

**Universidad de las Ciencias Informáticas**

**Facultad 6**



**Título:** Sistema de Gestión del Fondo de Materiales Audiovisuales en Formato Digital de la Batalla de Ideas.

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

**Autores:** Omar González Valdivia.  
Alejandro González García.

**Tutor:** Ing. Velmour Muñoz Casals.

**Asesor(es):** Msc. José Antonio Concepción.  
Lic. René Hidalgo Font.

Junio 2009

“Año del 50 Aniversario del triunfo de la Revolución”

*El hombre sale en busca de lo que ha de prometer  
Para mover sumundo, violentar su acontecer  
Le dicen que imposible, que eso no se puede hacer  
Lograr la maravilla de vencer..”*

*Pablo Milanés*

## **DECLARACIÓN DE AUTORÍA**

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los \_\_ días del mes de junio del 2009.

\_\_\_\_\_  
Firma de la Autor

\_\_\_\_\_  
Firma de la Autor

\_\_\_\_\_  
Firma de la tutora

\_\_\_\_\_  
Firma de la tutora

Datos de Contacto.

Velmour Muñoz Casals

[vcasals@uci.cu](mailto:vcasals@uci.cu)

[profeuci@oice.oa.ce.cu](mailto:profeuci@oice.oa.ce.cu)

Ingeniero en Ciencias Informáticas.

**Alejandro**

*Quisiera darles las gracias a todos los que tuvieron que ver con este triunfo, a los que me iniciaron en este mundo y que sin ellos nada hubiera sido posible, a todos mis compañeros de estudio a lo largo de mi carrera como estudiante, momentos difíciles pero siempre estuvimos como uno solo que era lo importante. A mi papá caramba, siempre un espejo donde mirarme, gracias por los consejos y toda la ayuda, se que sueñas con que trabajemos juntos, espero que algún día se logre, a mi madre por estar siempre para mi cuando más la necesite. A mis hermanos del bachillerato, Marlon, Raidel, Walter, Yoandry, en fin todos los que aun están y seguirán estando. A mi novia, siempre fuiste mi bastón donde apoyarme en momentos en los cuales creía que todo se acababa, a mi suegra, mi cuñada, Lucia, Níco, Miguel, gracias por todo, esto también es para ustedes, han tenido mucho que ver en ello. Agradecer a mis hermanos de causa, se que lo que aquí se ha forjado perdurará siempre, porque nada ha sido en vano, han sido 5 años, largos, duros y alegres pero creo que sin ustedes nunca lo hubiese logrado, muchas gracias por todo.*

**Omar**

*Quisiera darle las gracias a todos los que tuvieron que ver algo con mis logros hasta ahora, a los que me mostraron el camino que debía seguir; mi profesora Amelia, Ernestina y todas las que siguieron desde pre escolar hasta aquí, al profê Sergio. A mis compañeros de estudio que sin ánimo de competencia éramos uno la meta del otro y así nos superábamos. A mi madre quien más ha tenido que ver en mi desarrollo tanto intelectual como en la vida, ella ha sido para mí todo lo que yo quisiera ser para mis hijos cuando llegue la hora de ser padre, a mi padre siempre dándome sus consejos y también poniéndome metas bien altas seguro de que yo las alcanzaré. A mi novia la batería de reserva que alumbró en los momentos más oscuros de este último año, a mis suegros por tenerla y a su prima por presentármela aunque esa es una larga historia. Quisiera agradecer ahora a mis amigos, sin usted se hubiese sido por gusto, porque de no existir los amigos no sería posible mantener ese equilibrio que nos mantiene en pie aún cuando estamos en la más fina y resbaladiza cuerda floja. A todos los que aquí se identifican muchísimas gracias*

*De Alejandro.*

*A mi padre por todo su amor, su dedicación y su apoyo incondicional, por estar ahí siempre para mí, aunque no se entiendan pero siempre supieron hacer por un objetivo común, encaminarme correctamente en la vida.*

*A mis hermanos, mis tíos, mis abuelos, mis primos por todo el cariño y la atención que me han dedicado en todo momento, por ocupar un lugar tan especial en mi vida.*

*A mi novia por todo el amor, el cariño y el apoyo que me ha brindado en esta etapa de mi vida y por ser mi ángel de la guarda, por querer siempre lo mejor para mí.*

*A mi dúo de tesis Omar, Jose, Adolis, Julio, Mary, en fin todos aquellos que durante este tiempo supieron darme lo mejor incondicionalmente, por eso están donde están para mí.*

*A todas mis amistades que han estado presente en los buenos y malos momentos de mi vida, para todos es este triunfo.*

*De Omar:*

*Dedico este trabajo a todas esas personas que siempre supieron que yo podría llegar a donde estoy hoy. En especial a esas dos personas que aunque no se entiendan me tienen como factor común, mis padres A mi novia que me ha dado más apoyo que 100 bastones A mi hermano y mis sobrinos A fifi y Alberto que son como dos padres más, a mi tío siempre preguntándome como iba todo, a mis primos. Al profê Sergio que me ayudo pero además ayudo a mi novia cuando nadie más podía. A mis amigos, o bien hermanos porque considero que es una hermandad lo que hemos construido a través de los años: El Cory, el pacho, el Marlon y Julio (313), Marcel hermano de siempre, Sergio, Roger y Ernesto y a todos los otros hermanos y hermanas que confiaron en mí.*



Dentro de la Batalla de Ideas (BI) se está llevando a cabo un programa de identificación de largometrajes útiles en formato digital que tiene como objetivo que el país tenga un Fondo de Materiales de rápido acceso y utilización masiva que pueda respaldar los programas de Introducción de TV y VHS en las escuelas, el programa audiovisual dirigido a las escuelas y a la creación de los Video Club Juveniles.

El objetivo general del trabajo es el desarrollo de una solución informática que permita de manera automatizada multiplicar las posibilidades de utilización multifacética y operativa del Fondo de Materiales Audiovisuales en formato digital (DVD y algunos en Blue Ray), con más de 5 mil títulos, mediante el cual se amplíen las posibilidades de uso de estos Materiales ofreciendo el adecuado servicio que requieren los Programas de la Batalla de Ideas que utilizan los audiovisuales y la televisión.

Este documento recoge los resultados de todo el trabajo investigativo realizado. Se identifica y describe la problemática existente dentro del Fondo en cuanto a la gestión de la información y se dan argumentos para demostrar que la situación problemática requiere de un sistema nuevo, ninguna otra solución satisface todas sus necesidades.

Posteriormente se hace un análisis comparativo acerca de las tecnologías existentes y se seleccionan las más apropiadas para desarrollar una aplicación Web que se pueda actualizar de forma dinámica, además de abordar conceptos importantes y por último se muestran los resultados de la propuesta y se dejan algunas recomendaciones para el desarrollo futuro del mismo.

**Tabla de contenido**

**INTRODUCCIÓN.....1**

**CAPÍTULO I ..... ¡ERROR! MARCADOR NO DEFINIDO.**

**FUNDAMENTACIÓN TEÓRICA..... ¡ERROR! MARCADOR NO DEFINIDO.**

1.1 APLICACIONES CLIENTE-SERVIDOR..... ¡ERROR! MARCADOR NO DEFINIDO.

1.2 SISTEMA DE GESTIÓN DE INFORMACIÓN ..... ¡ERROR! MARCADOR NO DEFINIDO.

1.3 METODOLOGÍA A UTILIZAR. TECNOLOGÍAS Y TENDENCIAS ACTUALES¡ERROR! MARCADOR NO DEFINIDO.

    1.3.1 Metodologías de desarrollo ..... ¡Error! Marcador no definido.

        1.3.1.1 XP (Extreme Programing) ..... ¡Error! Marcador no definido.

        1.3.1.2 Rational Unified Process (RUP) ..... ¡Error! Marcador no definido.

            2.1.1 ¿Por qué RUP? .....9

    1.3.2 HERRAMIENTA CASE ..... ¡ERROR! MARCADOR NO DEFINIDO.

        1.3.2.1 Rational Rose..... ¡Error! Marcador no definido.

        1.3.2.2 Visual Paradigm..... ¡Error! Marcador no definido.

        1.3.2.3 ¿Por qué Visual Paradigm?..... ¡Error! Marcador no definido.

    1.3.3 LENGUAJE DE MODELADO ..... ¡ERROR! MARCADOR NO DEFINIDO.

        1.3.3.1 UML ..... ¡Error! Marcador no definido.

    1.3.4 HERRAMIENTAS DE DISEÑO..... ¡ERROR! MARCADOR NO DEFINIDO.

        Dreamweaver..... ¡Error! Marcador no definido.

    1.3.4 SOFTWARE LIBRE ..... ¡ERROR! MARCADOR NO DEFINIDO.

    1.3.5 LENGUAJES DE PROGRAMACIÓN..... ¡ERROR! MARCADOR NO DEFINIDO.

        PHP ..... ¡Error! Marcador no definido.

    1.3.6 HERRAMIENTAS PARA LA PROGRAMACIÓN ..... ¡ERROR! MARCADOR NO DEFINIDO.

        Zend Studio ..... ¡Error! Marcador no definido.

        PHP Designer ..... ¡Error! Marcador no definido.

        Eclipse ..... ¡Error! Marcador no definido.

        ¿Por qué Eclipse?..... ¡Error! Marcador no definido.

    1.3.6.1 FRAMEWORK ..... ¡ERROR! MARCADOR NO DEFINIDO.

        Codeigniter ..... ¡Error! Marcador no definido.

        Symfony..... ¡Error! Marcador no definido.

    1.3.7 SISTEMAS GESTORES DE BASE DE DATOS..... ¡ERROR! MARCADOR NO DEFINIDO.

        1.3.7.1 MySQL..... ¡Error! Marcador no definido.

        1.3.7.2 PostgreSQL ..... ¡Error! Marcador no definido.

        1.3.7.3 Microsoft Access..... ¡Error! Marcador no definido.

        ¿Por qué Postgre SQL?..... ¡Error! Marcador no definido.

    1.3.8.1 ARQUITECTURA CLIENTE-SERVIDOR ..... ¡ERROR! MARCADOR NO DEFINIDO.

    1.3.8.2 SERVIDOR WEB ..... ¡ERROR! MARCADOR NO DEFINIDO.

        1.3.8.2.1 Servidor Web Apache ..... ¡Error! Marcador no definido.

**CONCLUSIONES..... ¡ERROR! MARCADOR NO DEFINIDO.**

**CAPÍTULO II ..... ¡ERROR! MARCADOR NO DEFINIDO.**

<b>CARACTERÍSTICAS DEL SISTEMA</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
2.1 MODELO DEL NEGOCIO .....	¡ERROR! MARCADOR NO DEFINIDO.
2.2 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES	¡ERROR! MARCADOR NO DEFINIDO.
2.2.1 <i>Requerimientos Funcionales:</i> .....	¡Error! Marcador no definido.
2.2.2 <i>Requerimientos No Funcionales</i> .....	¡Error! Marcador no definido.
<i>El sistema podrá ser usado sobre los sistemas operativos Windows y Linux</i>	¡Error! Marcador no definido.
<i>Rendimiento</i> .....	¡Error! Marcador no definido.
<i>Software</i> .....	¡Error! Marcador no definido.
<i>Hardware</i> .....	¡Error! Marcador no definido.
2.3 <i>Definición de Actores del sistema</i> .....	¡Error! Marcador no definido.
CASOS DE USO DEL SISTEMA .....	¡ERROR! MARCADOR NO DEFINIDO.
DIAGRAMAS DE CASOS DE USO .....	¡ERROR! MARCADOR NO DEFINIDO.
DESCRIPCIONES DE LOS CASOS DE USO DEL SISTEMA.....	¡ERROR! MARCADOR NO DEFINIDO.
CONCLUSIONES.....	¡ERROR! MARCADOR NO DEFINIDO.
<b>CAPÍTULO III ANÁLISIS Y DISEÑO DEL SISTEMA</b>	¡ERROR! MARCADOR NO DEFINIDO.9
3.2 MODELO DEL DISEÑO .....	60
3.2.1 ARQUITECTURA .....	61
3.2.2 ARQUITECTURA CLIENTE-SERVIDOR .....	61
3.2.3 PATRONES DE ARQUITECTURA.....	62
3.2.3.1 Patrón Modelo-Vista-Controlador .....	62
3.2.4 DIAGRAMAS DE CLASES DEL DISEÑO.....	63
3.2.4.1 DIAGRAMAS DE DISEÑO.....	¡ERROR! MARCADOR NO DEFINIDO.4
3.2.5 DIAGRAMAS DE INTERACCIÓN. DIAGRAMAS DE SECUENCIA.....	70
3.2.5.1 DIAGRAMAS DE SECUENCIA .....	71
3.2.6 DIAGRAMA DE DESPLIEGUE .....	76
3.4 CONCLUSIONES .....	77
<b>IMPLEMETACIÓN DEL SISTEMA</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
4.1 DIAGRAMAS DE COMPONENTES .....	¡ERROR! MARCADOR NO DEFINIDO.
4.1.1 DIAGRAMAS DE COMPONENTES .....	80
4.2 CONCLUSIONES.....	¡ERROR! MARCADOR NO DEFINIDO.
<b>CONCLUSIONES GENERALES</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>RECOMENDACIONES</b> .....	¡ERROR! MARCADOR NO DEFINIDO.
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	87
<b>BIBLIOGRAFÍA</b> .....	91
<b>ANEXOS</b> .....	¡ERROR! MARCADOR NO DEFINIDO.

## INDICE DE TABLAS Y FIGURAS

TABLA 1.1 ACTORES DEL SISTEMA.....	35
TABLA 1.2 CASOS DE USO DEL SISTEMA.....	35
TABLA 1.3 DESCRIPCION DEL CASO DE USO GESTIONAR USUARIO.....	37
TABLA 1.4: DESCRIPCIÓN DEL CASO DE USO GESTIONAR GENEROS.....	42
TABLA 1.5: DESCRIPCIÓN DEL CASO DE USO GESTIONAR PELICULAS.....	43
TABLA 1.6: DESCRIPCIÓN DEL CASO DE USO GESTIONAR SERIE.....	45
TABLA 1.7: DESCRIPCIÓN DEL CASO DE USO GESTIONAR PRÉSTAMO.....	48
TABLA 1.8: DESCRIPCIÓN DEL CASO DE USO REALIZAR BÚSQUEDA.....	49
TABLA 1.9: DESCRIPCIÓN DEL CASO DE USO GENERAR REPORTE.....	50
TABLA 1.10: DESCRIPCIÓN DEL CASO DE USO AUTENTICAR USUARIO.....	51
FIGURA 1 DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	28
FIGURA 2 MODELO VISTA CONTROLADOR.....	62
FIGURA 3 DIAGRAMA DE CLASE DEL DISEÑO CU GESTIONAR INFORME.....	64
FIGURA 4 DIAGRAMA DE CLASE DEL DISEÑO CU GESTIONAR PELICULA.....	65
FIGURA 5 DIAGRAMA DE CLASE DEL DISEÑO CU GENERAR PRESTAMO.....	66
FIGURA 6 DIAGRAMA DE CLASE DEL DISEÑO CU GESTIONAR SERIE.....	67
FIGURA 7 DIAGRAMA DE CLASE DEL DISEÑO CU GESTIONAR DOCUMENTAL.....	68
FIGURA 8 DIAGRAMA DE SECUENCIA INSERTAR PELICULA.....	70
FIGURA 9 DIAGRAMA DE SECUENCIA ELIMINAR PELICULA.....	71
FIGURA 10 DIAGRAMA DE SECUENCIA INSERTAR DOCUMENTAL.....	72
FIGURA 11 DIAGRAMA DE SECUENCIA VISUALIZAR DOCUMENTAL.....	73
FIGURA 12 DIAGRAMA DE DESLPLIEGUE.....	74
FIGURA 13 DIAGRAMA DE CLASES PERSISTENTES.....	75
FIGURA 14 MODELO DE OBJETOS.....	76
FIGURA 15 DIAGRAMA DE COMPONENTES GENERAR INFORME.....	80
FIGURA 16 DIAGRAMA DE COMPONENTES GESTIONAR PRESTAMO.....	81
FIGURA 17 DIAGRAMA DE COMPONENTES GESTIONAR USUARIO.....	82
FIGURA 18 DIAGRAMA DE COMPONENTES GESTIONAR DOCUMENTAL.....	83

# Introducción

La Patria Cubana con 50 años de Revolución ha desarrollado toda una serie de estrategias de trabajo para elevar el nivel cultural y profesional de todo el país, que en estos momentos se encuentra inmerso en un proceso renovador social, denominado BI. Como parte de dicho proceso se concibió estratégicamente la creación de un Fondo de Materiales en formato digital de rápido acceso y utilización masiva que pudiera respaldar estos programas. Un Fondo de Materiales en formato digital de rápido acceso y utilización masiva que pudiera respaldar programas como:

1. La introducción de Televisión (TV) y Video Home System (VHS) en las escuelas.
2. Programa audiovisual para las escuelas primarias y de segunda enseñanza.
3. Creación de los Video Club Juveniles.

De ahí que se comienza la creación con lo más destacado de la cinematografía mundial, colecciones de documentales y de series de interés cultural, científico y recreativo, todos en formato DVD y algunos en Blu Ray. Se han realizado misiones de adquisición en el exterior (Francia, España y Venezuela) en siete operaciones de compra, por lo que hasta el momento el Fondo cuenta con más de 5000 títulos, que se encuentran clasificados en un sistema automático de búsqueda y control, adquirido de Internet, en el que se han estado registrado los datos fundamentales de cada uno de los materiales para la búsqueda ágil de las solicitudes realizadas. Este sistema a pesar de cumplir con algunas de las funcionalidades básicas, no permite almacenar la mayor cantidad de información de cada uno de los materiales existentes, no presenta una interfaz amigable, no posee una Base de Datos potente capaz de soportar la cantidad de información que se pretende recopilar en un futuro y como elemento fundamental no es un sistema propietario; de aquí que el centro del problema radique en ¿Cómo obtener un sistema que gestione la información detallada de cada uno de los Materiales Audiovisuales del Fondo de la Batalla de Ideas?

