descripción:	Dado una consulta a una tabla de la bd, insertar una tupla con los datos contenidos en el arreglo.
Nombre:	setServerObject(var type,)
Descripción:	Establece a que gestor de base de datos se va a conectar y devuelve un objeto con la conexión del mismo.
Nombre:	updateSQL(var query,var array,)

Descripción:	Dado una consulta a una tabla de la bd, actualizará una tupla con los datos contenidos en el arreglo.
---------------------	---

Nombre:DSN	
Clase controladora	
Atributo	Tipo
db	var
host	var
pass	var
port	var
serverType	var
user	var
Nombre:	DSN(var sT,var h,var p,var u,var pa,var d,)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto.

Nombre:Factory	
Clase controladora	
Atributo	Tipo
conn	var
Nombre:	execute(var query,)
Descripción:	Ejecuta una sentencia query a la BD
Nombre:	execute_insert(var query,)
Descripción:	Ejecuta una consulta a la DB y devuelve el último identificador insertado en una tabla si este es auto incremental.
Nombre:	Factory()
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto, en este caso no se le pasa parámetros.
Nombre:	get_Affected_Rows(var query,)
Descripción:	Devuelve el número de filas que fueron afectadas dada una consulta a la BD
Nombre:	insertSQL(var query,var array,)
Descripción:	Dado una consulta a una tabla de la bd, insertar una tupla con los datos contenidos en el arreglo.
Nombre:	updateSQL(var query,var array,)
Descripción:	Dado una consulta a una tabla de la bd, actualizara una tupla con los datos contenidos en el arreglo.

Nombre:MySqlConnection	
Clase controladora	
Atributo	Tipo

conn	var
DSN	var
Nombre:	MySqlConnection(var pDSN,)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto
Nombre:	serverConnection()
Descripción:	Devuelve una conexión persistente a la BD con el gestor MySQL

Nombre: myException	
Clase controladora	
Atributo	Tipo
die	var
msgError	var
msgInfo	var
msgSist	var
Nombre:	__construct(var _msgSist,var _msgInfo,var _die,)
Descripción:	Este es el constructor encargado de inicializar los atributos del objeto
Nombre:	getArrayError()
Descripción:	Devuelve un arreglo con los errores producidos.
Nombre:	GetError()
Descripción:	Devuelve un error producido por una consulta a la bd
Nombre:	setArrayError(var msg,)
Descripción:	Cambiar el error dado un mensaje y guardarlo en un arreglo

Clases de acceso a datos de la implementación.

Nombre: DAO_ServicioAsesor	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioAsesor()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioAsesor(\$idAsesor, \$tipoAsesor)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioBoletin	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioBoletin()
Descripción:	Este es el constructor encargado de inicializar el objeto

Nombre:	insertarServicioBoletin(\$idBoletin, \$tematica, \$email, \$tipoBoletin)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioCanalTv	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioCanalTv()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioCanalTv(\$idCanalTv, \$numeroCanal, \$nombreCanal)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioConRecurso	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioConRecurso()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioConRecurso(\$idConRecurso, \$tipoConRecurso)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioCurso	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioCurso()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioCurso(\$idCurso, \$tipoCurso)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioDigitalDoc	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioDigitalDoc()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioDigitalDoc(\$idDigitalDoc, \$tipoDigitalDoc)

Descripción:	Este método inserta los datos del servicio en la base de datos.
--------------	---

Nombre: DAO_ServicioDiseno	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioDiseno()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioDiseno(\$idDiseno, \$tituloCurso, \$fechaTope, \$tipoDiseno)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioFotocopiado	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioFotocopiado()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioFotocopiado(\$idFotocopiado, \$tipoFotocopiado)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioGrabacionProg	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioGrabacionProg()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioGrabacionProg(\$idGrabacionProg, \$numeroCanal, \$nombreCanal, \$programa, \$tipoformato)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioHemeroteca	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioHemeroteca()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioHemeroteca(\$idHemeroteca, \$autor, \$obra, \$tipoHemeroteca)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioImpresion	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioImpresion()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioImpresion(\$idImpresion, \$cant)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioMonitoreo	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioMonitoreo()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioMonitoreo(\$idMonitoreo, \$tematica)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioObservatorio	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioObservatorio()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioObservatorio(\$idObservatorio, \$tipoObservatorio)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioPcPortatil	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioPcPortatil()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioPcPortatil(\$idPcPortatil, \$tipoLugar)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioPresDoc	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioPresDoc()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioPresDoc(\$idPresDoc, \$tipoPresDoc)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioProgEnVivo	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioProgEnVivo()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioProgEnVivo(\$idProgEnVivo, \$nombrePrograma, \$numeroCanal, \$nombreCanal)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioServProg	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioServProg()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioServProg(\$idServProg, \$numeroCanal, \$nombreCanal, \$programa, \$idTipoFormato, \$tipoFormato2, \$idTipoServPrograma)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioReferencia	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioReferencia()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioReferencia(\$idReferencia, \$tipoReferencia)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_ServicioSalonTv	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioSalonTv()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioSalonTv(\$idSalonTv, \$programa, \$duracionPrograma, \$canales)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_Servicio	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_Servicio()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicio(\$a_servicio)
Descripción:	Este método inserta los datos del servicio en la base de datos.
Nombre:	autorizarServicio(\$a_idServicio)
Descripción:	Este método autoriza la solicitud del servicio.
Nombre:	denegarServicio(\$a_idServicio)
Descripción:	Este método deniega la solicitud del servicio.
Nombre:	eliminarServicio(\$a_idServicio)
Descripción:	Este método elimina el servicio seleccionado.
Nombre:	Cambia_Estado(\$id)
Descripción:	Este método cambia el estado del servicio.

Nombre: DAO_ServicioTransFormato	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioTransFormato()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioTransFormato(\$idTransFormato, \$tipoTransFormato, \$tipoTransFormato2)
Descripción:	Este método inserta un servicio en la base de datos.

Nombre: DAO_ServicioWiFi	
Clase Controladora	

Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	DAO_ServicioWiFi()
Descripción:	Este es el constructor encargado de inicializar el objeto
Nombre:	insertarServicioWiFi(\$idWiFi, \$tiempo, \$horario, \$tipo)
Descripción:	Este método inserta un servicio en la base de datos.