El **objeto de estudio** del presente trabajo radica en cómo contribuir al desarrollo de sistemas para la gestión de la información de materiales audiovisuales. Delimitando como el **campo de acción** el desarrollo de un sistema para la gestión de la información en el Fondo de Materiales Audiovisuales en formato digital de la Batalla de Ideas.

El **objetivo general** del trabajo es desarrollar un sistema que permita de manera automatizada multiplicar las posibilidades de utilización multifacético y operativa del Fondo de Materiales, mediante el cual se amplíen las posibilidades de uso de estos, ofreciendo el adecuado servicio que requieren los Programas de la Batalla de Ideas que utilizan los audiovisuales y la televisión.

Determinando como **objetivos específicos** para dar cumplimiento al objetivo general:

- ✍ Realizar el análisis de los sistemas para la gestión de la información de materiales audiovisuales.
- ✍ Diseñar un sistema para la gestión de la información de los Materiales Audiovisuales del Fondo de la Batalla de Ideas.
- ✍ Implementar un sistema para la gestión de la información de los Materiales Audiovisuales del Fondo de la Batalla de Ideas.

Para darle solución a los objetivos propuestos y resolver la situación problemática planteada, se proponen las siguientes **tareas Investigativas**:

- ✍ Estudio y definición de la metodología y herramientas actuales para el desarrollo de sistemas de gestión de información.
- ✍ Estudio del negocio donde se implantará el sistema
- ✍ Realización de entrevistas con el cliente para identificar las funcionalidades del sistema.
- ✍ Definición de las funcionalidades que debería tener el sistema.
- ✍ Descripción de las funcionalidades que se obtendrán.

Con la implementación de la solución informática que se propone se espera obtener como resultado:

Disponer de un sistema visible para diversos usuarios autorizados que facilite la gestión de la información del Fondo de Materiales Audiovisuales en formato digital de la BI. Los usuarios van a tener la posibilidad de un sistema que contenga la información más detallada de cada uno de los materiales del fondo.

El presente trabajo de diploma está estructurado de la siguiente manera:

**Capítulo 1 Fundamentación Teórica:**

Se abordarán temas sobre la utilidad de poner las nuevas tecnologías de la informática y las comunicaciones al servicio de los SGI, se expondrán las características de las herramientas y tecnologías escogidas para la solución del problema, así como se argumenta la elección de las mismas.

**Capítulo 2 Características del sistema:**

En este capítulo se abordará de forma general los requisitos funcionales y no funcionales, los actores que intervienen en el sistema, los casos de uso y sus descripciones, el diagrama de casos de uso del sistema, así como sus respectivas descripciones.

**Capítulo 3 Análisis y Diseño del sistema:**

En este capítulo se abordarán los estilos arquitectónicos y patrones del diseño evidenciando su utilización, los diagramas de clases del diseño, los diagramas de interacción y el modelo de despliegue.

**Capítulo 4 Implementación del sistema:**

En este capítulo se realizará el diagrama de componentes y se presentarán pantallas de la aplicación.

# CAPITULO I. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se brinda una visión general de los aspectos relacionados con los sistemas de Gestión de la Información; y los conceptos necesarios para el estudio, clasificación y creación de los mismos. También se describen los principales conceptos asociados al dominio del problema que son necesarios para entenderlo, así como la propuesta de solución del trabajo. Además se muestran los aspectos más importantes de distintos software desarrollados en el mundo para el proceso de gestión de la información. Además se establece la metodología de trabajo a seguir, así como las técnicas y herramientas que se aplicarán durante la investigación.

## 1.1 Aplicaciones Cliente-Servidor.

Son aplicaciones que proporcionan al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o clientes, resultan en un trabajo realizado por otros computadores llamados servidores. Así se sustenta la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información.

Decir además que las aplicaciones cliente/servidor son una relación entre procesos corriendo en máquinas separadas donde el servidor es un proveedor de servicios y el cliente es un consumidor de servicios. Tanto el servidor como el cliente interactúan por un mecanismo de pasaje de mensajes: pedido de servicio y respuesta. **[1]**

En dicha arquitectura toda aplicación de un sistema de información está caracterizada por tres componentes básicos:

Presentación / Captación de Información

Procesos

Almacenamiento de la Información

## 1.2 Sistema de Gestión de Información

Si nos concentramos en el significado etimológico del concepto, entenderíamos por Sistema de Gestión de la Información como un proceso que incluye operaciones como extracción, manipulación,



tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y los derechos de los usuarios sobre la misma.

Se establece, por lo tanto, como una disciplina transversal que aparece entrelazada en todas las diferentes capas o tejidos de una organización, en todos los conceptos de administración (recursos humanos, marketing, finanzas, estrategia, operaciones,...) y les proporciona soporte.

### **1.3 Metodología a utilizar. Tecnologías y tendencias actuales**

#### **1.3.1 Metodologías de desarrollo**

Todo desarrollo de software es riesgoso y difícil de controlar, pero si se utiliza una metodología de desarrollo, se consiguen clientes y desarrolladores más satisfechos con el resultado. La Ingeniería de Software tiene como objetivo construir un producto software o mejorar uno existente. Una metodología define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo. Su objetivo es elevar la calidad del sistema permitiendo una mayor transparencia y control sobre todo el proceso. Estas proporcionan normas para el desarrollo eficiente de un software de calidad, reducen el riesgo y hacen el proyecto más predecible. [2]

En la actualidad existen varias metodologías para el desarrollo de software como son: Extreme Programming (XP), Microsoft Solution Framework (MSF), Desarrollo Guiado por la Funcionalidad (FDD), Rational Unified Process (RUP) entre otras.

##### **1.3.1.1 XP (Extreme Programming)**

La Programación Extrema es una metodología ligera de desarrollo de Software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Posee una serie de prácticas para la planificación, diseño, codificación y prueba de software, logrando con estas el diseño. Posee además una gran agilidad en el desarrollo, conseguida mediante una comunicación intensiva del equipo en, una confianza en todos los desarrolladores, una disminución notable del protocolo y de las jerarquías dentro y fuera del equipo y una auto evaluación intensiva incluso a nivel de cada desarrollador. Todo esto no está del todo implícito en el ciclo de desarrollo y suele pasarse por alto, pero para algunos de nosotros es lo que hace a la XP tan valiosa como metodología. [3]

### 1.3.1.2 Rational Unified Process (RUP)

Es una metodología pesada, es efectiva para el desarrollo y construcción de software basado íntegramente en UML como soporte a la metodología. A través de un proyecto guiado por RUP, los requerimientos funcionales son expresados en forma de Casos de Uso, que guían la realización de una arquitectura ejecutable de la aplicación. Además el proceso focaliza el esfuerzo del equipo en construir los elementos críticos estructuralmente y del comportamiento (llamados Elementos Arquitecturales) antes de construir elementos menos importantes. También la mitigación de los riesgos más importantes guía la definición del alcance en las primeras etapas del ciclo de vida.

Posee 3 características fundamentales:

- ✍ **Dirigido por casos de uso:** Los casos de uso constituyen el hilo conductor del proceso de desarrollo. Capturan requerimientos funcionales y representan piezas de funcionalidad que brindan un resultado de valor al usuario.
- ✍ **Centrado en la arquitectura:** Comprende los aspectos estáticos y dinámicos más importantes del sistema, ya que la arquitectura describe los elementos del modelo que son importantes para la construcción, y desarrollo del software.
- ✍ **Iterativo e incremental:** El trabajo se divide en piezas pequeñas o mini proyectos. Las iteraciones hacen referencias a pasos en el flujo de trabajo y los incrementos, al crecimiento del producto. Cada iteración que se realice en el proyecto debe ser planificada.

### 1.3.2 Herramienta Case

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) de modelado con UML brindan la posibilidad de aplicar una metodología de análisis y diseño con abstracción del código fuente, de manera que la arquitectura y el diseño son más obvios y fáciles de comprender y modificar. En la medida que el proyecto es más grande y complejo, resulta más necesario utilizar una herramienta CASE ya que aporta grandes beneficios para todos los involucrados en un proyecto, por ejemplo, jefe del proyecto, analistas, arquitectos, desarrolladores y otros.

#### 1.3.2.1 Rational Rose

Es una herramienta de diseño unificada orientada al objeto del software de la lengua que modela (UML) prevista para modelar de la representación visual y la construcción componente de los usos del software del empresa-nivel. Más o menos de la misma manera un director de teatro bloquea hacia fuera un juego, las aplicaciones Rose racional de un diseñador del software de crear visualmente (modelo) el marco para un uso bloqueando hacia fuera clasifica con los agentes (figuras del palillo), los elementos del caso de uso (óvalos), los objetos (rectángulos) y los mensajes/las relaciones (flechas) en un diagrama de secuencia usando símbolos de la arrastrar-y-gota.

Dos características populares de Rose racional son su capacidad de proporcionar el desarrollo iterativo e ingeniería ida-vuelta. Rose racional permite que los diseñadores se aprovechen del desarrollo iterativo (a veces llamado desarrollo evolutivo) porque el nuevo uso se puede crear en etapas con la salida de una iteración que se convierte en la entrada al siguiente. (Aquí es en contraste con el desarrollo de la cascada donde el proyecto entero se termina de comienzo al final antes de que un usuario consiga intentarlo hacia fuera.) Entonces, como el revelador comienza a entender cómo los componentes obran recíprocamente y hacen modificaciones en el diseño, Rose racional puede realizarse qué es llamada “ingeniería ida-vuelta” yendo detrás y la puesta al día del resto del modelo para asegurar el código sigue siendo constante. [4]

### **1.3.2.2 Visual Paradigm**

Visual Paradigm es una herramienta CASE que brinda la posibilidad a los desarrolladores de visualizar y diseñar los elementos de software, utiliza como lenguaje de modelado UML y permite a los desarrolladores diseñar un producto de forma rápida. Facilita la interoperabilidad con otras herramientas CASE y se integra con entorno de desarrollo. Tiene disponibilidad para disímiles versiones y para integrarse en múltiples plataformas. Esta herramienta necesita de altos requerimientos computacionales para su óptima ejecución.

### **1.3.2.3 ¿Por qué Visual Paradigm?**

Visual Paradigm es una colección premiada de herramientas que facilita a las organizaciones visuales y el diagrama de diseño, integrar y desplegar su misión crítica y aplicaciones en sus bases de datos. Nos permite además en el proceso de desarrollo de software a sobresalir en todo el modelo de construcción y despliegue de software y acelerar al máximo tanto el grupo de trabajo como las contribuciones individuales. Racional Rose además de ser poco amigable, posee una instalación costosa, y sus proyectos a veces son difíciles de empezar y tiene como característica principal que no están definidas las iteraciones se van a dar en una fase determinada. Puede no resultar adecuado para proyectos pequeños.

### **1.3.3 Lenguaje de Modelado**

Los lenguajes de modelado permiten la modelación de objetos mediante un conjunto estandarizados de símbolos y técnicas de modelado. Modelan una simplificación de la realidad donde se toma de los objetos los aspectos significativos brindando así una mejor comprensión del sistema.

#### **1.3.3.1 UML**

Lenguaje Unificado de Modelado (UML), es un lenguaje gráfico utilizado para visualizar, especificar y documentar los elementos del software. UML permite hacer un modelo del sistema, especifica pero no describe los métodos y procesos del sistema o como implementarlo, sino lo que supuestamente hará el sistema.

El UML esta compuesto por diversos elementos gráficos que se combinan para formar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema. UML permite que diseñadores diferentes modelen sistemas diferentes y puedan ampliamente entender cada uno los diseños de los otros. El lenguaje de modelado escogido para la modelación del sistema, es UML por las potencialidades de sus características y su grado de comprensión.

### **1.3.4 Herramientas de diseño**

Los factores o atributos de calidad de una aplicación o sitio Web que influirán en dicho diseño y programación se puede clasificar en aquellos relacionados con: la calidad y utilidad de los contenidos; la calidad del lenguaje de programación, la calidad del servicio y asistencia del proveedor; y la calidad del diseño de la aplicación. Para el trabajo se utilizaron los siguientes:

#### **1.3.4 Software Libre**

El software libre es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Más precisamente, se refiere a cuatro tipos de libertades para los usuarios del software:

- ? La libertad de ejecutar el programa, para cualquier propósito (libertad 0).
- ? La libertad de estudiar cómo trabaja el programa, y adaptarlo a sus necesidades (libertad 1). El acceso al código fuente es una condición necesaria.
- ? La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).

- ? La libertad de mejorar el programa y publicar sus mejoras, y versiones modificadas en general, para que se beneficie toda la comunidad (libertad 3). El acceso al código fuente es una condición necesaria.

Un programa es software libre si los usuarios tienen todas esas libertades. Entonces, debería ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución, a cualquiera en cualquier parte. El ser libre de hacer estas cosas significa, entre otras cosas, que no tiene que pedir o pagar el permiso. También debería tener la libertad de hacer modificaciones y usarlas en privado, en su propio trabajo u obra, sin siquiera mencionar que existen. Si publica sus cambios, no debería estar obligado a notificarlo a alguien en particular, o de alguna forma en particular.

La libertad de ejecutar el programa significa la libertad para cualquier tipo de persona u organización de usarlo en cualquier tipo de sistema de computación, para cualquier tipo de trabajo y propósito, sin estar obligado a comunicarlo a su programador, o alguna otra entidad específica. En esta libertad, el propósito de los *usuarios* es el que importa, no el propósito de los *programadores*. Como usuario es libre de ejecutar un programa para sus propósitos; y si lo distribuye a otra persona, también es libre para ejecutarlo para sus propósitos, pero usted no tiene derecho a imponerle sus propios propósitos.

#### **[8]**

La libertad de redistribuir copias debe incluir las formas binarias o ejecutables del programa, así como el código fuente; tanto para las versiones modificadas como para las no lo están. (Distribuir programas en forma de ejecutables es necesario para que los sistemas operativos libres se puedan instalar fácilmente). Resulta aceptable si no existe un modo de producir una formato binario o ejecutable para un programa específico, dado que algunos lenguajes no incorporan esa característica, pero debe tener la libertad de redistribuir dichos formatos si encontrara programara una forma de hacerlo.

### **1.3.5 Lenguajes de programación**

#### **PHP**

PHP es un lenguaje interpretado basado principalmente en C, C++ y Java, con los que comparte prácticamente toda su sintaxis y semántica, y aporta también algunas características de lenguajes interpretados como Perl. Debido a esto, una de sus principales características es que la curva de aprendizaje para programadores que ya conozcan estos lenguajes es muy suave, prácticamente pueden sentarse delante del ordenador y comenzar a escribir código. Tiene buena integración con el Servidor Apache y su capacidad de acceder a múltiples sistemas de Bases de Datos distintos, lo ha convertido en un fuerte competidor frente a las “soluciones” de Microsoft. **[9]** PHP es un lenguaje

embebido en páginas HTML y que se ejecuta en el Servidor. Es fácil de aprender comparado con otros mecanismos para obtener la misma funcionalidad. No requiere un conocimiento exhaustivo del lenguaje de programación. A diferencia de Perl, PHP tiene una sintaxis muy fácil de comprender y a diferencia de ASP, no requiere conocer más de un lenguaje de programación o de la instalación de módulos externos o comerciales para realizar tareas más complicadas no previstas en el lenguaje más usado (Visual Basic Script). **[10]** Teniendo en cuenta las características y las comparaciones anteriormente descritas se ha seleccionado PHP (versión 5.0) para el desarrollo de la aplicación Web. Esta versión presenta las características necesarias para trabajar con una Programación Orientada a Objeto (POO), permitiendo la reutilización de funciones o métodos.

### **1.3.6 Herramientas para la programación.**

#### **Zend Studio**

Se trata de un programa de la casa Zend, impulsores de la tecnología de Servidor PHP, orientada a desarrollar aplicaciones Web en lenguaje PHP. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

Zend Studio es compatible con PHP 5, el framework incluye validación, es multiplataforma (compatible con Linux, Windows y Mac), incluye todos los componentes necesarios durante el ciclo de vida de una aplicación en PHP, agiliza el trabajo, cuenta con un buen depurador, infinitas opciones que permiten un magnifico desarrollo profesional de aplicaciones. A pesar de que el mismo es un software propietario, presenta algunas ventajas importantes sobre los editores mencionados con anterioridad, es por esto que se ha seleccionado para el desarrollo de la aplicación. **[11]**

#### **PHP Designer**

Es un completo entorno de desarrollo y programación especialmente diseñado para desarrolladores de PHP, aunque también permite trabajar con comodidad en otros lenguajes de programación como HTML, XHTML, CSS y SQL. Ofrece toda una serie de asistentes y diálogos integrados que facilitan en todo momento tu tarea, además de acceso directo a librerías de código o scripts de uso habitual, utilidades diversas y toda suerte de herramientas, todo ello en una interfaz de diseño sencillo y elegante que puedes personalizar con nada menos que dieciocho temas distintos. Cuenta con cliente de FTP y navegador de ficheros integrado, utilidades de corrección y auto completado, búsqueda integrada en Google y soporte para proyectos Posee un editor de HTML a diferencia de Zend Studio y

consume muchos menos recursos que otros entornos de desarrollo como el propio Zend Studio y Eclipse. [12]

## **Eclipse**

Eclipse es un IDE (Entorno de Desarrollo Integrado) tan potente como popular que incorpora un sin fin de utilidades para simplificar la labor de los programadores. Aparte de ser un entorno de desarrollo supercompleto, una de las particularidades más interesantes para la comunidad es que es de código libre y gratuito. Para Eclipse existen diversos plugins o añadidos para proveer de nuevas utilidades al programa, enfocadas a diversos usos que los distintos tipos de programadores pueden necesitar. Unos de los añadidos de Eclipse que más pueden interesar a los desarrolladores de páginas web sería el módulo para programación en PHP. [13]

### **¿Por qué Eclipse?**

Eclipse es completamente gratuita y libre, al ser tan usada, tiene un desarrollo constante y muchísimo soporte, diferencia de Zend Studio y PHP Designer por los cuales hay que pagar grandes sumas de dinero, para su compra y las licencias posteriores. Eclipse es muy completo, posee innumerables herramientas que podemos utilizar de un IDE, es multiplataforma y maneja muchas perspectivas, es decir cada uno tiene su botón y pestaña a donde le quede más cómodo, tiene cientos de plugins. Zend Studio y PHP Designer ambos no tienen la función ctrl.+click para inspeccionar, funciones constante, no poseen una lista de funciones declaradas en un archivo PHP, en el caso del Zend Studio que no posee la lista vertical, algo muy importante que se debería incluir en los IDE de este tipo.

#### **1.3.6.1 Framework**

Los Framework ayudan en el desarrollo de software, proporcionan una estructura definida la cual permite crear aplicaciones con mayor rapidez. Ayudan a la hora de realizar el mantenimiento del sitio gracias a la organización durante el desarrollo de la aplicación. Los mismos son desarrollados con el objetivo de brindarles a los programadores y diseñadores una mejor organización y estructura a sus proyectos. Se utiliza la Programación Orientada a Objetos (POO), permitiendo la reutilización del código. [14] Existen varios framework que facilitan el desarrollo de aplicaciones Web dentro de los que se encuentran: Symfony, CakePHP y Prado.

## **Codeigniter**

Es un framework para desarrollo de aplicaciones en PHP. Es Open Source, tiene una interfaz simple y un acceso a sus librerías bien estructurado. Es adecuado para desarrollos que no requieran un

framework que marque mucho la aplicación, y para cuando sea necesario mucho rendimiento. Está pensado para aquellas aplicaciones que se ejecutan en hosting compartido que ejecutan muchas versiones de PHP con diferentes configuraciones. Presenta buena documentación. **[15]** Ventajas

- ✍ Rendimiento excelente
- ✍ Compatible con varias versiones de PHP
- ✍ Casi no tiene configuraciones
- ✍ No requiere usar la línea de comandos
- ✍ No requiere reglas de codificación estrictas
- ✍ No necesitas librerías tan grandes como PEAR
- ✍ Sin complejidad, favoreciendo las soluciones simples

## Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web.

Symfony está desarrollado completamente con PHP. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL. Se puede ejecutar tanto en plataformas Unix, Linux, etc.) como en plataformas Windows. Algunas de sus características **[16]**:

- ✍ Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Linux y estándares)
- ✍ Independiente del sistema gestor de bases de datos
- ✍ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✍ Basado en la premisa de "*convenir en vez de configurar*", en la que el desarrollador solo debe configurar aquello que no es convencional.
- ✍ Sigue la mayoría de *mejores prácticas* y patrones de diseño para la web.



- ✍ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✍ Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo
- ✍ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Teniendo en cuenta estas características, la buena documentación que presenta, la Comunidad de Desarrollo existente que posibilita un rápido aprendizaje, que trabaja con el Modelo Vista Controlador y que es una herramienta de desarrollo fácil para crear aplicaciones Web simples y elegantes, se utiliza como framework Symfony.

### 1.3.7 Sistemas Gestores de Base de Datos.

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de Datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- ✍ Definir una Base de Datos: especificar tipos, estructuras y restricciones de datos.
- ✍ Construir la Base de Datos: guardar los datos en algún medio controlado por el mismo SGBD.
- ✍ Manipular la Base de Datos: realizar consultas, actualizarla, generar informes.

Algunas de las características deseables en un Sistema Gestor de Base de Datos SGBD son:

- ✍ Control de redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- ✍ Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- ✍ Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad. **[17]**

#### 1.3.7.1 MySQL.

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como

también de la marca. Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL. [18]

Como principales características que posee este SGBD tenemos que:

- ? Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
- ? Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.).
- ? Gran portabilidad entre sistemas.
- ? Soporta hasta 32 índices por tabla.
- ? Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- ? Soporta gran cantidad de tipos de datos para las columnas.

MySQL surgió cómo una necesidad de un grupo de personas sobre un gestor de bases de datos rápido, por lo que sus desarrolladores fueron implementando únicamente lo que precisaban, intentando hacerlo funcionar de forma óptima. Es por ello que, aunque MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas de sus principales características:

- ? Subconsultas: Tal vez ésta sea una de las características que más se echan en falta, aunque gran parte de las veces que se necesitan, es posible reescribirlas de manera que no sean necesarias.
- ? Triggers y Procedures: Se tiene pensado incluir el uso de procedures almacenados en la base de datos, pero no el de triggers, ya que los triggers reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.
- ? Transacciones: a partir de las últimas versiones ya hay soporte para transacciones, aunque no por defecto (se ha de activar un modo especial).
- ? Integridad referencial: aunque sí que admite la declaración de claves ajenas en la creación tablas, internamente no las trata de forma diferente al resto de campos.

### **1.3.7.2 PostgreSQL.**

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS), considerado como el sistema de Bases de Datos de código abierto más avanzado del mundo. [19]

Dentro de sus características presenta las siguientes:

- ✍ Una documentación muy bien organizada, pública y libre, para los usuarios.
- ✍ Comunidades muy activas, varias comunidades en castellano.
- ✍ Es multiplataforma
- ✍ Altamente adaptable a las necesidades del cliente
- ✍ Soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python.
- ✍ Soporte de todas las características de una Base de Datos profesional (triggers, procedimientos almacenados, funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas, etc.)
- ✍ PostgreSQL ofrece muchas ventajas respecto a otros sistemas de Bases de Datos:
  - ✍ En PostgreSQL no hay costo asociado a la licencia del software. Esto tiene varias ventajas adicionales: no existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento, flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.
  - ✍ Extensible: El código fuente está disponible para todos sin costo.
  - ✍ Diseñado para ambientes de alto volumen: PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de Bases de Datos comerciales usan también esta tecnología. [14]

PostgreSQL está ampliamente considerado como el sistema de Bases de Datos de código abierto más avanzado del mundo. Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la Base de Datos. Es multiplataforma. Presenta una buena documentación. Es por ello que lo elegimos como Gestor de Base de Datos para desarrollar la aplicación Web.

### **1.3.7.3 Microsoft Access.**

Microsoft Office Access ayuda a los trabajadores de la información a controlar y crear informes de datos de forma rápida y sencilla gracias a la interfaz de usuario de Microsoft Office Fluent y funciones de diseño interactivas que no requieren conocimientos especializados de bases de datos. Los usuarios pueden compartir la información a través del Web en listas de Microsoft Office Share Point Server en las que se pueden realizar auditorías y copias de seguridad. Además es compatible con SQL y Visual

Basic, gran facilidad para la creación de consultas (vistas) , formularios e informes. Es compatible con Visual Basic.

Entre sus principales características tenemos:

- ✍ Empezar rápidamente usando soluciones prediseñadas.
- ✍ Crea varios informes con vistas diferentes de la misma información.
- ✍ Crear tablas rápidamente sin preocuparse de la complejidad de las bases de datos.
- ✍ Obtener acceso a información procedente de orígenes diversos.

### **¿Por qué PostgreSQL?**

Luego de ser analizadas las características y potencialidades de los distintos SGBD presentados y teniendo en cuenta las bases de Informatización que existen en estos momentos en el Fondo de Materiales en Formato Digital y las de la aplicación a desarrollar se decide usar el Postgre SQL como SGBD, por las siguientes razones:

Postgre no tiene precio en el mercado, se puede acceder de forma gratuita, diferencia de Microsoft Access, que está fabricado por el gigante del software, Microsoft. Además Access es un SGBD que no te da los principios de un administrador, si tenemos en cuenta su robustez, navegabilidad, también podemos decir que con muchos usuarios de red va lento, la seguridad no es real, ya que es muy fácil acceder a una base de datos Access aunque este protegida. Las replicas son lentas. En el caso de MySQL es un SGBD no muy rápida a la hora de utilizar el motor no transaccional, puede provocar problemas de integridad, en entornos de alta concurrencia. No incluye además librerías de API nativo, estas son librerías preparadas para usarlas como paquetes embebidos en las aplicaciones, como una forma de acelerar el acceso al gestor y al sector de aplicaciones en gestión.

### **1.3.8 Arquitectura**

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. [21]

#### **1.3.8.1 Arquitectura Cliente-Servidor**

La arquitectura Cliente-Servidor se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD).

Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red. Podemos decir además que esta arquitectura necesita 3 tipos de software para su correcto funcionamiento:

- ✍ **Software de gestión de datos:** este software se encarga de la manipulación y gestión de los datos almacenados y requeridos por las diferentes aplicaciones. Normalmente este software se aloja en el Servidor.
  
- ✍ **Software de desarrollo:** este tipo de software se aloja en los Clientes y solo en aquellos que se dediquen al desarrollo de aplicaciones.
  
- ✍ **Software de interacción con los usuarios:** reside en los clientes y es la aplicación gráfica de usuario para la manipulación de datos, siempre a nivel usuario (consultas principalmente).

A parte de estos existen más aplicaciones software para el correcto funcionamiento de esta arquitectura pero ya están condicionados por el tipo de sistema operativo instalado, el tipo de red en la que se encuentra. [22]

La arquitectura es un artefacto decisivo en la calidad del software que se desarrolla. Su evaluación mitiga los diferentes riesgos asociados con el desarrollo del software, tomar acciones tempranas y valorar los atributos no funcionales sin esperar a que el software se construya. Por los elementos anteriormente mencionados se decide utilizar la arquitectura Cliente-Servidor para la creación del sistema, debido a que la misma permite una mayor organización del código, una funcionalidad más proporcionada, además de contar con una mayor fortaleza y un menor riesgo de que intrusos corrompan el sistema.

### 1.3.8.2 Servidor Web

Un Servidor Web es un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo

HTTPS. [23] Dependiendo del tipo de petición, el Servidor buscará una página Web o bien ejecutará un programa en el Servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. Algunos servidores web permiten también especificar directivas de seguridad (quién puede acceder a los recursos), mientras que otros hacen posible la especificación de los ficheros que se deben considerar como índice del directorio. [24]

#### **1.3.8.2.1 Servidor Web Apache.**

Apache es un Servidor Web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características se destacan:

- ? Es modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- ? Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- ? Se desarrolla de forma abierta
  
- ? Es extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del Servidor [25].

Apache está diseñado para ser un Servidor Web rápido, potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular. Este diseño permite a los administradores de sitios Web elegir que características van a ser incluidas en el Servidor seleccionando que módulos se van a cargar.

#### **Conclusiones.**

En este capítulo se abordaron los disímiles conceptos y términos para lograr un mayor entendimiento al problema en cuestión, así como las diferentes tecnologías de desarrollo web, tendencias actuales, metodologías de desarrollo y herramientas a usar en la solución del mismo cada una con sus

características haciéndolas ver apropiadas para este tipo de proyecto. Luego de un profundo análisis de todos estos aspectos mencionados para elegir a los más adecuados para la construcción del sistema que se pretende desarrollar se decidió utilizar como metodología de desarrollo RUP y herramienta case Visual Paradigm, como lenguaje de programación PHP y Symfony como framework ya que está diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web con la ayuda de herramientas que agilizan los procesos de implementación. Como SGBD PostgreSQL ya que permite soportar grandes volúmenes de información, teniendo como referencia las grandes empresas que manejan mucha información utilizan esta tecnología y presentan buenos resultados.

# CAPITULO II. CARACTERÍSTICAS DEL SISTEMA

## **2.1 Introducción.**

En este capítulo se realiza una descripción del objeto de estudio. Se utiliza el modelo de negocio para describir los procesos, se definen las funcionalidades de la aplicación a través de los requerimientos funcionales y no funcionales; los actores y el diagrama de casos de usos del sistema, finalmente se realiza la descripción de los casos de usos.

## **2.2 Requerimientos Funcionales y No Funcionales**

### **2.2.1 Requerimientos Funcionales:**

Los requerimientos funcionales son capacidades o condiciones que deben ser alcanzadas o poseídas por un sistema o componente para satisfacer un contrato, estándar, u otro documento impuesto formalmente. Entre los requerimientos funcionales de este sistema se encuentran



## **R1 Autenticar Usuario**

### **R2 Gestionar Géneros**

**R2.1** Insertar nuevo género en el sistema.

**R2.2** Modificar géneros existentes en el sistema.

**R2.3** Eliminar géneros existentes en el sistema.

### **R3 Gestionar Películas**

**R3.1** Insertar nueva película en el sistema.

**R3.2** Modificar películas existentes en el sistema.

**R3.3** Eliminar películas existentes en el sistema.

### **R4 Gestionar Series**

**R4.1** Insertar nueva serie en el sistema.

**R4.2** Modificar series existentes en el sistema.

**R4.3** Eliminar series existentes en el sistema.

### **R5 Gestionar Préstamos**

**R5.1** Insertar nuevo préstamo en el sistema.

**R5.2** Modificar préstamo existente en el sistema.

**R5.3** Eliminar préstamo existentes en el sistema.

## **R6 Gestionar Usuario**

**R6.2** Insertar nuevo usuario en el sistema.

**R6.3** Modificar usuario ya existente en el sistema.

**R6.4** Eliminar usuario ya existente en el sistema.

## **R7 Realizar Búsqueda**

**R7.1 Realizar búsquedas de películas.**

**R7.1.1** Por Título

**R7.1.2** Por Duración

**R7.1.3** Por Género

**R7.1.4** Por Año

**R7.1.5** Por País

## **R7.2 Realizar búsquedas de Series**

**R7.2.1** Por Título

**R7.2.2** Por Duración

**R7.2.3** Por Cantidad de CD

**R7.2.4** Por Año

**R7.2.5** Por País

## **R7.3 Realizar búsquedas de Préstamos.**

**R7.3.1** Por Entidad que solicita el préstamo

**R7.3.2** Por Cantidad de Títulos que conforman el préstamo

**R7.3.3** Por Nombre de Persona que recibe

**R7.3.4** Por Año

## **R8 Generar Reporte**

### **2.2.2 Requerimientos No Funcionales.**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a requerimientos funcionales, es decir una vez que sepamos lo que el sistema debe hacer, podemos determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser.

### **Usabilidad**

Se garantizará un acceso fácil y rápido a los usuarios. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora y de un ambiente Web en sentido general. Además el sistema debe estar disponible las 24 horas del día.

## **Rendimiento**

Los tiempos de respuestas deben ser generalmente rápidos al igual que la velocidad de procesamiento de la información.

## **Seguridad.**

El sistema debe contar con protección contra acciones no autorizadas o que puedan afectar la integridad de los datos que en él se manejan.

## **Confiabilidad.**

El sistema contará con la seguridad requerida para mantener la integridad de los datos y estará disponible para su acceso en cualquier momento que se requiera manteniendo así su disponibilidad, siendo así confiable para su uso.

## **Software**

Se requiere disponer para el funcionamiento del sistema de un servidor que cuente con Sistema Operativo Windows XP o versiones superiores, y servidores Apache 2.0 o una versión superior y como gestor de bases de datos PostgreSQL 8.1 o una versión superior. Los usuarios del sistema deberán contar con un navegador capaz de interpretar código JavaScript, como: Internet Explorer 5.5 o superior, para poder acceder a las opciones que brinda el sistema.

## **Hardware**

Para el desarrollo y ejecución de la aplicación propuesta se necesita como mínimo:

Para el servidor de aplicación:

- ✍ Pentium a 333 MHz o superior
- ✍ 512 MB RAM o superior
- ✍ 10 GB de espacio libre en Disco Duro

Para el client page:

- ✍ Pentium a 233 MHz o superior.

- ✍ 64 MB RAM o superior.
- ✍ Tarjeta de red (MODEM o red con TCP-IP para conexión al servidor).

Además es necesario contar con una impresora para poder imprimir los diferentes tipos de reportes.

### 2.3 Definición de Actores del sistema

Los actores no son ninguna parte del sistema, ellos representan a cualquiera o algo que debe interactuar con el sistema. Un actor puede que:

- ✍ Sólo brinde información de entrada al sistema.
- ✍ Sólo reciba la información del sistema.
- ✍ Brinde y reciba información.

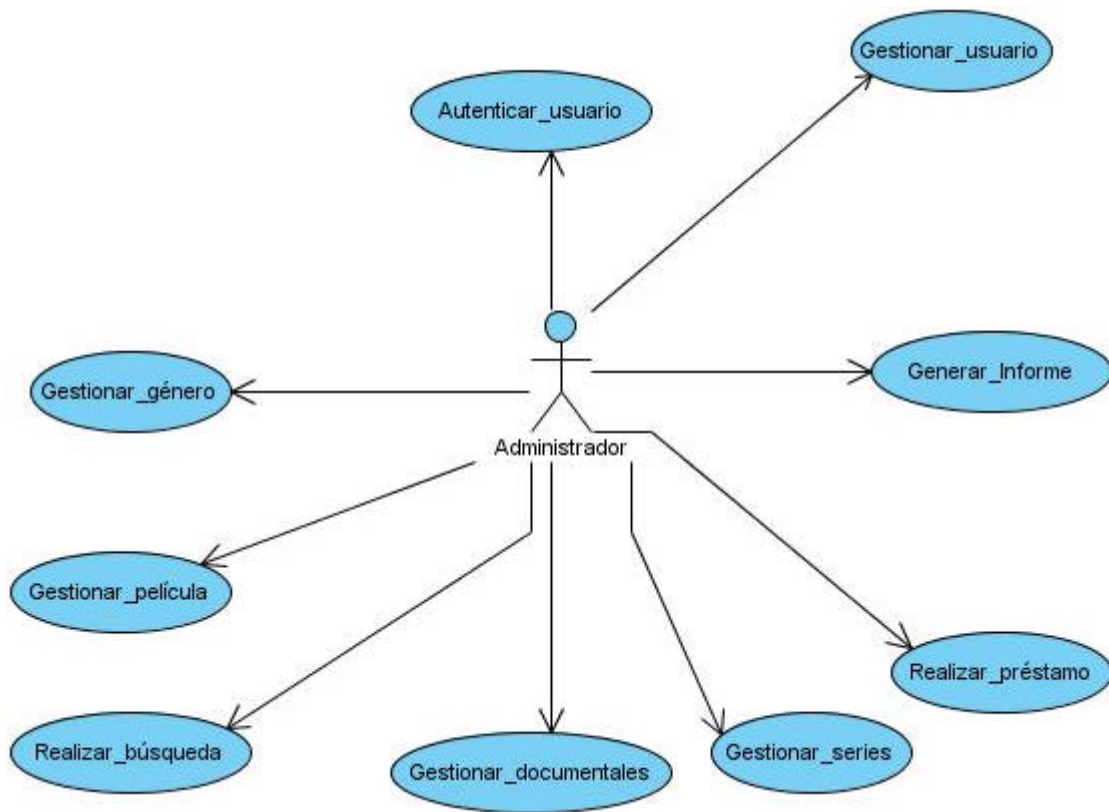
ACTORES DEL SISTEMA	JUSTIFICACIÓN
<b>Administrador</b>	Gestiona todo los medios audiovisuales que controla el sistema, además tiene el privilegio de realizar préstamos.