Nombre: DAO_TipoBoletin	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoBoletin(\$nombreTipoBoletin)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoCurso	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoCurso(\$nombreTipoCurso, \$plazas)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoDiseno	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoDiseno(\$nombreTipoDiseno)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoDocumento	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoDocumento(\$nombreTipoDocumento)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoFormato

Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoFormato(\$idTipoFormato, \$nombreTipoFormato)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoFotocopiado	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoFotocopiado(\$idTipoFotocopiado, \$nombreTipoFotocopiado)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoHorario	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoHorario(\$idTipoHorario, \$nombreTipoHorario)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoLineaSala	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoLineaSala(\$idTipoLineaSala, \$nombreTipoLineaSala)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoLugar	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoLugar(\$idTipoLugar, \$nombreTipoLugar)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoObservatorio	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoObservatorio(\$idTipoObservatorio, \$idpadre, \$nombreTipoObservatorio)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoRecurso	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoRecurso(\$idTipoRecurso, \$nombreTipoRecurso)
Descripción:	Este método inserta los datos del servicio en la base de datos.

Nombre: DAO_TipoServicio	
Clase Controladora	
Atributo:	Tipo
Para cada responsabilidad:	
Nombre:	insertarNomencladorTipoServicio(\$idTipoServicio, \$nombreTipoServicio)
Descripción:	Este método inserta los datos del servicio en la base de datos.

3.2 Diseño de la base de datos.

Con el objetivo de lograr la persistencia de los datos y que estos puedan ser consultados en cualquier momento por cualquier usuario se utilizó, en la concepción de este prototipo, una base de datos relacional. El uso de esta técnica de almacenamiento está muy difundido actualmente debido a las garantías que ofrece en cuanto a la durabilidad de los datos se refiere.

3.2.1 Representación UML del Diagrama de Entidad - Relación.

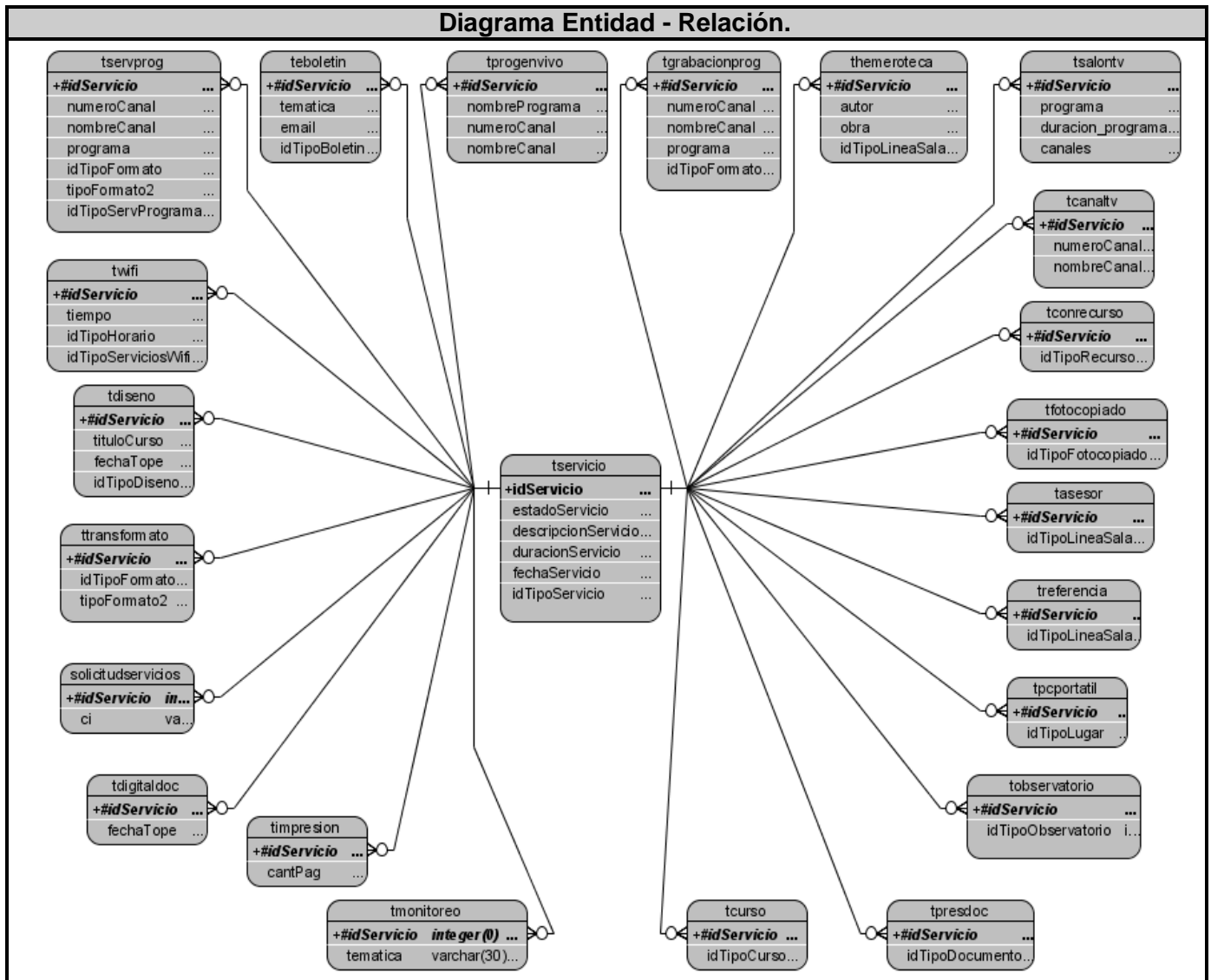


Fig.3.14. Diagrama Entidad-Relación.

3.2.2 Descripción de las tablas de la Base de Datos.

La descripción de las tablas que componen la base de datos se encuentra en los Anexos.

CAPÍTULO 4: IMPLEMENTACIÓN.

En el presente capítulo se muestra como se encuentra estructurado físicamente el sistema implementado mediante los modelos de despliegue y de componentes, según la propuesta realizada.

4.1 Diagrama de despliegue.

El modelo de despliegue contiene los nodos que conforman la topología de hardware sobre la cual se ejecuta el sistema software implementado (servicio web), además muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware. Las estaciones de trabajo, dispositivos y procesadores son reflejados como nodos.