### Casos de Uso del Sistema

CASOS DE USO DEL SISTEMA	JUSTIFICACIÓN
<b>Gestionar Usuario</b>	Necesario porque permite al Administrador Insertar, Modificar y Eliminar Usuario.

<b>Gestionar Género</b>	Permite Insertar, Modificar y Eliminar los distintos géneros existentes.
<b>Gestionar Película</b>	Permite al sistema Insertar, Modificar y Eliminar las películas en el Fondo de Medios.
<b>Gestionar Serie</b>	Permite al sistema Insertar, Modificar y Eliminar las series en el Fondo de Medios.
<b>Gestionar Préstamo</b>	Permite al sistema Insertar, Modificar y Eliminar los distintos préstamos que se llevaran a cabo en el Fondo de Medios.
<b>Generar Reporte</b>	Permite al sistema generar un reporte en busca de cierta información relacionada con los medios audiovisuales.
<b>Realizar Búsquedas</b>	Nos brinda la posibilidad de realizar distintas búsquedas teniendo en cuenta los diferentes criterios para la misma.
<b>Autenticar Usuario</b>	Permite mantener la seguridad en el sistema.

## Diagrama de Caso de Uso del Sistema



**Figura # 1 Diagrama de Casos de Uso**

**Descripciones de los Casos de Uso del Sistema.**

La descripción de los casos de uso del sistema, detalla las acciones que tienen lugar durante la interacción actor-sistema, es decir, describe el flujo de actividades que realiza el actor al hacer uso del sistema y las correspondientes respuestas del mismo.

**Tabla # 1 Descripción del Caso de Uso Autenticar Usuario.**

<b>Caso de Uso:</b>	<b>Autenticar usuario</b>
<b>Actores:</b>	<b>Administrador (Inicia el CU)</b>

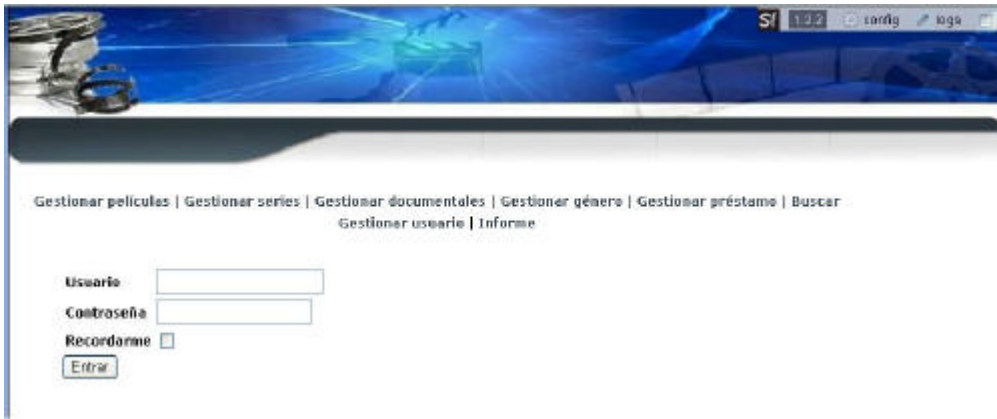
<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador introduce los datos para autenticarse en el sistema. El sistema verifica que todos los datos necesarios hayan sido insertados y de forma correcta, dando acceso al sistema según nivel de acceso, finalizando el caso de uso.
<b>Precondiciones:</b>	-
<b>Referencias:</b>	R.1
<b>Prioridad:</b>	<b>Crítico</b>

### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El caso de uso inicia cuando el administrador inserta los datos para la autenticación (nombre de usuario, contraseña)	2. El sistema muestra el formulario para autenticarse con los campos a llenar para entrar a la aplicación: ? Usuario ? Contraseña
3. El usuario llena los campos	4. El sistema verifica que estos campos estén llenos.
	5. El sistema verifica que este usuario se encuentre en la Base de Datos del sistema.
	6. El sistema verifica la contraseña en la base de datos del sistema.
	7. El sistema autentica al usuario en la aplicación, finalizando el caso de uso.

### Prototipo de interfaz





### Flujos Alternos

Acción del Actor	Respuesta del Sistema
	4.1 El sistema informa que estos campos son requeridos para entrar a la aplicación por lo que se deben llenar y va a la actividad 1 de la sección “signin”.

### Flujos Alternos

Acción del Actor	Respuesta del Sistema
	5.1 El sistema informa que estos campos no tienen una información válida y muestra el mensaje de error “Usuario o contraseña incorrecta” y va a la actividad 1 de la Sección “signin”.

### Flujos Alternos

Acción del Actor	Respuesta del Sistema
<b>Postcondiciones</b>	El administrador queda autenticado con acceso a todos los medios audiovisuales del Fondo de Largometrajes que gestiona el sistema.

Tabla # 2 Descripción del Caso de Uso Gestionar Usuario.

<b>Caso de Uso:</b>	<b>Gestionar Usuarios.</b>
<b>Actores:</b>	<b>Administrador(Inicia el CU)</b>
<b>Resumen:</b>	<p>El CU se inicia cuando el administrador indica realizar alguna de las operaciones siguientes:</p> <p><b>Insertar Usuario:</b> cuando el administrador inserta un nuevo usuario en el sistema, para esto llena los datos necesarios, finalizando el caso de uso.</p> <p><b>Modificar Usuario:</b> cuando el administrador necesita modificar datos de un usuario existente, finalizando el caso de uso.</p> <p><b>Eliminar Usuario:</b> el administrador elimina un usuario del sistema, finalizando el caso de uso.</p>
<b>Precondiciones:</b>	<p>El usuario autenticado debe ser administrador del sistema.</p> <p>Deben existir usuarios insertados siempre que se desee listar, eliminar, modificar usuarios del sistema.</p>
<b>Referencias:</b>	RF.2, RF.2.1, RF.2.2, RF.2.3
<b>Prioridad:</b>	<b>Crítico</b>

#### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> <li>1. El Administrador del sistema accede a la pantalla de Gestionar Usuario.</li> <li>2. Selecciona una de las opciones que desea realizar</li> </ol>	<p>Muestras los usuarios insertados en el sistema.</p>

## Sección “Insertar Usuario”

Acción del Actor	Respuesta del Sistema
1. El Administrador selecciona la opción nuevo usuario.	2. El sistema muestra la interfaz Nuevo Usuario con los campos a llenar del usuario:
3.El administrador llena los campos	4. El sistema verifica que la información en todos los campos sea válida.
	5. El sistema verifica que todos los campos obligatorios estén llenos.
	6. Adiciona el usuario y se muestra un mensaje informando que el usuario se creó satisfactoriamente, finalizando el caso de uso.

### Prototipo de Interfaz

**Nuevo usuario**

Usuario

Clave

Repita la clave

---

**Permisos a grupos**

Activado

Administrador

### Flujos Alternos Insertar Usuario

Acción del Actor	Respuesta del Sistema
------------------	-----------------------

4.1. El sistema muestra un mensaje indicando que el usuario no fue creado debido a algunos errores y va a la actividad 3 de la Sección “Insertar Usuario”

**Flujos Alternos**

Acción del Actor	Respuesta del Sistema
	5.1. El sistema informa que se deben llenar los campos obligatorios que están vacíos y va a la actividad 3 de la Sección “Insertar Usuario”

**Sección “Modificar Usuario”**

Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción Modificar Usuario.	2. El sistema muestra la interfaz con los campos correspondientes.
El administrador modifica los campos requeridos	3. El sistema verifica que la información en todos los campos sea válida.
	4. El sistema verifica que todos los campos obligatorios estén llenos.
	5. Modifica los datos del usuario y muestra un mensaje informando que el usuario se modificó satisfactoriamente, finalizando el caso de uso.

**Prototipo de interfaz**

**Flujos Alternos Modificar Usuario**

Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje indicando que el usuario no fue

	modificado debido a algunos errores y va a la actividad 3 de la Sección “Modificar Usuario”
	5.1 El sistema informa que se deben llenar los campos obligatorios que están vacíos y va a la actividad 3 de la Sección “Modificar Usuario”

### Sección “Eliminar Usuario”

Acción del Actor	Respuesta del Sistema
1. El administrador selecciona la opción Eliminar.	2. El sistema muestra un aviso de confirmación.
3. El administrador acepta el aviso.	4. El sistema elimina el usuario.

### Prototipo de Interfaz

### Flujos Alternos Eliminar Usuario

Acción del Actor	Respuesta del Sistema
3.1 El administrador cancela el aviso.	4.1 El sistema no elimina al usuario y muestra los usuarios insertados en el sistema.

### Prototipo de Interfaz

**Tabla # 3 Descripción del Caso de Uso Gestionar Géneros**

<b>Caso de Uso:</b>	<b>Gestionar Género</b>	
<b>Actores:</b>	<b>Administrador</b>	
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el Administrador del sistema tiene la necesidad de realizar una de las siguientes acciones:</p> <p>Insertar Género, el administrador del sistema registra en el sistema los géneros correspondientes a los medios audiovisuales que se encuentran en el Fondo.</p> <p>Modificar Género, el administrador del sistema selecciona el Género al cual se le realizara al menos un cambio, es decir al que desea modificar, luego marca la opción guardar, el sistema actualiza los nuevos cambios y finaliza el caso de uso.</p>	
<b>Precondiciones:</b>	El usuario debe autenticarse.	
<b>Referencias</b>	RF 3, RF 3.1, RF 3.2	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1) El Administrador del sistema accede a la pantalla de Gestionar Género.	Muestras los géneros insertados en el sistema.
	3) Selecciona una de las opciones que desea realizar.	
<b>Sección "Insertar Género"</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
		1) El sistema muestra el campo para registrar un Género.
	2) El Administrador del sistema llena todos los campos y marca guardar.	3) El sistema verifica que el dato entrado es correcto.
		4) El sistema guarda el dato.
		5) El sistema informa a través de un mensaje que fue registrado

satisfactoriamente.

### Prototipo de Interfaz

# Nuevo género

Género\*

Cancelar

### Flujos Alternos Sección "Insertar Género"

Acción del Actor	Respuesta del Sistema
	3.1 Si el dato entrado no es correcto, el sistema muestra un mensaje de error "campo no valido".

### Sección "Modificar Género"

Acción del Actor	Respuesta del Sistema
1) Selecciona el Género a modificar.	2) El sistema muestra el campo del género a modificar.
3) Modifica los campos que desea y selecciona la opción modificar.	4) El sistema valida los datos que fueron modificados.
	5) El sistema actualiza los nuevos datos del género.
	6) El sistema da la posibilidad de realizar otra modificación.
7) Selecciona otra máquina a modificar.	8) El sistema realiza las acciones correspondientes en el paso 2 de esta sección.

### Flujos Alternos Sección "Modificar Género"

Acción del Actor	Respuesta del Sistema
	4.1) Si algún campo está vacío muestra un mensaje de error.
7.1) El administrador del sistema decide no realizar otra modificación, finaliza así el caso de uso.	

**Tabla # 4 Descripción del Caso de Uso Gestionar Película.**

<b>Caso de Uso:</b>	<b>Gestionar Película</b>	
<b>Actores:</b>	<b>Administrador</b>	
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el Administrador del sistema tiene la necesidad de realizar una de las siguientes acciones:</p> <p><b>Insertar Película</b>, el administrador del sistema registra todos los datos de cada una de las películas que se reciben y luego selecciona la opción guardar, el sistema guarda los datos y finaliza el caso de uso.</p> <p><b>Modificar Película</b>, el administrador del sistema selecciona la película a la cual se le realizara al menos un cambio, es decir la que desea modificar, luego marca la opción guardar, el sistema actualiza los nuevos cambios y finaliza el caso de uso.</p> <p><b>Eliminar Película</b>, el administrador del sistema selecciona la película que se desea eliminar y marca la opción eliminar, el sistema elimina la película seleccionada y finaliza el caso de uso.</p>	
<b>Precondiciones:</b>	<b>El usuario debe autenticarse.</b>	
<b>Referencias</b>	<b>RF 4, RF 4.1, RF 4.2, RF 4.3</b>	
<b>Prioridad</b>	<b>Crítico</b>	
<b>Flujo Normal de Eventos</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<b>2) El Administrador del sistema accede a la pantalla de Gestionar Película.</b>	<b>Muestra las películas insertadas en el sistema.</b>



Sección "Insertar Película"	
Acción del Actor	Respuesta del Sistema
	1) El sistema muestra los campos para registrar una película.
2) El Administrador del sistema llena todos los campos y marca guardar.	3) El sistema verifica que los datos entrados sean correctos.
	4) El sistema guarda los datos.
	5) El sistema informa a través de un mensaje que fue registrado satisfactoriamente.

### Prototipo de Interfaz

**Nueva película**

Título\*

Género\*

País\*

Año de estreno\*

Director\*

**Protagonistas**

1º\*  2º  3º

4º  5º

Duración (min.)\*

Síntesis\*

Cant de originales

Cant de copias

Estante\*

Nivel\*

Sitio oficial

Imagen Principal\*

### Flujos Alternos Sección "Insertar Película"

Acción del Actor	Respuesta del Sistema
	3.1 Si los datos entrados no son correctos, el sistema muestra un mensaje de error "campos no validos" e indica además cuál o cuáles campos, no están en el formato correcto y mantiene los correctos en caso de existir alguno.

### Sección "Modificar Película "

Acción del Actor	Respuesta del Sistema
1) Selecciona la película a modificar.	2) El sistema muestra los campos de la película a modificar.
3) Modifica los campos que desea y selecciona la opción modificar.	4) El sistema valida los datos que fueron modificados.
	5) El sistema actualiza los nuevos datos de la película.
	6) El sistema da la posibilidad de realizar otra modificación.
7) Selecciona otra máquina a modificar.	8) El sistema realiza las acciones correspondientes en el paso 2 de esta sección.

### Prototipo de Interfaz

## Editar película

Título\*

Género\*

País\*

Año de estreno\*

Director\*

Protagonistas

1º\*  2º:  3º:

4º:  5º:

Duración (min.)\*

Sinopsis\*

Cant de originales

Cant de copias

Estante\*

Nivel\*

Sitio oficial

### Flujos Alternos Sección "Modificar Película"

Acción del Actor	Respuesta del Sistema
	4.1) Si algún campo está vacío muestra un mensaje de error.
8.1) El administrador del sistema decide no realizar otra modificación, finaliza así el caso de uso.	

### Sección "Eliminar Película"

Acción del Actor	Respuesta del Sistema
1) El administrador del sistema selecciona la película a eliminar	2) El sistema elimina la película y muestra el mensaje “Fue eliminada satisfactoriamente”.
	3) El sistema brinda la posibilidad de eliminar otras películas.
4) El Administrador del sistema decide realizar otra eliminación. Realiza las acciones descritas en el paso 1 de la Sección.	
Flujos Alternos Sección “Eliminar Película”	
Acción del Actor	Respuesta del Sistema
4.1) No desea realizar otra eliminación, finaliza así el caso de uso.	

Tabla # 5 Descripción del Caso de Uso Gestionar Serie.

<b>Caso de Uso:</b>	<b>Gestionar Serie</b>
<b>Actores:</b>	<b>Administrador del sistema.</b>
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el Administrador del sistema tiene la necesidad de realizar una de las siguientes acciones:</p> <p><b>Insertar Serie</b>, el administrador del sistema registra todos los datos de cada una de las series que se reciben y luego selecciona la opción guardar, el sistema guarda los datos y finaliza el caso de uso.</p> <p><b>Modificar Serie</b>, el administrador del sistema selecciona la serie a la cual se le realizara al menos un cambio, es decir la que desea modificar, luego marca la opción guardar, el sistema actualiza los nuevos cambios y finaliza el caso de uso.</p> <p><b>Eliminar serie</b>, el administrador del sistema selecciona la serie que se desea eliminar y marca la opción eliminar, el sistema elimina la serie seleccionada y finaliza el caso de uso.</p>

<b>Precondiciones:</b>	<b>El usuario debe autenticarse.</b>
<b>Referencias</b>	<b>RF 5, RF 5.1, RF 5.2, RF 5.3</b>
<b>Prioridad</b>	<b>Crítico</b>
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>3) El Administrador del sistema accede a la pantalla de Gestionar Serie.</b>	<b>Muestra las series insertadas en el sistema.</b>
<b>4) Selecciona una de las opciones que desea realizar.</b>	
<b>Sección "Insertar Serie"</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	<b>1) El sistema muestra los campos para registrar una serie.</b>
<b>2) El Administrador del sistema llena todos los campos y marca guardar.</b>	<b>3) El sistema verifica que los datos entrados sean correctos.</b>
	<b>4) El sistema guarda los datos.</b>
	<b>5) El sistema informa a través de un mensaje que fue registrado satisfactoriamente.</b>
<b>Prototipo de Interfaz</b>	

## Nueva serie

Título\*   
 Género\*   
 País\*   
 Año de estreno\*   
 Director\*   
 Protagonistas  
 1º:  2º:  3º:   
 4º:  5º:   
 Temporadas\*   
 Sinopsis\*   
 Estante\*   
 Nivel\*   
 Sitio oficial   
 Imagen Principal\*    
 En préstamo

### Flujos Alternos Sección "Insertar Serie"

Acción del Actor	Respuesta del Sistema
	3.1 Si los datos entrados no son correctos, el sistema muestra un mensaje de error "campos no validos" e indica además cuál o cuáles campos, no están en el formato correcto y mantiene los correctos en caso de existir alguno.

### Sección "Modificar Serie "

Acción del Actor	Respuesta del Sistema
1) Selecciona la serie a modificar.	2) El sistema muestra los campos de la serie a modificar.
3) Modifica los campos que desea y selecciona la opción modificar.	4) El sistema valida los datos que fueron modificados.
	5) El sistema actualiza los nuevos datos de la película.
	6) El sistema da la posibilidad de realizar otra modificación.
7) Selecciona otra serie a modificar.	8) El sistema realiza las acciones correspondientes en el paso 2 de esta sección.

## Prototipo de Interfaz

### Editar serie

Título\*

Género\*

País\*

Año de estreno\*

Director\*

Protagonistas

1º:  2º:  3º:

4º:  5º:

Temporadas\*

Sinopsis\*

Estante\*

Nivel\*

Sitio oficial

#### Flujos Alternos Sección “Modificar Serie”

Acción del Actor	Respuesta del Sistema
	4.1) Si algún campo está vacío muestra un mensaje de error.
8.1) El administrador del sistema decide no realizar otra modificación, finaliza así el caso de uso.	

#### Sección “Eliminar Serie”

Acción del Actor	Respuesta del Sistema
1) El administrador del sistema selecciona la Serie a eliminar	2) El sistema elimina la Serie y muestra el mensaje “Fue eliminada satisfactoriamente”.
	3) El sistema brinda la posibilidad de eliminar otras series.
4) El Administrador del sistema decide realizar otra eliminación. Realiza las acciones descritas en el paso 1 de la Sección.	

#### Flujos Alternos Sección “Eliminar Serie”

Acción del Actor	Respuesta del Sistema
4.1) No desea realizar otra eliminación, finaliza	

así el caso de uso.	
---------------------	--

**Tabla # 6 Descripción del Caso de Uso Gestionar Préstamo.**

<b>Caso de Uso:</b>	<b>Gestionar Préstamo</b>
<b>Actores:</b>	<b>Administrador</b>
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el Administrador del sistema tiene la necesidad de realizar una de las siguientes acciones:</p> <p><b>Insertar Préstamo</b>, el administrador del sistema registra todos los datos de cada una de los préstamos que se solicitan y luego selecciona la opción guardar, el sistema guarda los datos y finaliza el caso de uso.</p> <p><b>Modificar Préstamo</b>, el administrador del sistema selecciona el préstamo al cual se le realizara al menos un cambio, es decir la que desea modificar, luego marca la opción guardar, el sistema actualiza los nuevos cambios y finaliza el caso de uso.</p> <p><b>Eliminar Préstamo</b>, el administrador del sistema selecciona el préstamo que se desea eliminar y marca la opción eliminar, el sistema elimina el préstamo seleccionado y finaliza el caso de uso.</p>
<b>Precondiciones:</b>	<b>El usuario debe autenticarse.</b>
<b>Referencias</b>	<b>RF 6, RF 6.1, RF 6.2, RF 6.3</b>
<b>Prioridad</b>	<b>Crítico</b>
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
4) El Administrador del sistema accede a la pantalla de Gestionar datos de préstamo.	Muestra los préstamos insertados en el sistema.
5) Selecciona una de las opciones que desea realizar.	

### Sección "Insertar Préstamo"

Acción del Actor	Respuesta del Sistema
	1) El sistema muestra los campos para registrar un préstamo.
2) El Administrador del sistema llena todos los campos y marca guardar.	3) El sistema verifica que los datos entrados sean correctos.
	4) El sistema guarda los datos.
	5) El sistema informa a través de un mensaje que fue registrado satisfactoriamente.

#### Prototipo de Interfaz

## Nuevo préstamo

Usuario que entrega\*   
 Persona que recibe\*   
 Carnet de identidad\*   
 Centro de trabajo\*   
 Dirección\*   
 Teléfono\*   
 Fecha de entrega\* --  
 Fecha de recogida\* --  
 Pelicula a prestar    
 Serie a prestar    
 Documental a prestar    
 Pelicula a prestar 2    
 Pelicula a prestar 3    
 Pelicula a prestar 4    
 Pelicula a prestar 5

#### Flujos Alternos Sección "Insertar Préstamo"

Acción del Actor	Respuesta del Sistema
	3.1 Si los datos entrados no son correctos, el sistema muestra un mensaje de error "campos no validos" e indica además cuál o cuáles campos, no están en el formato correcto y mantiene los correctos en caso de existir alguno.



### Sección “Modificar Préstamo”

Acción del Actor	Respuesta del Sistema
1) Selecciona el préstamo a modificar.	2) El sistema muestra los campos del préstamo a modificar.
3) Modifica los campos que desea y selecciona la opción modificar.	4) El sistema valida los datos que fueron modificados.
	5) El sistema actualiza los nuevos datos del préstamo.
	7) El sistema da la posibilidad de realizar otra modificación.
8) Selecciona otro préstamo a modificar.	9) El sistema realiza las acciones correspondientes en el paso 2 de esta sección.

#### Prototipo de Interfaz

## Editar préstamo

Usuario que entrega\*   
 Persona que recibe\*   
 Carnet de identidad\*   
 Centro de trabajo\*   
 Dirección\*   
 Teléfono\*   
 Fecha de entrega\*     
 Fecha de recogida\*     
 Pelicula a prestar    
 Serie a prestar    
 Documental a prestar    
 Pelicula a prestar 2    
 Pelicula a prestar 3    
 Pelicula a prestar 4    
 Pelicula a prestar 5

### Flujos Alternos Sección “Modificar Préstamo”

Acción del Actor	Respuesta del Sistema
	4.1) Si algún campo está vacío muestra un mensaje de error.
8.1) El administrador del sistema decide no realizar otra modificación, finaliza así el caso de uso.	

<b>Sección “Eliminar Préstamo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1) El administrador del sistema selecciona el préstamo a eliminar	2) El sistema elimina el préstamo y muestra el mensaje “Fue eliminado satisfactoriamente”.
	3) El sistema brinda la posibilidad de eliminar otros préstamos.
4) El Administrador del sistema decide realizar otra eliminación. Realiza las acciones descritas en el paso 1 de la Sección.	
<b>Flujos Alternos Sección “Eliminar Préstamo”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
4.1) No desea realizar otra eliminación, finaliza así el caso de uso.	

**Tabla # 7 Descripción del Caso de Uso Gestionar Documentales.**

<b>Caso de Uso:</b>	<b>Gestionar Documental</b>
<b>Actores:</b>	<b>Administrador</b>
<b>Resumen:</b>	<p>El caso de uso se inicia cuando el Administrador del sistema tiene la necesidad de realizar una de las siguientes acciones:</p> <p><b>Insertar Documental</b>, el administrador del sistema registra todos los datos de cada uno de los documentales que se reciben y luego selecciona la opción guardar, el sistema guarda los datos y finaliza el caso de uso.</p> <p><b>Modificar Documental</b>, el administrador del sistema selecciona el documental a la cual se le realizara al menos un cambio, es decir la que desea modificar, luego marca la opción guardar, el sistema actualiza los nuevos cambios y finaliza el caso de uso.</p> <p><b>Eliminar Documental</b>, el administrador del sistema selecciona el documental que se desea eliminar y marca la opción eliminar, el sistema elimina el documental seleccionado y finaliza el caso de uso.</p>

<b>Precondiciones:</b>	El usuario debe autenticarse.
<b>Referencias</b>	RF 7, RF 7.1, RF 7.2, RF 7.3
<b>Prioridad</b>	Crítico

### Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
5) El Administrador del sistema accede a la pantalla de Gestionar Documental.	Muestra los documentales insertados en el sistema.
6) Selecciona una de las opciones que desea realizar.	

### Sección “Insertar Documental”

Acción del Actor	Respuesta del Sistema
	1) El sistema muestra los campos para registrar un documental.
2) El Administrador del sistema llena todos los campos y marca guardar.	3) El sistema verifica que los datos entrados sean correctos.
	4) El sistema guarda los datos.
	5) El sistema informa a través de un mensaje que fue registrado satisfactoriamente.

### Prototipo de Interfaz

## Nuevo documental

Título\*   
 Género\* Acción  
 País\* Afganistán  
 Año de estreno\*   
 Duración (min.)\*   
 Estante\*   
 Nivel\*   
 En préstamo   
Cancelar Insertar

### Flujos Alternos Sección “Insertar Documental”

Acción del Actor	Respuesta del Sistema
	3.1 Si los datos entrados no son correctos, el sistema muestra un mensaje de error

	“campos no validos” e indica además cuál o cuáles campos, no están en el formato correcto y mantiene los correctos en caso de existir alguno.
--	---

**Sección “Modificar Documental”**

Acción del Actor	Respuesta del Sistema
1) Selecciona el documental a modificar.	2) El sistema muestra los campos de la documental a modificar.
3) Modifica los campos que desea y selecciona la opción modificar.	4) El sistema valida los datos que fueron modificados.
	5) El sistema actualiza los nuevos datos del documental.
	6) El sistema da la posibilidad de realizar otra modificación.
7) Selecciona otro documental a modificar.	8) El sistema realiza las acciones correspondientes en el paso 2 de esta sección.

**Prototipo de Interfaz**

## Editar documental

Título\*

Género\*

País\*

Año de estreno\*

Duración (min.)\*

Estante\*

Nivel\*

En préstamo

**Flujos Alternos Sección “Modificar Documental”**

Acción del Actor	Respuesta del Sistema
	4.1) Si algún campo está vacío muestra un mensaje de error.
8.1) El administrador del sistema decide no realizar otra modificación, finaliza así el caso de uso.	

<b>Sección “Eliminar Documental”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>1) El administrador del sistema selecciona el documental a eliminar</b>	<b>2) El sistema elimina el documental y muestra el mensaje “Fue eliminada satisfactoriamente”.</b>
	<b>3) El sistema brinda la posibilidad de eliminar otros documentales.</b>
<b>4) El Administrador del sistema decide realizar otra eliminación. Realiza las acciones descritas en el paso 1 de la Sección.</b>	
<b>Flujos Alternos Sección “Eliminar Documental”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>4.1) No desea realizar otra eliminación, finaliza así el caso de uso.</b>	

**Tabla # 8 Descripción del Caso de Uso Realizar Búsquedas.**

<b>Caso de Uso:</b>	<b>Realizar Búsquedas</b>	
<b>Actores:</b>	<b>Administrador</b>	
<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador del sistema tiene la necesidad de realizar distintos tipos de búsquedas teniendo en cuenta distintos tipos de criterios.	
<b>Precondiciones:</b>	El usuario debe autenticarse.	
<b>Referencias</b>	RF 6, RF 6.1, RF 6.2, RF 6.3	
<b>Prioridad</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1) El Administrador del sistema accede a la pantalla de Realizar Búsquedas	2) Muestra los distintos medios audiovisuales a buscar. ? Película ? Serie ? Documental	
3) Selecciona una de las opciones que desea realizar.		
<b>Sección Buscar Película</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1) El administrador escoge la opción de buscar una película.	3) El sistema muestra los criterios de búsqueda para obtener las películas correspondientes.	
2) Llena los campos que desee	4) El sistema realiza una búsqueda y muestra los resultados según los criterios de búsqueda.	
<b>Prototipo de Interfaz</b>		

**Llene al menos un campo para realizar la búsqueda**

**Buscar películas**

Título

Género

Director

Protagonista principal

Año

Estante

Nivel

En préstamo

**Premios Oscar**

<input type="checkbox"/> Mejor película	<input type="checkbox"/> Mejor director
<input type="checkbox"/> Mejor actor	<input type="checkbox"/> Mejor actriz
<input type="checkbox"/> Mejor actor de reparto	<input type="checkbox"/> Mejor actriz de reparto
<input type="checkbox"/> Mejor guión original	<input type="checkbox"/> Mejor guión adaptado
<input type="checkbox"/> Mejor edición	<input type="checkbox"/> Mejor fotografía
<input type="checkbox"/> Mejor película animada	<input type="checkbox"/> Mejor sonido
<input type="checkbox"/> Mejor canción original	<input type="checkbox"/> Mejor banda sonora
<input type="checkbox"/> Mejor película de habla no inglesa	<input type="checkbox"/> Mejores efectos visuales

**Flujos Alternos Sección Buscar Película**

Acción del Actor	Respuesta del Sistema
	<b>4.1 El sistema muestra un mensaje de error si los datos especificados por el usuario no son los requeridos.</b>

**Sección “Buscar Documental”**

Acción del Actor	Respuesta del Sistema
<b>1) El administrador escoge la opción de buscar documental.</b>	<b>3) El sistema muestra los criterios de búsqueda para obtener los documentales correspondientes.</b>
<b>2) Llena los campos que desee</b>	<b>4) El sistema realiza una búsqueda y muestra los resultados según los criterios de búsqueda.</b>

**Prototipo de Interfaz**

**Llene al menos un campo para realizar la búsqueda**

**Buscar documentales**

Título

Género

Año

Estante

Nivel

En préstamo

**Flujos Alternos Sección Buscar Documental.**

Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje de error si los datos especificados por el usuario no son los requeridos.

**Sección "Buscar Serie"**

Acción del Actor	Respuesta del Sistema
1) El administrador escoge la opción de buscar Serie	3) El sistema muestra los criterios de búsqueda para obtener las series correspondientes.
2) Llena los campos que desee	4) El sistema realiza una búsqueda y muestra los resultados según los criterios de búsqueda.

**Prototipo de Interfaz**

**Llene al menos un campo para realizar la búsqueda**

**Buscar series**

Título

Género

Director

Protagonista principal

Año

Estante

Nivel

En préstamo

**Flujos Alternos Sección "Buscar Serie"**



Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje de error si los datos especificados por el usuario no son los requeridos.

Tabla # 9 Descripción del Caso de Uso Generar Informe.

<b>Caso de Uso:</b>	<b>Generar Informe</b>
<b>Actores:</b>	<b>Administrador</b>
<b>Resumen:</b>	El caso de uso se inicia cuando el Administrador va a solicitar algún Informe. Se muestra la interfaz de cuantos medios se encuentran almacenados en la aplicación y cuantos de estos están en préstamos.
<b>Precondiciones:</b>	El usuario debe autenticarse.
<b>Referencias</b>	RF 8
<b>Prioridad</b>	Secundario
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1) El Administrador del sistema accede a la pantalla de Generar Informe para obtener información acerca de la existencia de medios en el sistema.	2) Muestra el Informe detallado de cuantos medios existen controlados por el sistema y cuantos de ellos están en préstamo.

## **Conclusiones**

Para lograr un mayor entendimiento del problema, en este capítulo se abordaron conceptos, que fueron relacionados mediante un diagrama de casos de uso del sistema. En dicho diagrama se muestran de forma general los distintos objetos existentes en el sistema Además se describieron los requerimientos que el sistema debe cumplir para su correcto y eficaz funcionamiento, analizando así las distintas funcionalidades del mismo, determinadas en 9 casos de uso, de los cuales se realizó una descripción detallada que permite un mayor entendimiento acerca de lo que debe realizar el sistema.

# CAPITULO III. ANÁLISIS Y DISEÑO DEL SISTEMA

## 3.1 Introducción

El objetivo de este capítulo es traducir los requisitos a una descripción de cómo desarrollar el sistema, se muestra el modelado visual de su desarrollo, se presenta la arquitectura, se adapta el diseño para hacerlo corresponder con el ambiente de implementación, se muestra el modelo físico de la base de datos y se realiza el modelo de despliegue. Todos estos elementos brindan una guía indispensable para la posterior implementación del sistema.

## 3.2 Modelo del Diseño

“El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Sirve de abstracción de la implementación y es utilizada como entrada fundamental de las actividades de implementación” [27]

### 3.2.1 Arquitectura.

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

### 3.2.2 Arquitectura Cliente-Servidor.

La arquitectura Cliente-Servidor se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD).

Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red. [28]

Podemos decir además que esta arquitectura necesita 3 tipos de software para su correcto funcionamiento:

- ✍ **Software de gestión de datos:** este software se encarga de la manipulación y gestión de los datos almacenados y requeridos por las diferentes aplicaciones. Normalmente este software se aloja en el Servidor.
  
- ✍ **Software de desarrollo:** este tipo de software se aloja en los Clientes y solo en aquellos que se dediquen al desarrollo de aplicaciones.
  
- ✍ **Software de interacción con los usuarios:** reside en los clientes y es la aplicación gráfica de usuario para la manipulación de datos, siempre a nivel usuario (consultas principalmente).

A parte de estos existen más aplicaciones software para el correcto funcionamiento de esta arquitectura pero ya están condicionados por el tipo de sistema operativo instalado, el tipo de red en la que se encuentra. [29]

La arquitectura es un artefacto decisivo en la calidad del software que se desarrolla. Su evaluación mitiga los diferentes riesgos asociados con el desarrollo del software, tomar acciones tempranas y valorar los atributos no funcionales sin esperar a que el software se construya. Por los elementos anteriormente mencionados se decide utilizar la arquitectura Cliente-Servidor para la creación del sistema, debido a que la misma permite una mayor organización del código, una funcionalidad más proporcionada, además de contar con una mayor fortaleza y un menor riesgo de que intrusos corrompan el sistema.