4.2 Representación UML del Diagrama de Despliegue.

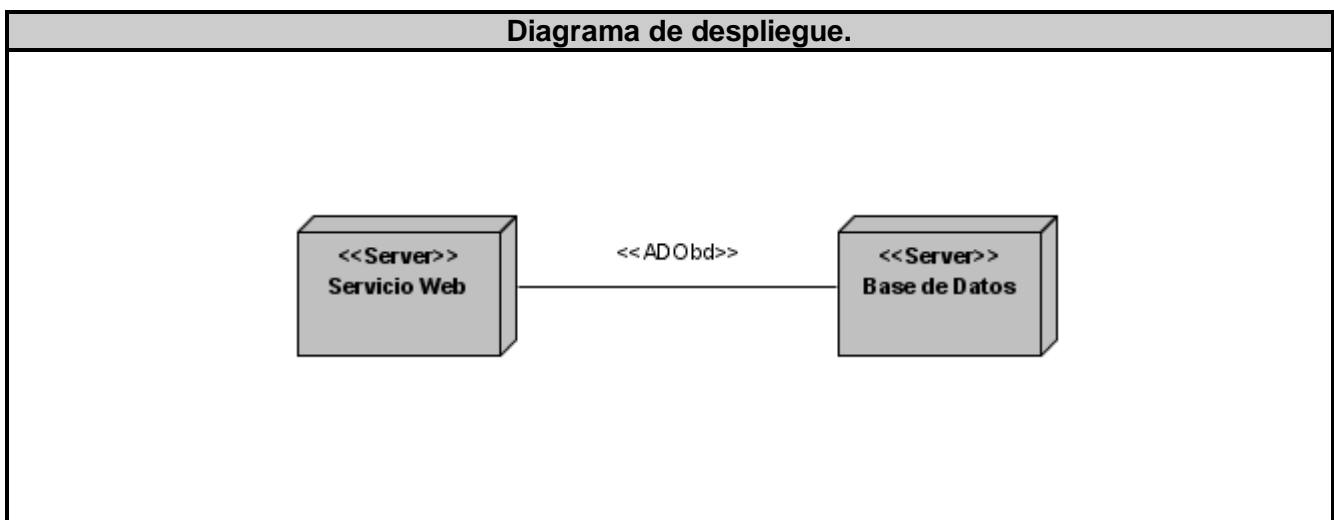


Fig.4 -1. Diagrama de despliegue.

4.3 Modelo Implementación

Entre los artefactos que se construyen durante la fase de implementación se encuentran los modelos de implementación, estos describen los componentes y la organización de acuerdo a los nodos; así como las dependencias de compilación entre ellos.

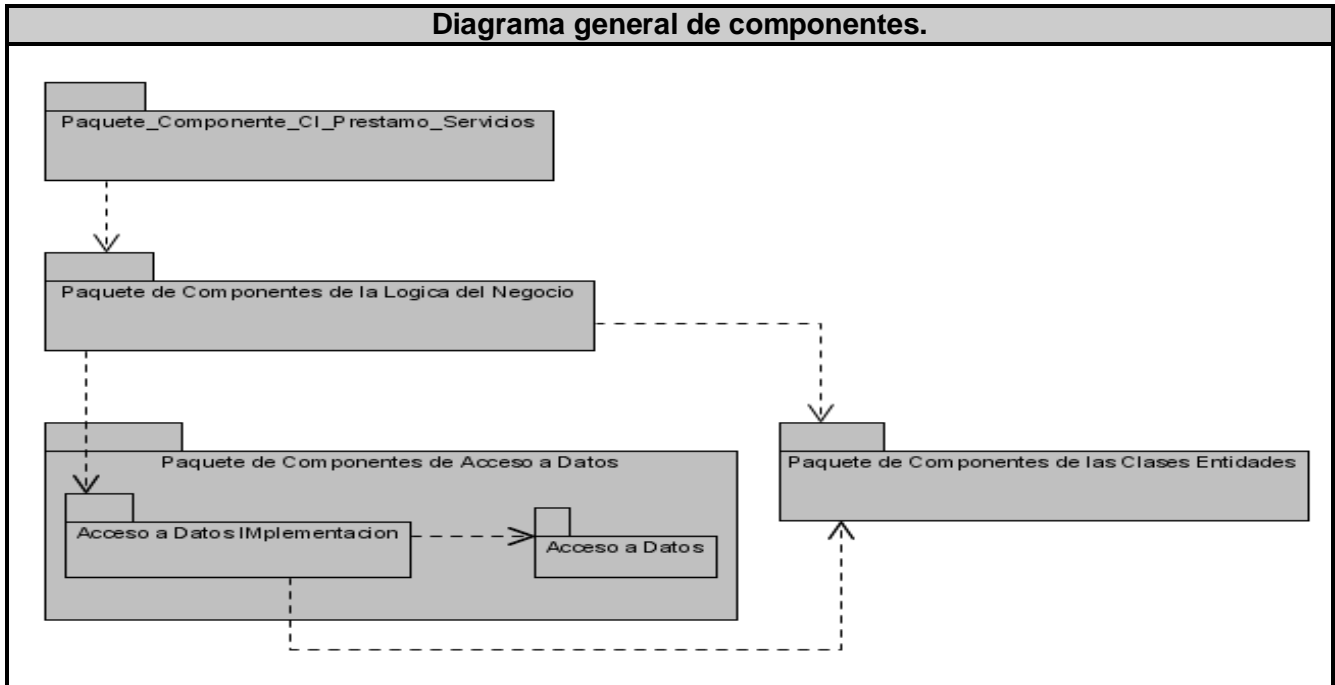


Fig.4-2. Diagrama general de componentes.