### 3.2.3 Patrones de Arquitectura

Un patrón de arquitectura de software describe un problema particular y recurrente del diseño, que surge en un contexto específico, y presenta un esquema genérico y probado de su solución. [30] Los patrones de arquitectura expresan el esquema fundamental de organización para sistemas de software. Proveen un conjunto de subsistemas predefinidos; especifican sus responsabilidades e incluyen reglas y guías para organizar las relaciones entre ellos. [31]

#### 3.2.3.1 Patrón Modelo-Vista-Controlador

El Modelo-Vista-Controlador (MVC) divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- ✍ Modelo: Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- ✍ Vista: Muestra la información al usuario. Obtiene los datos del modelo. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- ✍ Controlador: Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista. El usuario interactúa con el sistema a través de los controladores.

Las Vistas y los Controladores conforman la interfaz de usuario. Un mecanismo de propagación de cambios asegura la consistencia entre la interfaz y el modelo. La separación del modelo de los componentes vista y del controlador permite tener múltiples vistas del mismo modelo. Si el usuario cambia el modelo a través del controlador de una vista, todas las otras vistas dependientes deben reflejar los cambios. Por lo tanto, el modelo notifica a todas las vistas siempre que sus datos cambien. Las vistas, en cambio, recuperan los nuevos datos del modelo y actualizan la información que muestran al usuario [32]. Algunos de sus principales beneficios son:

#### ✍ Menor acoplamiento

- ? Desacopla las vistas de los modelos
- ? Desacopla los modelos de la forma en que se muestran e ingresan los datos

#### ✍ Mayor cohesión

- ? Cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio)
- ? Las vistas proveen mayor flexibilidad y agilidad
- ? Se puede crear múltiples vistas de un modelo
- ? Se puede crear, añadir, modificar y eliminar nuevas vistas dinámicamente
- ? Las vistas pueden anidarse
- ? Se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual
- ? Se puede sincronizar las vistas
- ? Las vistas pueden concentrarse en diferentes aspectos del modelo

#### ✍ Mayor facilidad para el desarrollo de clientes ricos en múltiples dispositivos

- ? Una vista para cada dispositivo que puede variar según sus capacidades
- ? Una vista para la Web y otra para aplicaciones de escritorio

- ✍ Más claridad de diseño
- ✍ Facilita el mantenimiento
- ✍ Mayor escalabilidad

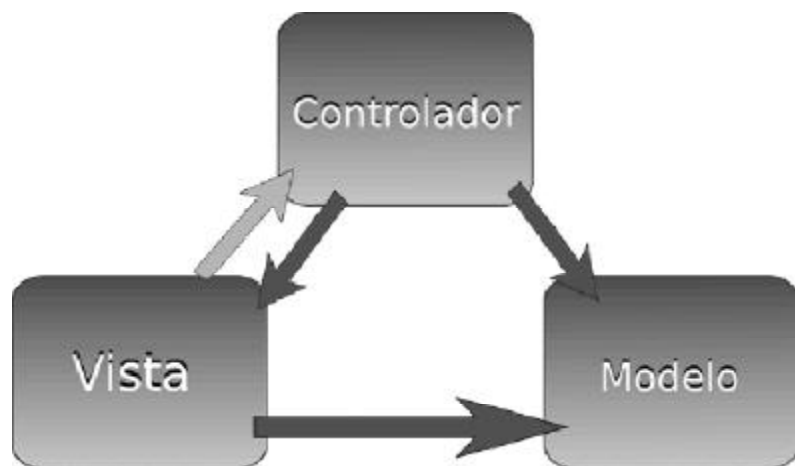


Figura # 2 Modelo Vista Controlador

### 3.2.4 Diagramas de Clases del Diseño

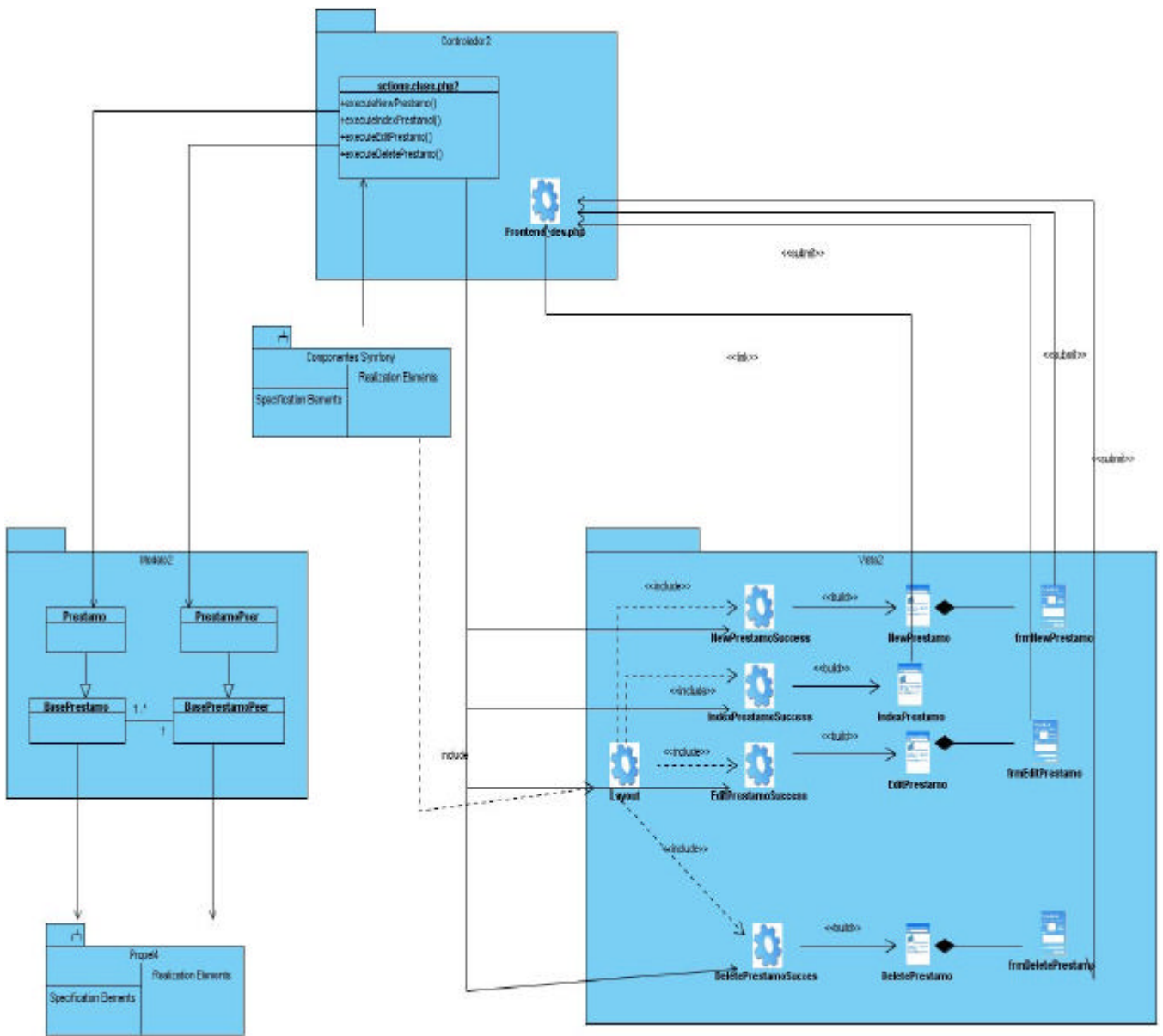
Los diagramas de clases del diseño son utilizados con el objetivo de representar las relaciones que existen entre los distintos tipos de clases. Para la realización de los siguientes diagramas se utilizó el patrón de diseño MVC ya que el funcionamiento del framework Symfony se basa en dicho modelo, lo cual permite separar la vista de la aplicación de las clases del modelo gracias al uso de una capa controladora que se encarga realizar los cambios ya sea tanto en la vista como en el modelo.

Es válido resaltar que los componentes de symfony que intervienen en la construcción de la aplicación, al ser transparentes al programador se decidió modelarlos como un subsistema que recibió el nombre de Componentes de Symfony; además de la existencia de otro componente fundamental de symfony, el cual se encarga de realizar el mapeo relacional de los objetos, el cual se denomina Propel. A continuación se mostrará una representación de los principales diagramas de clases correspondientes a varios de los casos de usos del sistema.









Figura# 5 Diagrama de Clases del Diseño Gestionar Préstamo

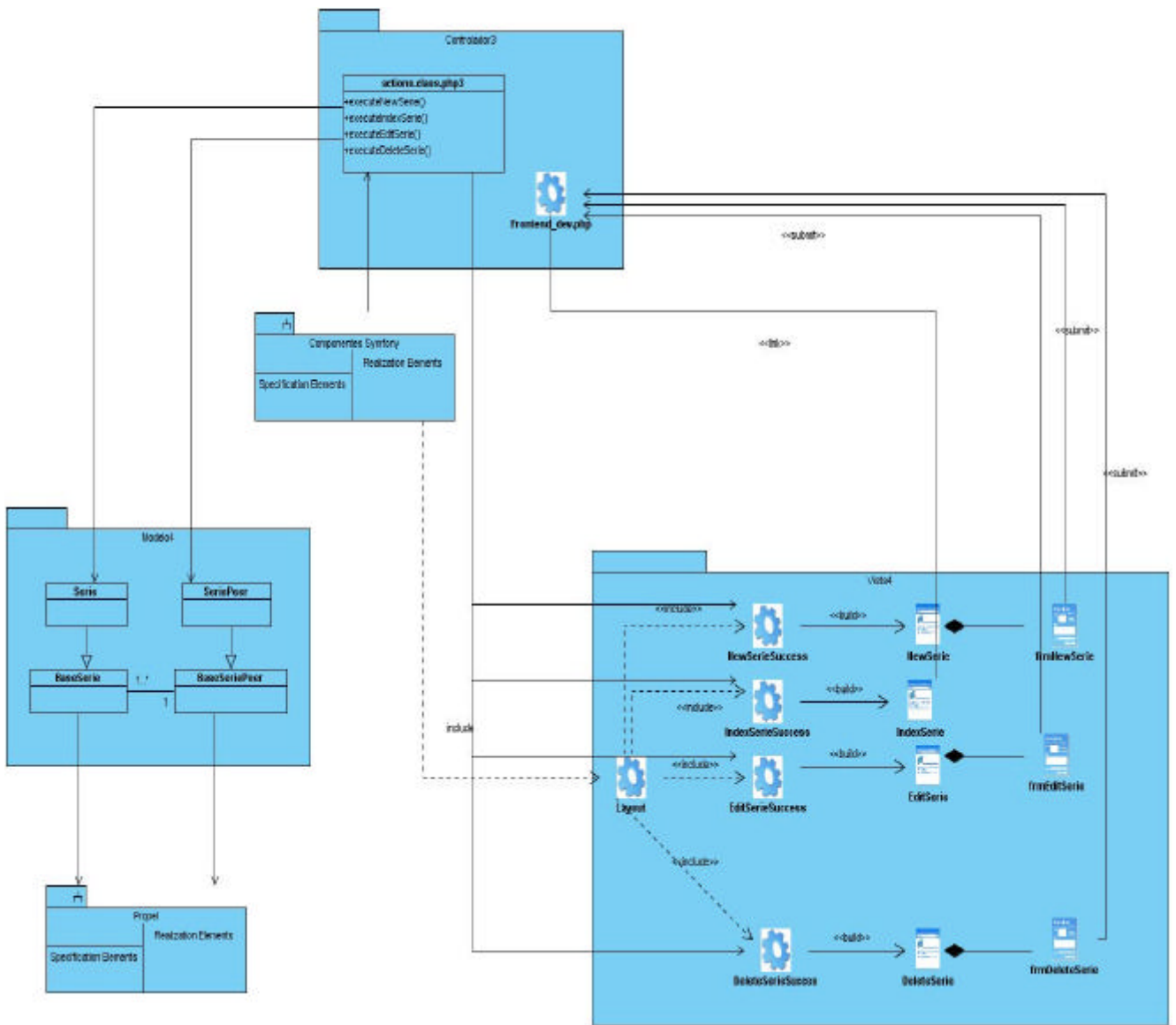


Figura # 6 Diagrama de Clases del Diseño Gestionar Serie

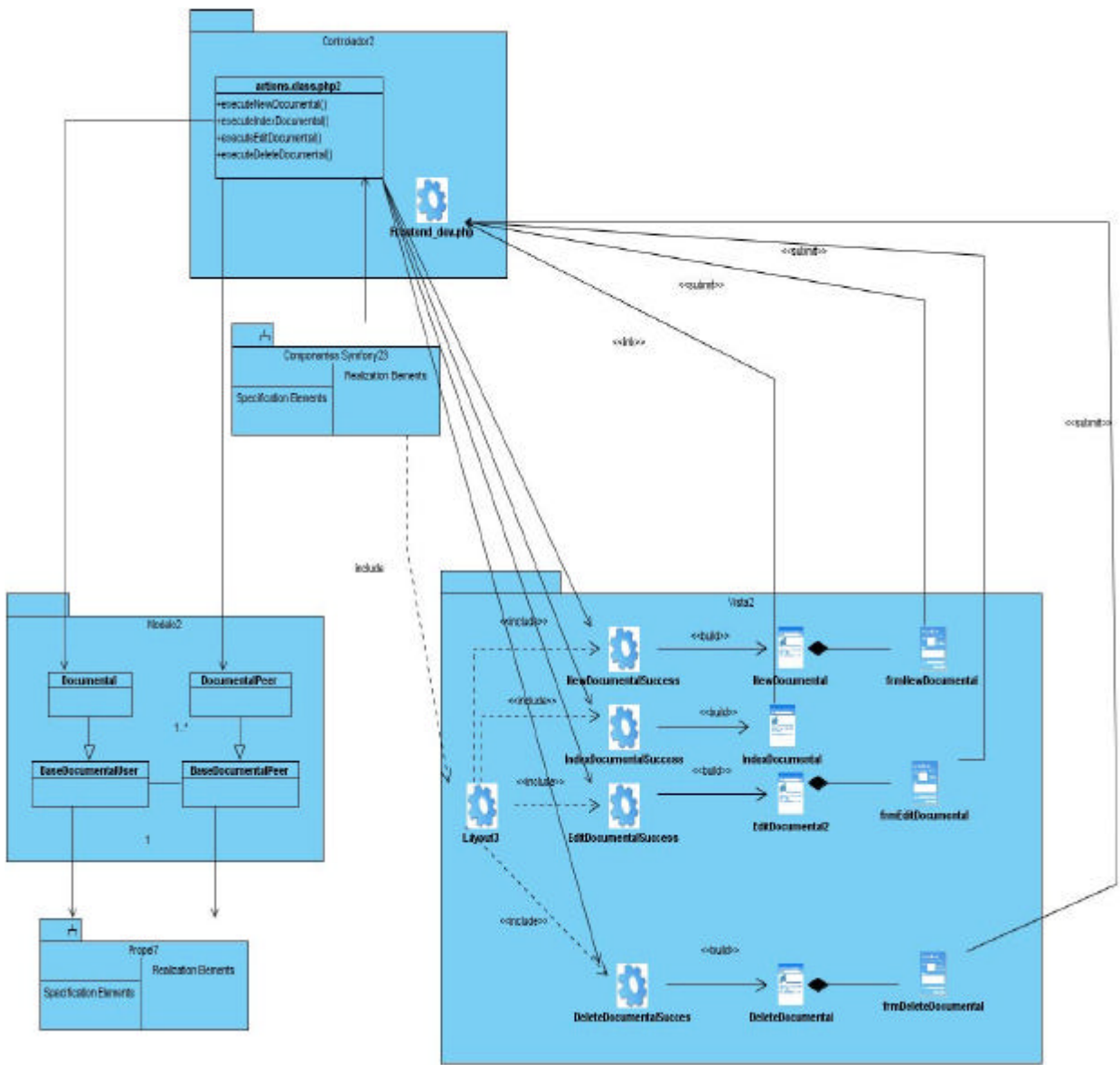


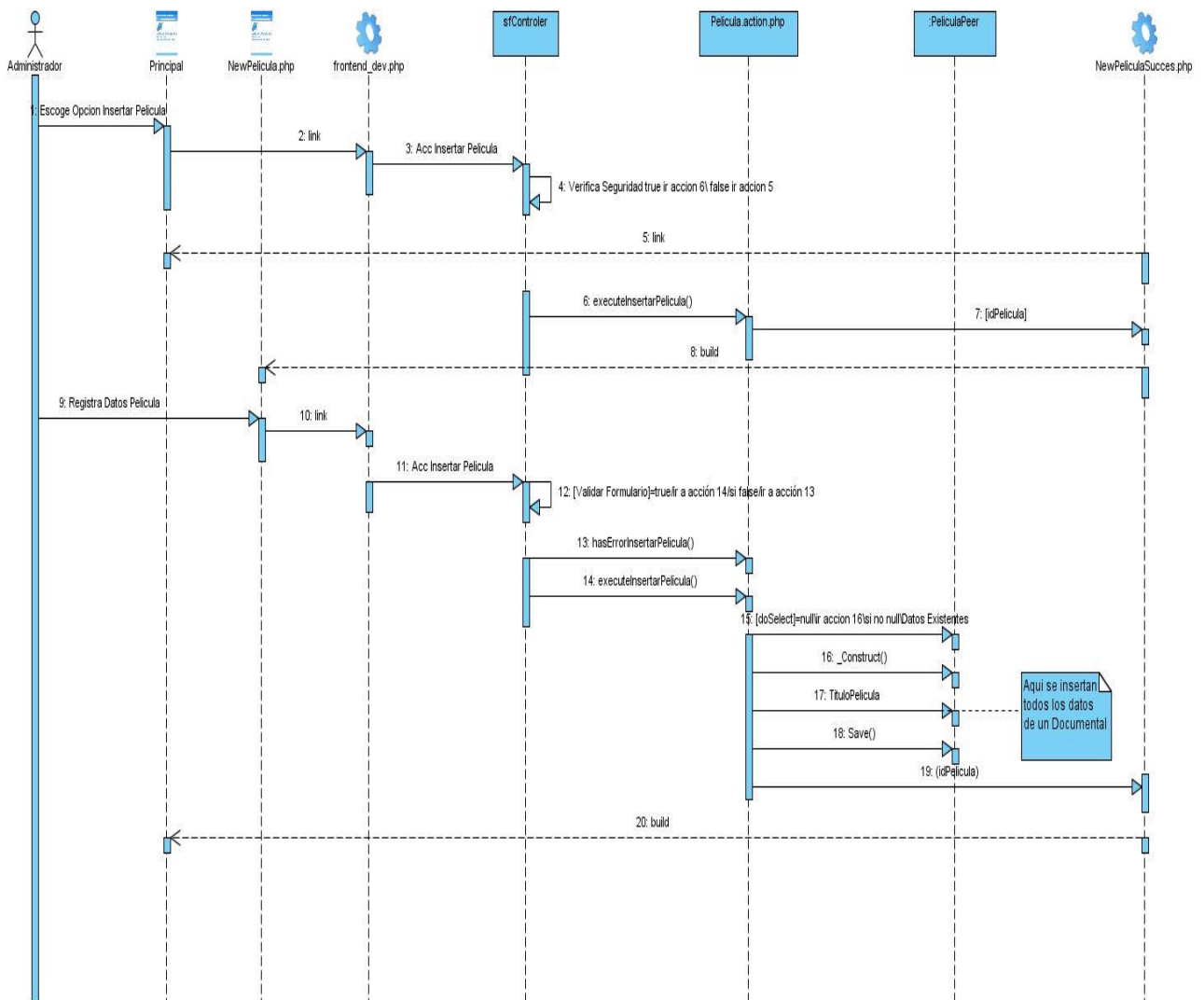
Figura # 7 Diagrama de Clases de Diseño Gestionar Documental

### **3.2.5 Diagramas de Interacción. Diagramas de Secuencia.**

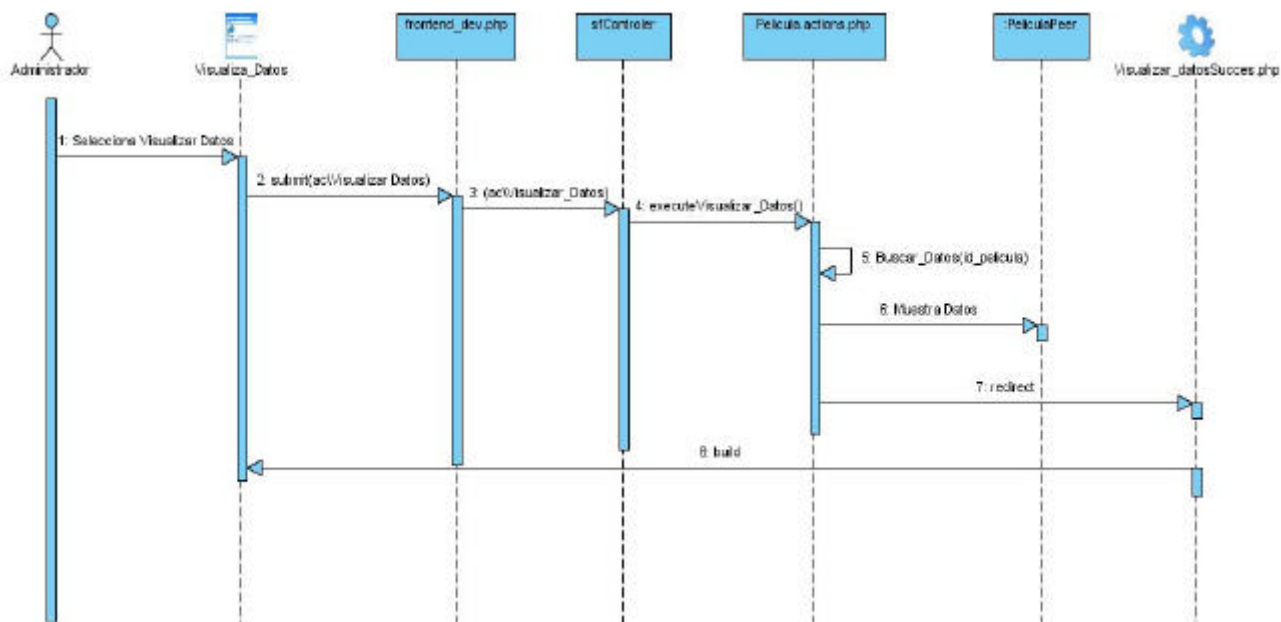
Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, y componentes, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. En el contexto de las clases describen la forma en que grupos de objetos colaboran para proveer un comportamiento. Mientras que un diagrama de casos de uso presenta una visión externa del sistema, la funcionalidad de dichos casos de uso se recoge como un flujo de eventos utilizando para ello interacciones entre sociedades de objetos.

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes pasados entre los objetos. En el diseño de este sistema se utiliza este tipo de diagrama. Debido a la enorme cantidad de escenarios que presenta el mismo y a la similitud que existe entre estos se decidió reflejar en el presente documento solo los principales escenarios.

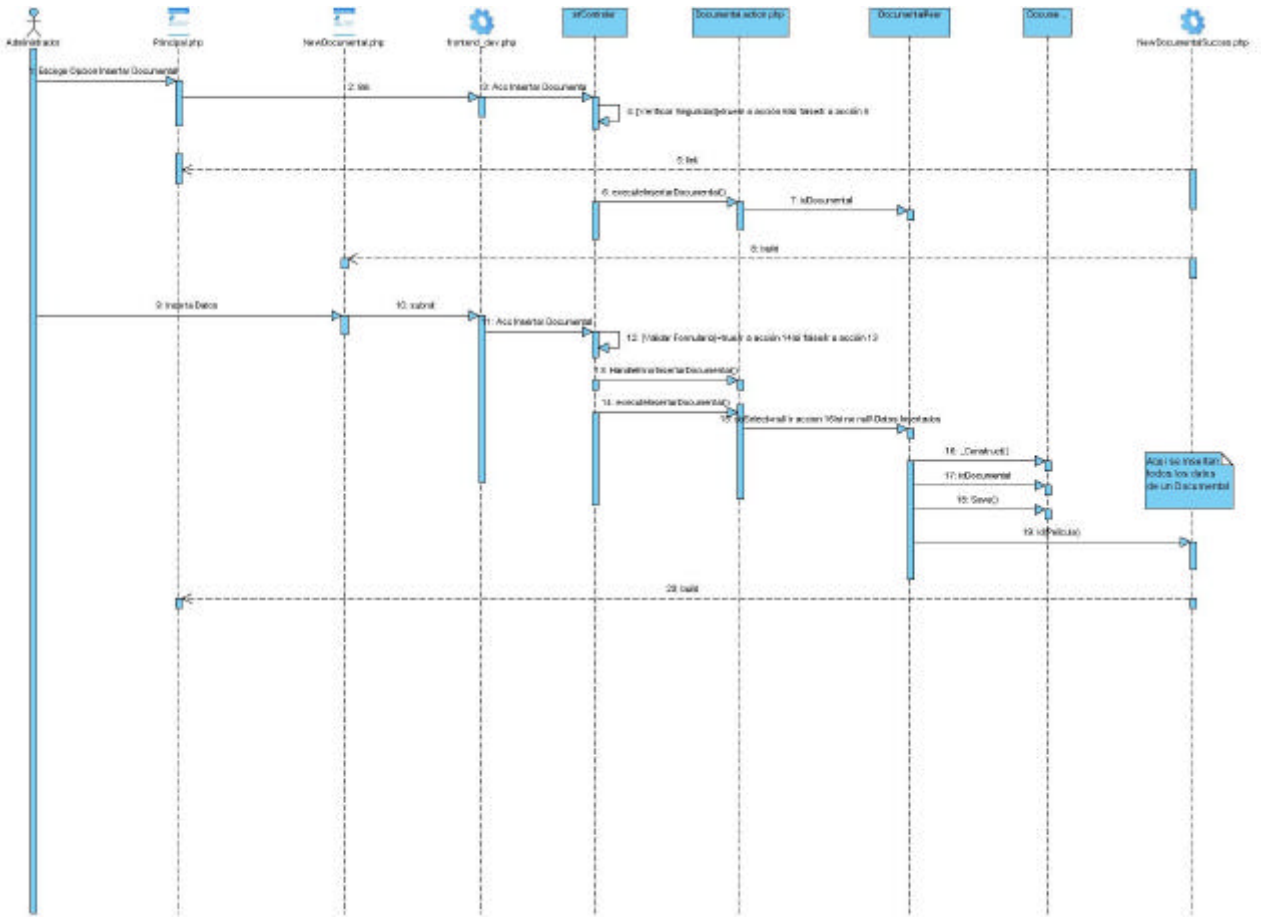
### 3.2.5.1 Diagramas de Secuencia.



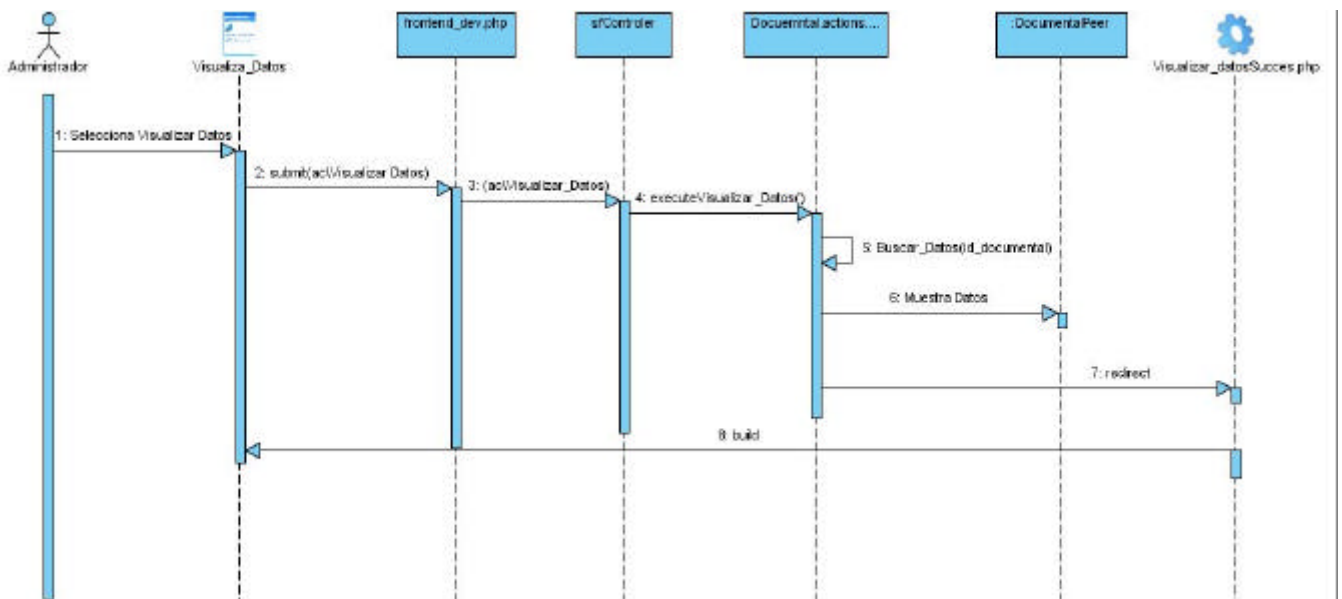
**Figura # 8 Diagrama de Secuencia Gestionar Película.  
Escenario Insertar Película.**



**Figura # 9 Diagrama de Secuencia Gestionar Película.  
Escenario Visualizar Película.**



Figura# 10 Diagrama de Secuencia Gestionar Documental  
Escenario Insertar Documental





## Figura # 11 Diagrama de Secuencia Gestionar Documental Escenario Visualizar Documental

### 3.2.6 Diagrama de Despliegue

El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones.

La mayoría de las veces el modelado de la vista de despliegue implica modelar la topología del hardware sobre el que se ejecuta el sistema. Aunque UML no es un lenguaje de especificación de hardware de propósito general, se ha diseñado para modelar muchos de los aspectos de hardware de un sistema a un nivel suficiente para que un ingeniero de software pueda especificar la plataforma sobre la que se ejecuta el software del sistema.

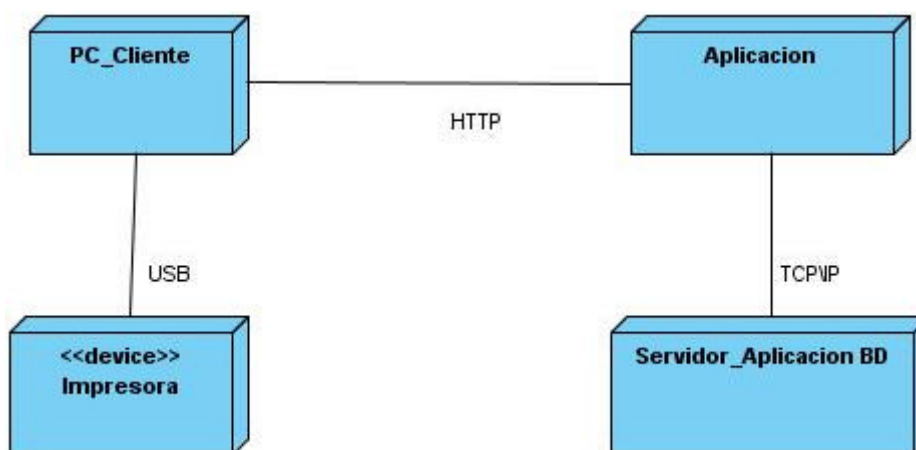


Figura # 12 Diagrama de Despliegue

**PC Cliente:** Estación de trabajo desde donde interactúa el cliente con la aplicación.

**Aplicación:** Contiene la aplicación Web.

**Servidor Aplicación BD:** Servidor donde se almacenan los datos de la aplicación.

**Impresora:** Dispositivo que permite editar los reportes generados en la aplicación.

### 3.2.7 Diagrama de Clases Persistentes.

Este diagrama muestra las relaciones que se establecen entre las tablas de la Base de Datos del Sistema.

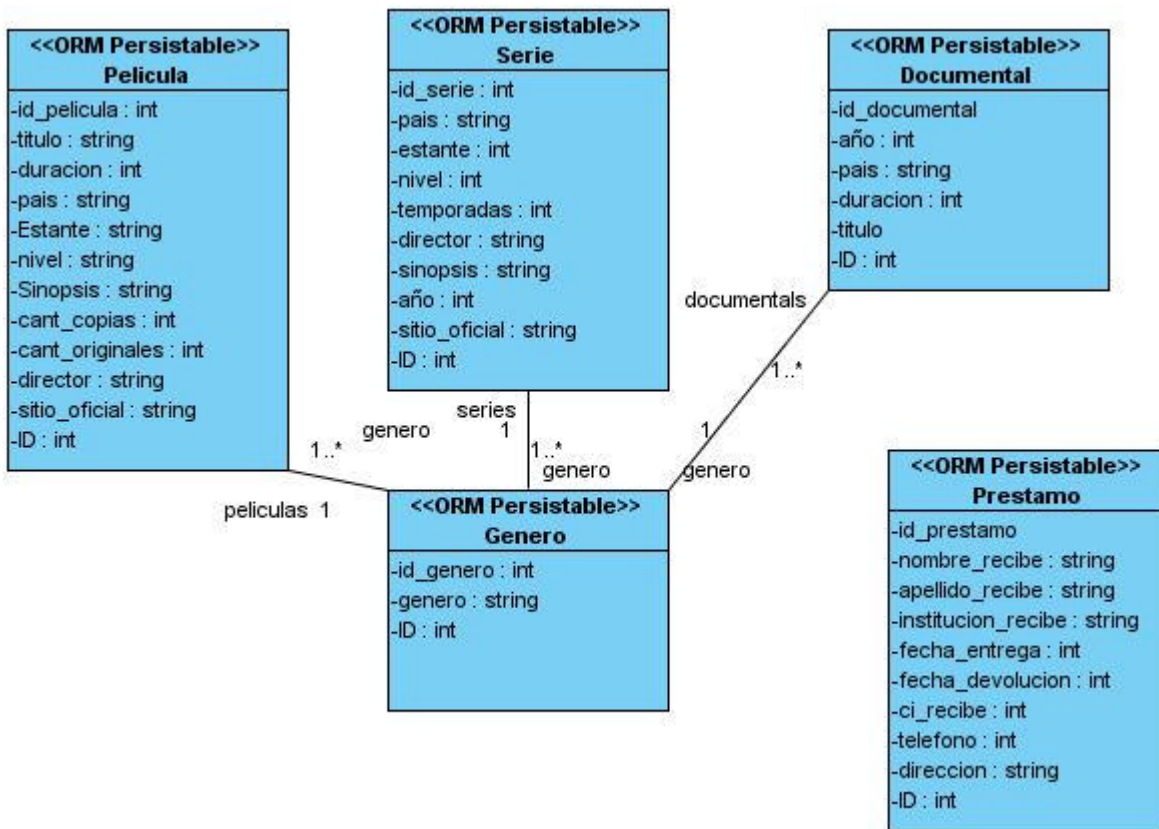


Figura # 13 Diagrama de Clases Persistentes.

### 3.2.8 Diagrama de Modelo de Datos.

Muestra las tablas y las relaciones que se generan a partir de la relación de las clases persistentes.