4.3.1 Diagrama de Componentes en Paquetes.

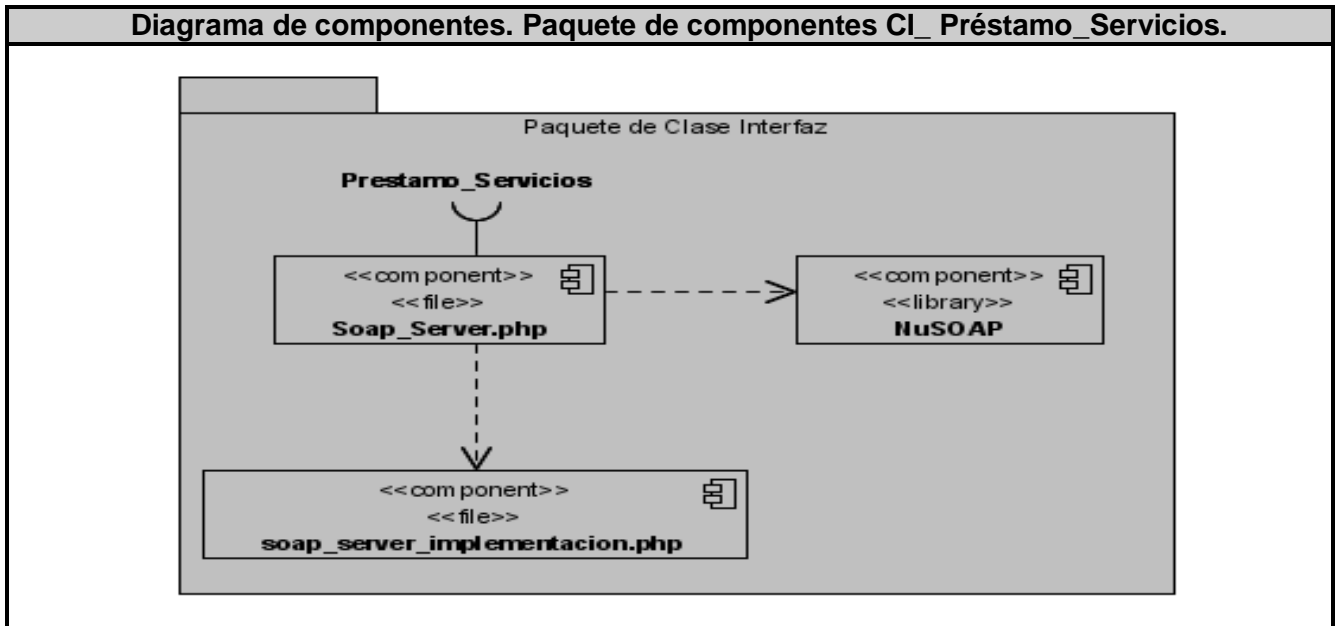


Fig.4-3. Diagrama componentes. Paquete de componentes CI_ Préstamo_Servicios.

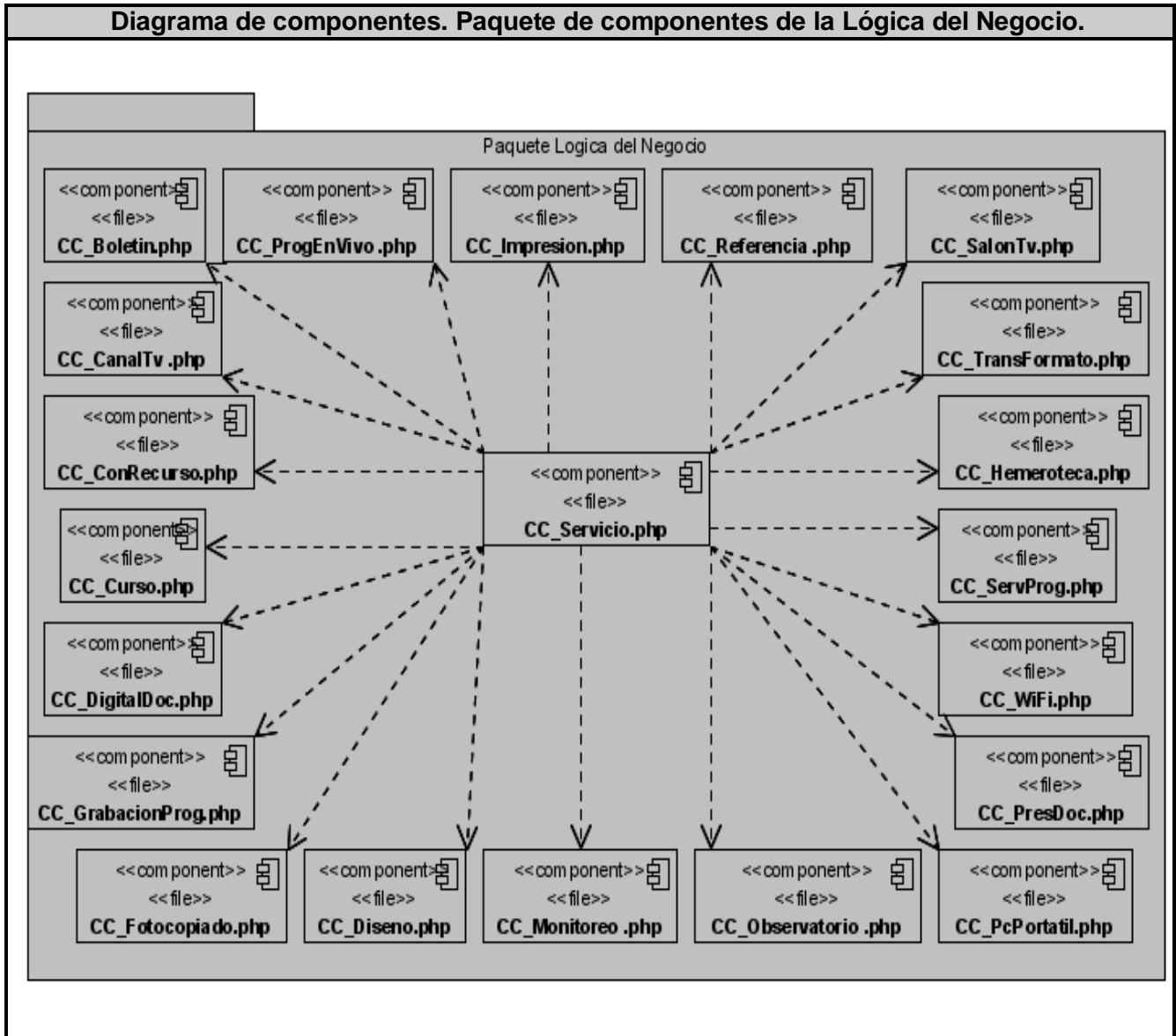


Fig.4-4. Diagrama componentes. Paquete de componentes de la Lógica del Negocio.

Diagrama de componentes. Paquete de componentes de Acceso a Datos de Implementación.

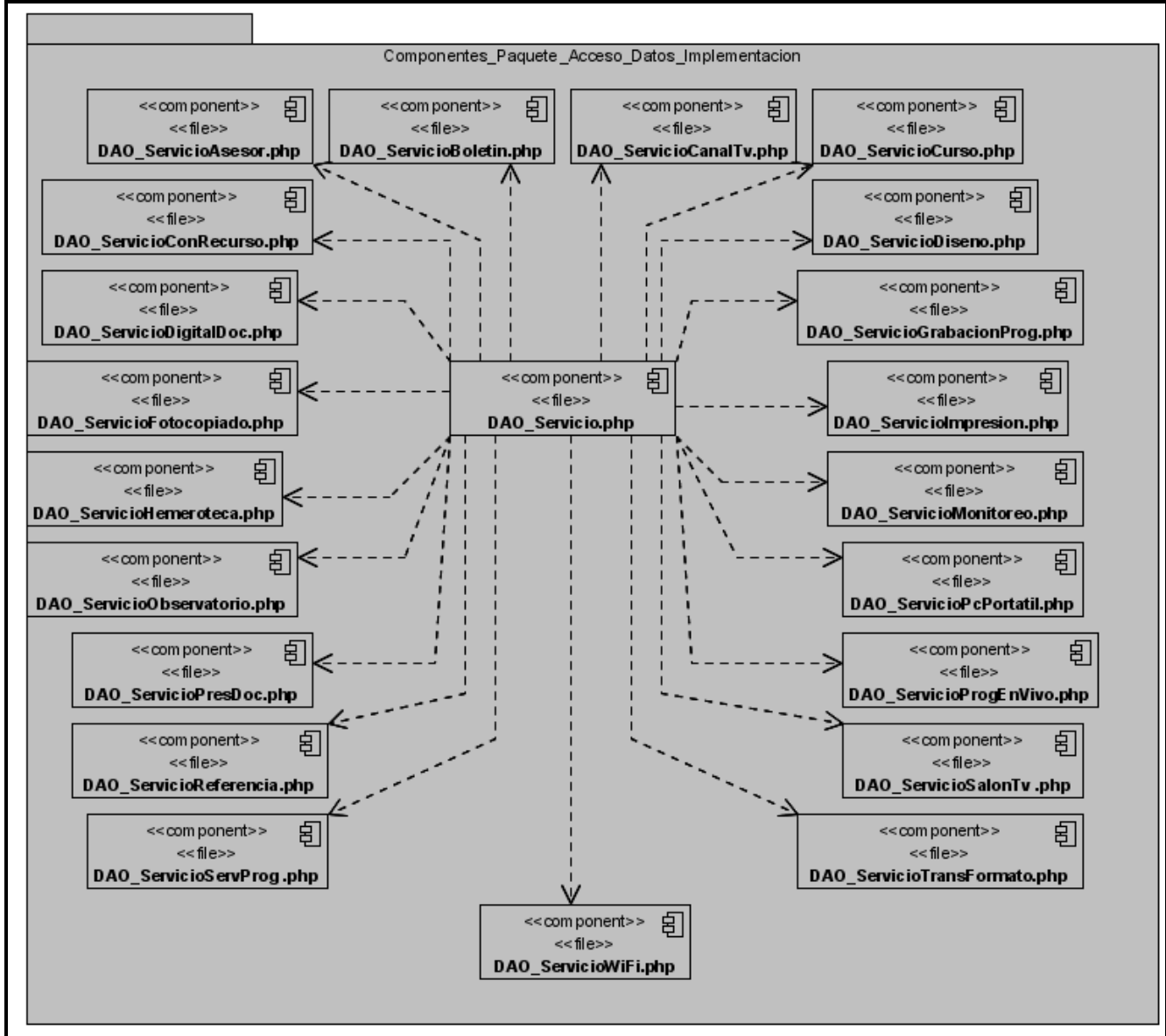


Fig.4-5. Diagrama componentes. Paquete de componentes de Acceso a Datos de Implementación.

Diagrama de componentes. Paquete de componentes de las Clases Entidades.

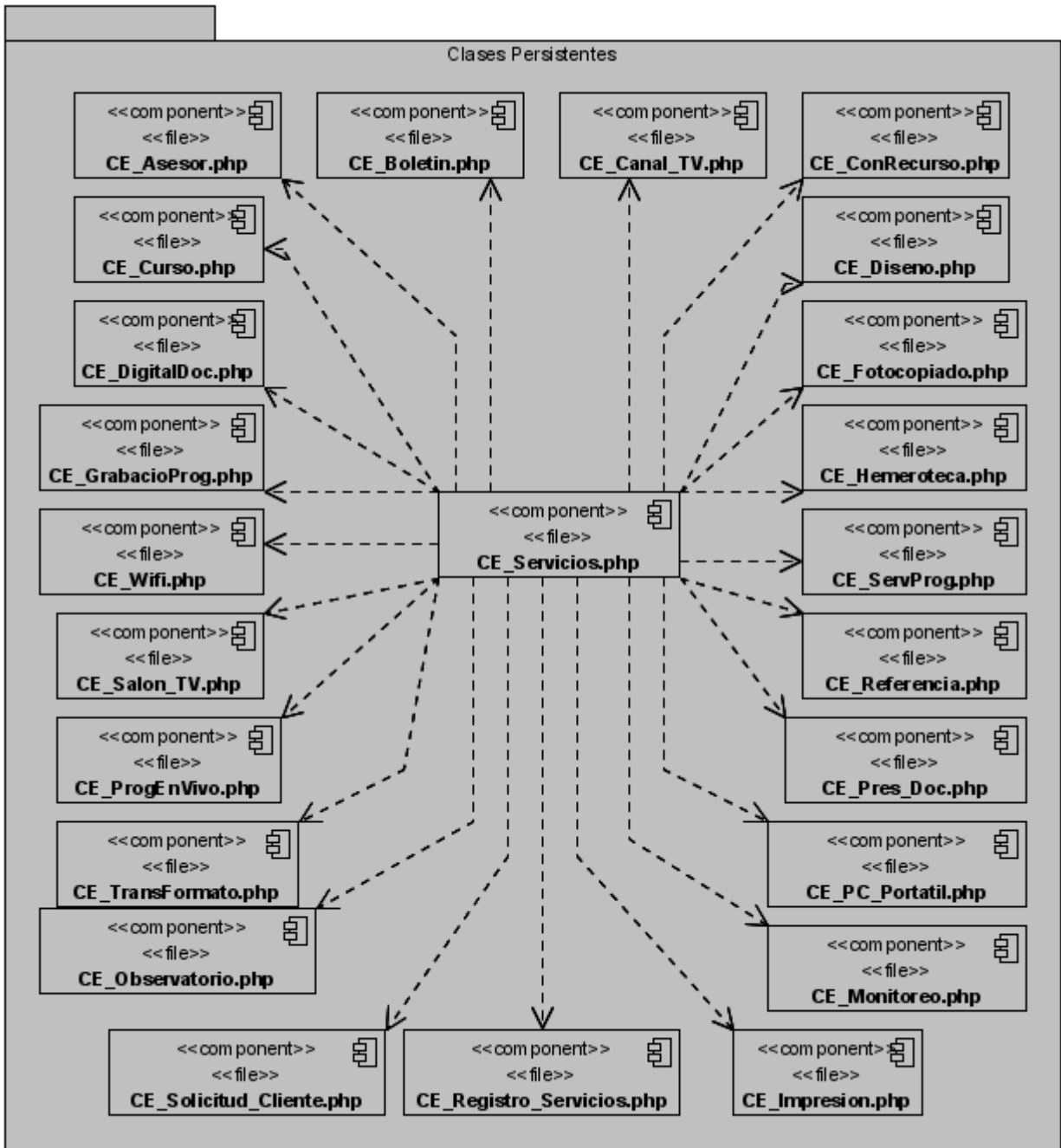


Fig.4-6. Diagrama componentes. Paquete de componentes de las Clases Entidades.

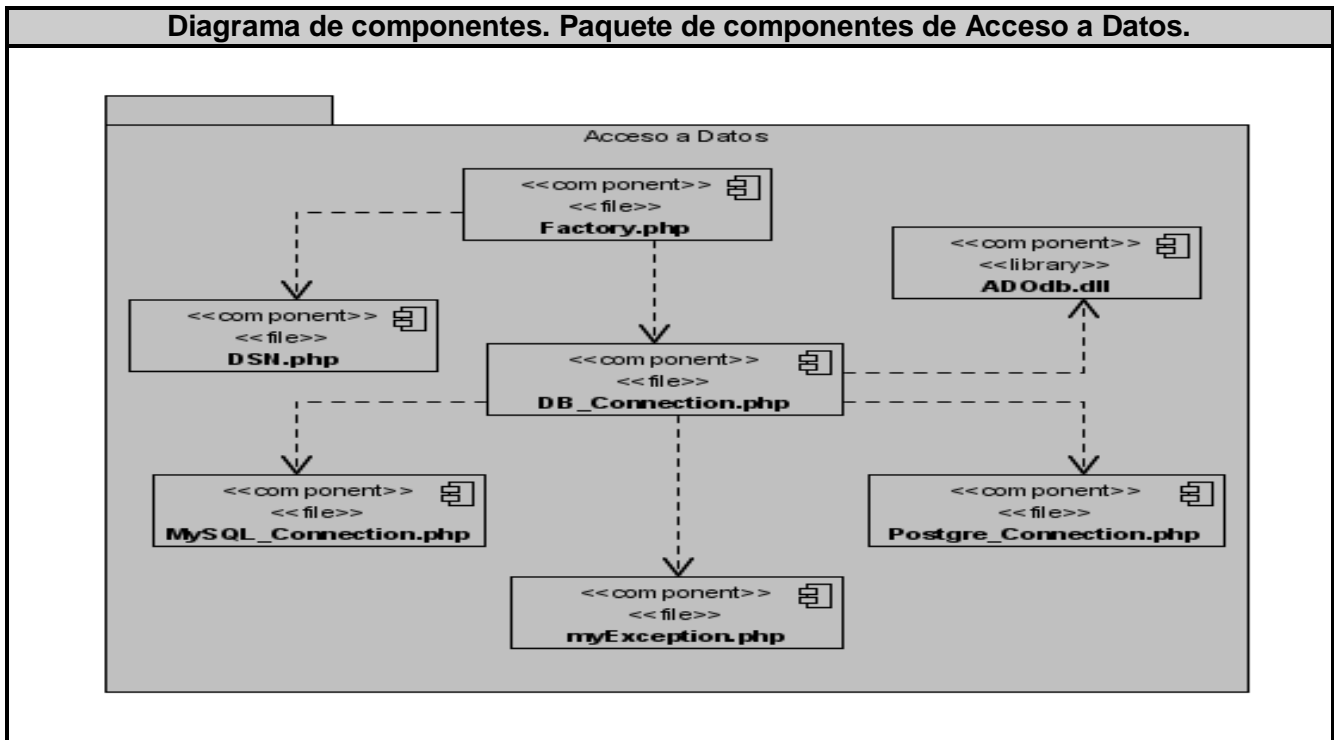


Fig.4-7. Diagrama componentes. Paquete de componentes de Acceso a Datos.

4.3.2 Descripción de cada uno de los Paquetes

Paquete_Componente_CI_Prestamo_Servicios: En este paquete se encuentran los componentes que contienen las implementaciones de las clases interfaz del Servicio Web, donde se exponen todas las funcionalidades que brinda el mismo, así como la librería NuSOAP que posibilitó su creación.

Paquete_Componente de la lógica del Negocio: En este paquete se encuentran los componentes que contienen las implementaciones de clases controladoras de entidades.

Paquete de Acceso a Datos Implementación: En este paquete se encuentran los componentes que contienen las implementaciones de clases que se ocupan de la persistencia en base de datos de las entidades.

Paquete de Componentes de clases Entidades: En este paquete se encuentran los componentes que contienen las implementaciones de las clases que representan las entidades del sistema.

Paquete de Acceso a Datos: En este paquete se encuentran los componentes que contienen las implementaciones de clases y librerías que permiten conectarnos a diferentes bases de datos.

CONCLUSIONES

Debido a la necesidad de mejoras en los sistemas de control de datos y a la pérdida de tiempo que significa llevar este control de forma manual, la implementación del sistema automatizado para controlar el préstamo de servicios en el Centro Rector del Claustro de Universidad para Todos, se traduce en una mejora significativa en la calidad de este servicio.

Para desarrollar el sistema necesario se estudió y modeló el dominio, identificando las actividades que eran objeto de automatización. Para comenzar la implementación se definieron los requisitos funcionales y no funcionales, además se realizó la modelación y descripción de los casos de uso del sistema los cuales dieron el punto de partida y apoyo para lograr el objetivo final: poner en funcionamiento el Servicio Web.

En la fase de diseño se llevó a cabo el modelado de los diagramas de clases del diseño y el de entidad relación, así como la descripción de las clases y tablas de la base de datos. Por otra parte se propuso la realización del diagrama de despliegue en el cual se expone la ubicación física de cada componente hardware y software, y el diagrama de componentes para describir los componentes físicos del sistema y sus relaciones.

El sistema se desarrolló siguiendo la metodología RUP, y se utilizaron representaciones UML para la modelación de todas las fases del proyecto. Luego de realizado todo este trabajo se puede concluir que el Servicio Web implementado brinda solución a la situación problemática planteada, la cual posibilitó el origen a esta investigación.

Por tanto, se concluye que los objetivos propuestos para el presente proyecto, han sido cumplidos satisfactoriamente. Se dan una serie de recomendaciones que deben tenerse en cuenta para el trabajo futuro.

RECOMENDACIONES

De manera general los objetivos trazados en la apertura de este trabajo han sido alcanzados, pero, a lo largo del proceso de desarrollo, ha quedado claro que la propuesta es sólo la primera etapa de un proyecto que puede ser mucho más ambicioso. Por tanto se hacen las siguientes recomendaciones:

- ✓ Implementar más funcionalidades al Servicio Web, adecuándolo a demandas de clientes interesados de otras empresas o instituciones, para que solamente no se ajuste a este centro para el cual se creó, sino que sirva para cualquier organismo del país.
- ✓ Someter a una prueba de calidad rigurosa al software para su posterior uso.

REFERENCIAS BIBLIOGRÁFICAS.

[1] Préstamo. R.A.E. Diccionario de la Lengua Española 2002.

[2] Gestión de servicios. Disponible en:

<http://www.loquo.com/ps/clases-cursos/informatica/itil-curso-gratuito/9870621>

[3]. Servicios Web con PHP (NuSOAP) [Consultado el: febrero de 2008]. Disponible en:

<http://www.nociondigital.com/webmasters/php-tutorial-servicios-web-con-php-nusoap-detalle-168.html>.

[4]. Servicios Web con PHP (SOAP) [Consultado el: febrero de 2008]. Disponible en:

<http://www.nociondigital.com/webmasters/php-tutorial-servicios-web-con-php-nusoap-detalle-168.html>.

[5]. Proceso Unificado de Rational. Publicado el: febrero 2007 de última actualización: febrero 2008.

Disponible en: http://es.wikipedia.org/wiki/Rational_Unified_Process.

[6]. CABALLERO, I. Práctica de Ingeniería de Software 3 [Consultado en: febrero de 2008]. Disponible en:

http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/LabTr1_VP.pdf.

BIBLIOGRAFÍA

1. **Caro, Patricio Salinas y Histchfeld K, Nancy.** Tutorial de UML.
<http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>.
2. **Brea, Orlando Fabián.** Introducción a los Web Services en PHP.
<http://www.desarrolloweb.com/articulos/1852.php>.
3. **RUMBAUGH, J. J., I. Y BOOCH, G.** El proceso unificado de desarrollo de software. 2000. vol. 1.
4. **Larman, Craig.** UML y Patrones.
5. **SOAP y WebServices.** 13 de Enero de 2003.
<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art51.asp>.
6. **Booch, G., Rumbaugh, J., Jacobson, I.** “El Lenguaje Unificado de Modelado”. Addison-Wesley. 1999
7. **PARADIGM, V.** Visual Paradigm for UML 6.0 SP1, 2007. [2007]. Disponible en:
<http://www.visual-paradigm.com/news/vpsuite30sp1/vpuml60sp1.jsp>
8. **PRESSMAN., R. S.** Tomado de la biblioteca UCI. Ingeniería del software. Un enfoque práctico
<http://bibliodoc.uci.cu/pdf/reg02689.pdf>.
9. **SCHMULLER, J.** Tomado de la biblioteca UCI. Aprendiendo UML en 24 horas.
<http://bibliodoc.uci.cu/pdf/reg00004.pdf> , 2000.
10. **UML**, consultado en Abril del 2008. Disponible en:
<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/c12.html>
11. **Pérez Sacristán, Juan Ignacio.** Web Services: XML-RPC, SOAP, sobre PHP, Perl, y otros conceptos. 2006. <http://www.programacion.net/tutorial/xmlrpcsoap>
12. **Saorín, Antonio.** El servidor web Apache.
<http://es.tldp.org/Tutoriales/doc-servir-web-escuela/doc-servir-web-escuela-html/apache.html>
13. **Vegas, Jesús.** El Servidor Web. 21 de Marzo de 2002.
<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>.

ANEXOS

Descripción de las tablas.

Nombre: SolicitudServicios.		
Descripción: Almacena la información referente a las solicitudes.		
Atributo	Tipo	Descripción
ci	CHAR	Llave primaria de la tabla
idServicio	INT	Llave primaria de la tabla Servicio.
idTipoCliente	INT	Llave primaria de la tabla TipoCliente.

Nombre: tAsesor.		
Descripción: Almacena el servicio y el tipo de uso.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
idTipoLineaSala	INT	Llave primaria de la tabla TipoLineaSala.

Nombre: tCanaltv.		
Descripción: Almacena la información referente al servicio de TV.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
numeroCanal	VARCHAR	Número del canal de TV.
nombreCanal	VARCHAR	Nombre del canal de TV.

Nombre: tClienteInvitado.		
Descripción: Almacena la información del cliente invitado.		
Atributo	Tipo	Descripción
ci	CHAR	Identificador de la tabla ClienteInvitado.
nombre	VARCHAR	Nombre del cliente.
apellidos	VARCHAR	Apellido del cliente.

Nombre: tConRecurso.		
Descripción: Almacena el servicio y los recursos de este.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
idTipoRecurso	INT	Llave primaria de la tabla TipoRecurso.

Nombre: tCurso.		
Descripción: Almacena el tipo de curso.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
idTipoCurso	INT	Llave primaria de la tabla TipoCursos.

Nombre: tDigitalDoc.		
Descripción: Almacena los datos referentes a la digitalización de documentos.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
fechaTope	VARCHAR	Fecha limite para la entrega.

Nombre: tDiseno.		
Descripción: Almacena los datos referentes al servicio de diseño.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
tituloCurso	VARCHAR	Nombre del curso.
fechaTope	VARCHAR	Fecha limite para la entrega.
idTipoDiseno	INT	Llave primaria de la tabla Diseño.

Nombre: tEboletin.		
Descripción: Almacena los datos del servicio de boletín informativo.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
tematica	VARCHAR	Tema que abordará el boletín.
email	VARCHAR	Dirección de correo electrónico.
idTipoBoletin	INT	Llave primaria de la tabla TipoBoletin.

Nombre: tFotocopiado.		
Descripción: Almacena los datos del servicio de fotocopiado.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
idTipoFotocopiado	INT	Llave primaria de la tabla TipoFotocopiado.

Nombre: tGrabacionProg.		
Descripción: Almacena los datos del servicio de grabación de programas.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
numeroCanal	VARCHAR	Número del canal de TV.
nombreCanal	VARCHAR	Nombre del canal de TV.
programa	VARCHAR	Nombre del programa de TV.
idTipoFormato	INT	Llave primaria de la tabla TipoFormato.

Nombre: tHemeroteca.		
Descripción: Almacena Los datos referentes al servicio de hemeroteca.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
autor	VARCHAR	Autor de la obra en cuestión.
obra	VARCHAR	Nombre de la obra.
idTipoLineaSala	INT	Llave primaria de la tabla TipoLineaSala.

Nombre: tImpresion.		
Descripción: Almacena los datos del servicio de impresión de documentos.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
cantPag	INT	Cantidad de páginas a imprimir.

Nombre: tMonitoreo.		
Descripción: Almacena los datos del servicio de monitoreo.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
tematica	VARCHAR	Tema que abordará el monitoreo.

Nombre: tObservatorio.		
Descripción: Almacena los datos del servicio de observatorio.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
idTipoObservatorio	INT	Llave primaria de la tabla TipoObservatorio.

Nombre: tPcportatil.		
Descripción: Almacena los datos del servicio de préstamo de PC portátil.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
idTipoLugar	INT	Llave primaria de la tabla TipoLugar.

Nombre: tPresDoc.		
Descripción: Almacena los datos del servicio de préstamo de documentos.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
idTipoDocumento	INT	Llave primaria de la tabla TipoDocumentos.

Nombre: tProgenvivo.		
Descripción: Almacena los datos del servicio de grabación de programas en vivo.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
nombrePrograma	VARCHAR	Nombre del programa.
numeroCanal	VARCHAR	Número del canal de TV.
nombreCanal	VARCHAR	Nombre del canal de TV.

Nombre: tReferencia.		
Descripción: Almacena los datos del servicio de referencia bibliográfica.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
tipoReferencia	VARCHAR	Tipo de referencia.
idTipoLineaSala	INT	Llave primaria de la tabla TipoLineaSala.

Nombre: tRegistroServicios.		
Descripción: Almacena los datos de los servicios ejecutados por los clientes.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
nombreCliente	VARCHAR	Nombre del cliente.
fechaServicio	VARCHAR	Fecha del servicio.

Nombre: tSalonTv.		
Descripción: Almacena del servicio de salón de TV.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
programa	VARCHAR	Nombre del Programa.
duracion_programa	VARCHAR	Duración del programa.
canales	VARCHAR	Nombre del canal.

Nombre: tServicio.		
Descripción: Almacena los datos de los servicio que brinda la entidad.		
Atributo	Tipo	Descripción
idServicio	INT	Identificador de la tabla Servicio.
estadoServicio	VARCHAR	Estado en que se encuentra el servicio.
descripcionServicio	VARCHAR	Objetivo de la solicitud.
duracionServicio	VARCHAR	Duración del servicio.
fechaServicio	VARCHAR	Fecha de ejecución del servicio.
idTipoServicio	INT	Llave primaria de la tabla TipoServicio.

Nombre: tServProg.		
Descripción: Almacena los datos del servicio de programas de televisión.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
numeroCanal	VARCHAR	Número del canal de TV.
nombreCanal	VARCHAR	Nombre del canal de TV.
programa	VARCHAR	Nombre del programa.
idTipoFormato	INT	Llave primaria de la tabla TipoFormato.
tipoFormato2	INT	Tipo de formato que desea el cliente.
idTipoServPrograma	INT	Llave primaria de la tabla TipoServPrograma.

Nombre: tTransFormato.		
Descripción: Almacena los datos del servicio de transferencia de formatos.		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
idTipoFormato	INT	Llave primaria de la tabla TipoFormato.
tipoFormato2	INT	Formato que desea.

Nombre: tWifi.		
Descripción: Almacena los datos del servicio de red inalámbrica (Wifi).		
Atributo	Tipo	Descripción
idServicio	INT	Llave primaria de la tabla Servicio.
tiempo	VARCHAR	Duración del servicio.
idTipoHorario	VARCHAR	Llave primaria de la tabla TipoHorario.
idTipoServiciosWifi	INT	Llave primaria de la tabla TipoServiciosWifi.

Nombre: tTipoHorarios.		
Descripción: Almacena los horarios del servicio de red inalámbrica (Wifi).		
Atributo	Tipo	Descripción
idTipoHorario	INT	Llave primaria de la tabla TipoHorarios.
nombreTipoHorario	VARCHAR	Horarios del servicio.

GLOSARIO

SOAP: Es un protocolo elaborado para facilitar la llamada remota a funciones a través de Internet, permitiendo que dos programas se comuniquen de una manera muy similar técnicamente a la invocación de páginas Web.

UML: Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.

XML: Es el acrónimo de eXtensible Markup Language (lenguaje de marcado extensible) desarrollado por el World Wide Web Consortium (W3C).

CASE: Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador), aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

CUS: Caso de uso del sistema.

C/C++: Lenguaje de programación compilado.

HTTP: Protocolo de Transferencia de Hipertexto. Protocolo para transferir archivos o documentos hipertexto a través de la red.

MySQL: Servidor de bases de datos recomendable para desarrollos que necesiten manejar numerosos registros y sesiones simultáneas.

PHP: Lenguaje de programación interpretado.

Usuario: Todo el personal del Centro Rector de Universidad para Todos.

ITIL: Biblioteca de Infraestructura de la Tecnología de la Información.

RMBM: La Red Municipal de Bibliotecas de Murcia.