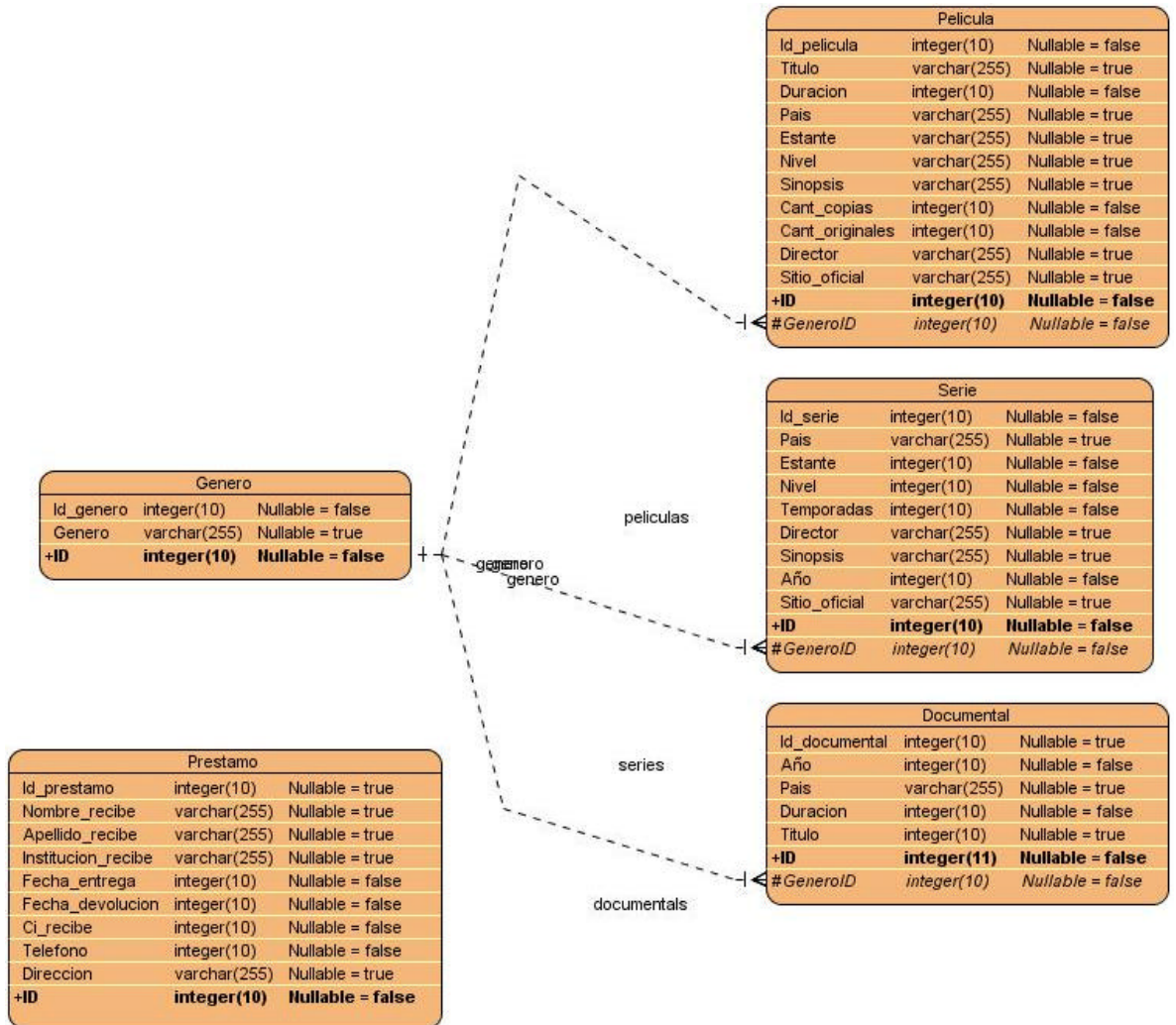


Figura # 14 Modelo de Datos.

### **3.4 Conclusiones**

En este capítulo se realizó la construcción de la solución realizando un refinamiento de los artefactos obtenidos, los diagramas de clases del diseño, así como todos los diagramas de secuencia por cada escenario de la aplicación, los cuales brindan una panorámica de como el usuario interactúa con la misma. Además se obtuvo el diagrama de clases persistentes, el modelo de datos y el diagrama de despliegue que permite interpretar las interacciones entre los distintos componentes que forman parte de la aplicación.

# CAPITULO IV. IMPLEMETACIÓN DEL SISTEMA

En el presente capítulo se desarrolla el modelo de implementación. Se muestran los diagramas de componentes, donde se detallan en componentes las diferentes clases y objetos utilizados en la implementación, para que se tenga una idea del funcionamiento de las mismas. Además se mostraran algunas pantallas de la aplicación.

## 4.1 Diagramas de Componentes

*“Un diagrama de componentes representa las dependencias entre componentes software, incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Un módulo de software se puede representar como componente. Algunos componentes existen en tiempo de compilación, algunos en tiempo de enlace y algunos en tiempo de ejecución, otros en varias de éstas”* [30]. Los diagramas de componentes fueron realizados por casos de uso, con el objetivo de facilitar la comprensión de éstos.

En los diagramas realizados, se muestran como están distribuidos los componentes según el patrón arquitectónico Modelo-vista-Controlador que utiliza Symfony como paradigma en su organización interna.

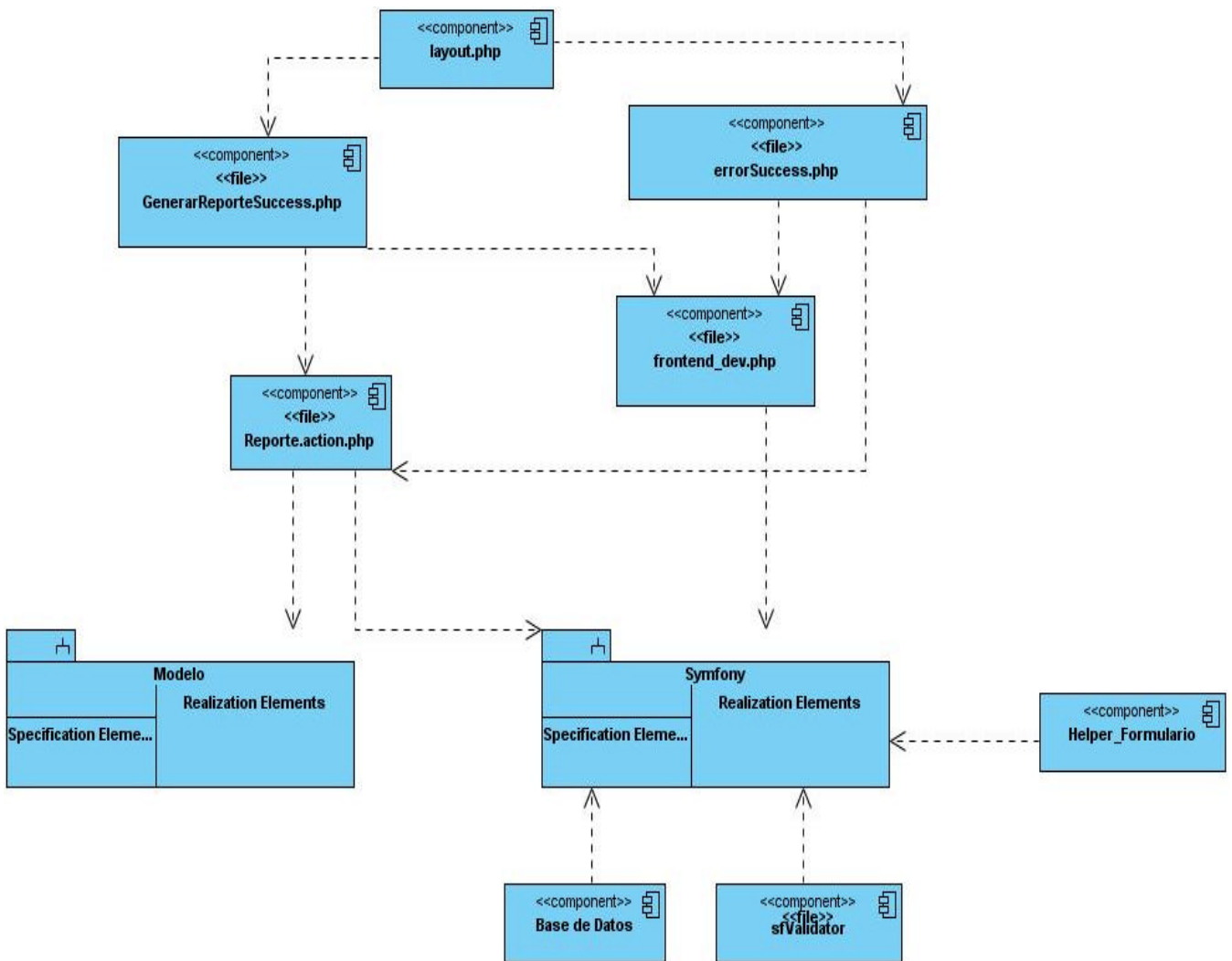
El componente `sfFrontWebController` o Controlador Frontal maneja todas las peticiones web, siendo el punto de entrada de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con la URL escrita o pinchada por el usuario. El controlador frontal se encarga de despachar las peticiones, lo que implica algo más que detectar la acción que se ejecuta. De hecho, ejecuta el código común a todas las acciones. En pocas palabras el controlador frontal es el encargado de determinar qué combinación de módulo-acción se ejecutará. El paquete Vista se encarga de producir las páginas que se muestran como resultado de las acciones que se soliciten, las cuales se integran con el layout.

En el paquete controlador se representan todas las acciones, estas `actions.php` están relacionadas con todos los archivos `Success.php` de la vista que contiene el código que liga la lógica de negocio con la presentación. Al acceder a las diferentes acciones lo primero que se hace es verificar si el usuario está autenticado y tiene los permisos correspondientes a la acción que desea realizar.

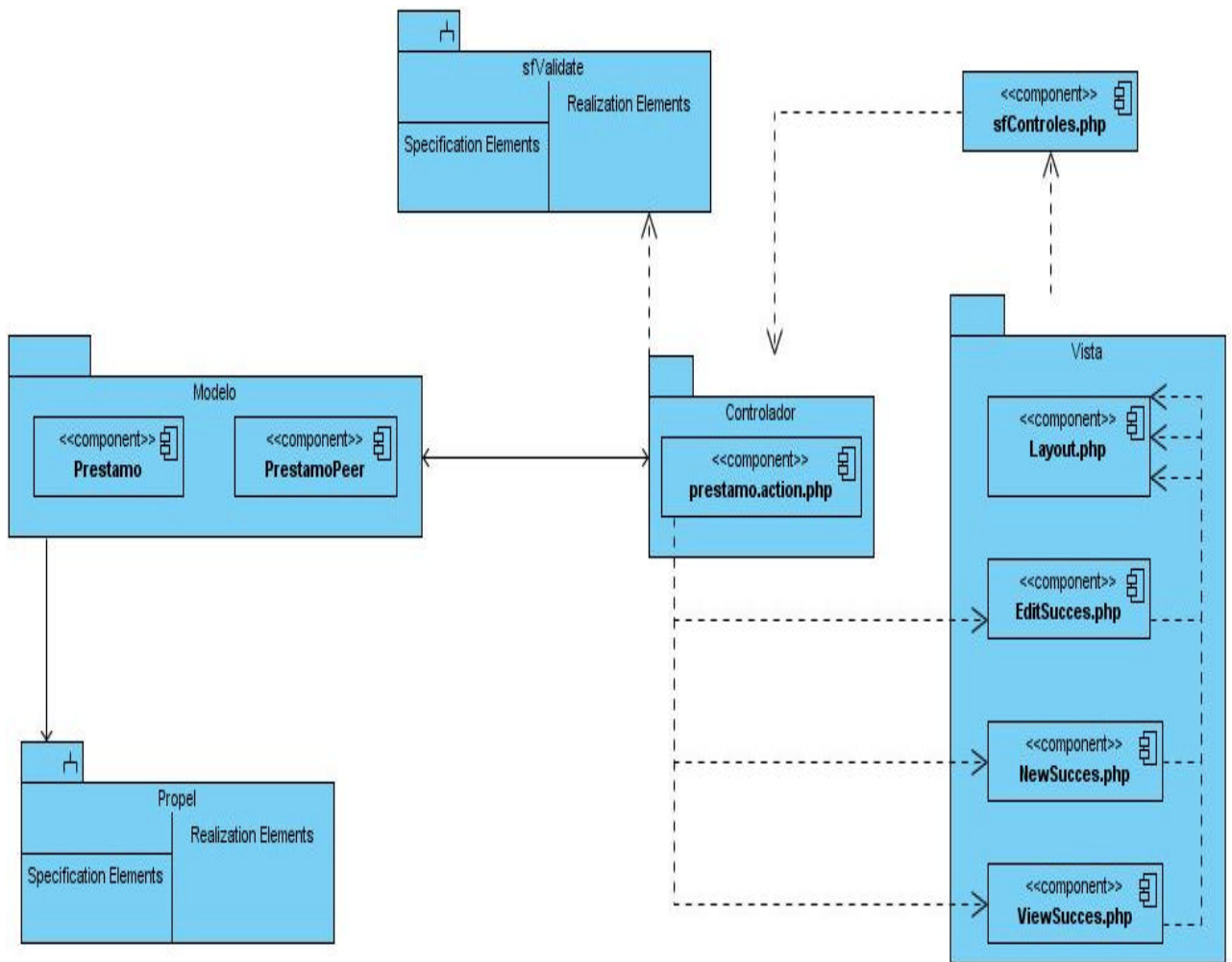
En el paquete `sfValidators` están recogidos las clases que permiten las validaciones de los formularios para cada tipo de datos. El modelo es la capa que contiene las clases: `php`, las `Peer`, las `Bases` y las `Base Peer`. Estas clases son construidas por el subsistema Propel de Symfony para el acceso a datos,

permitiendo el acceso a la base de datos mediante el mapeo de objetos a bases de datos. A continuación mostramos algunos de los diagramas de componentes de la aplicación.

### 4.1.1 Diagramas de Componentes

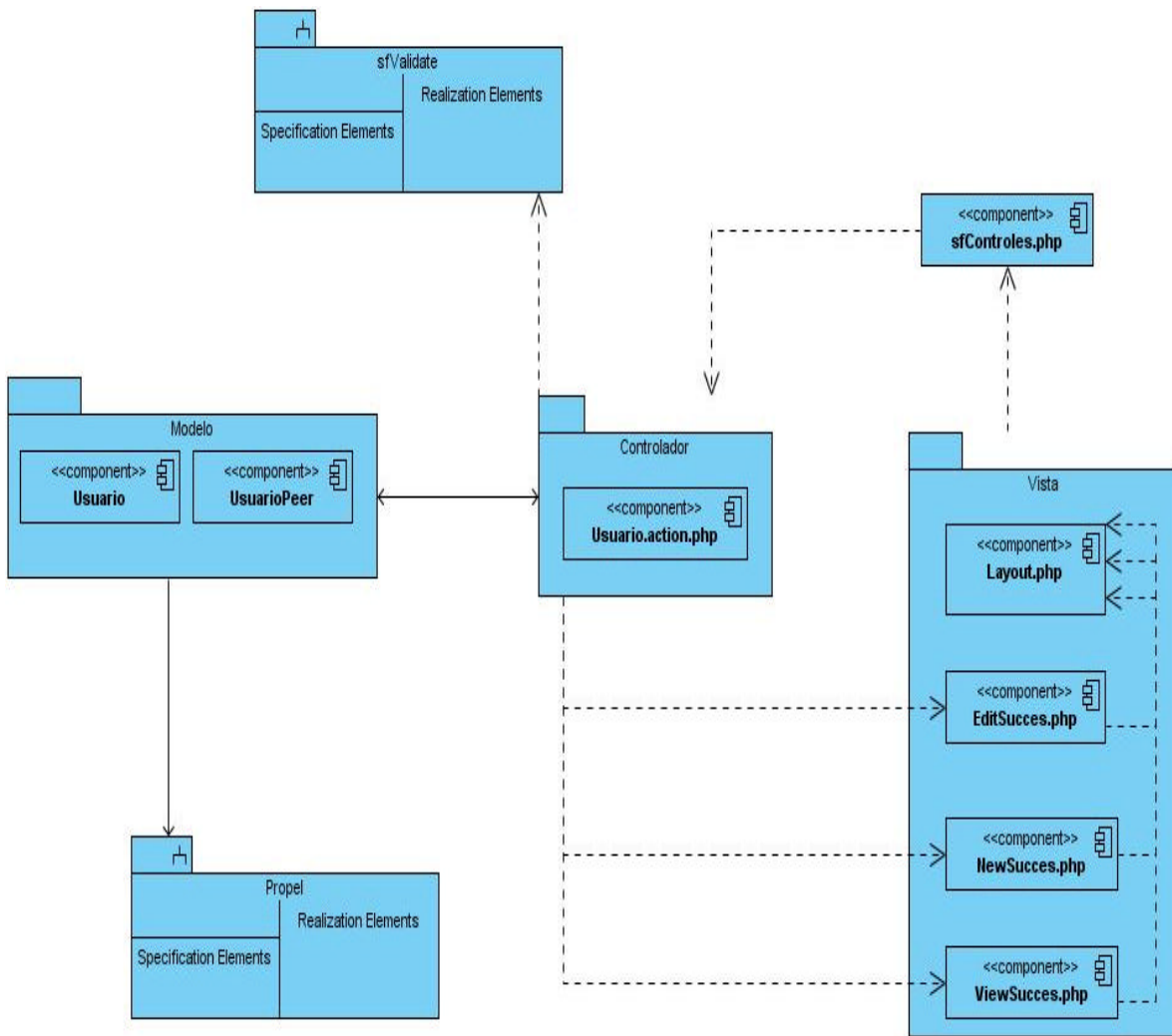


Figura# 15 Diagrama de Componentes Generar Informe

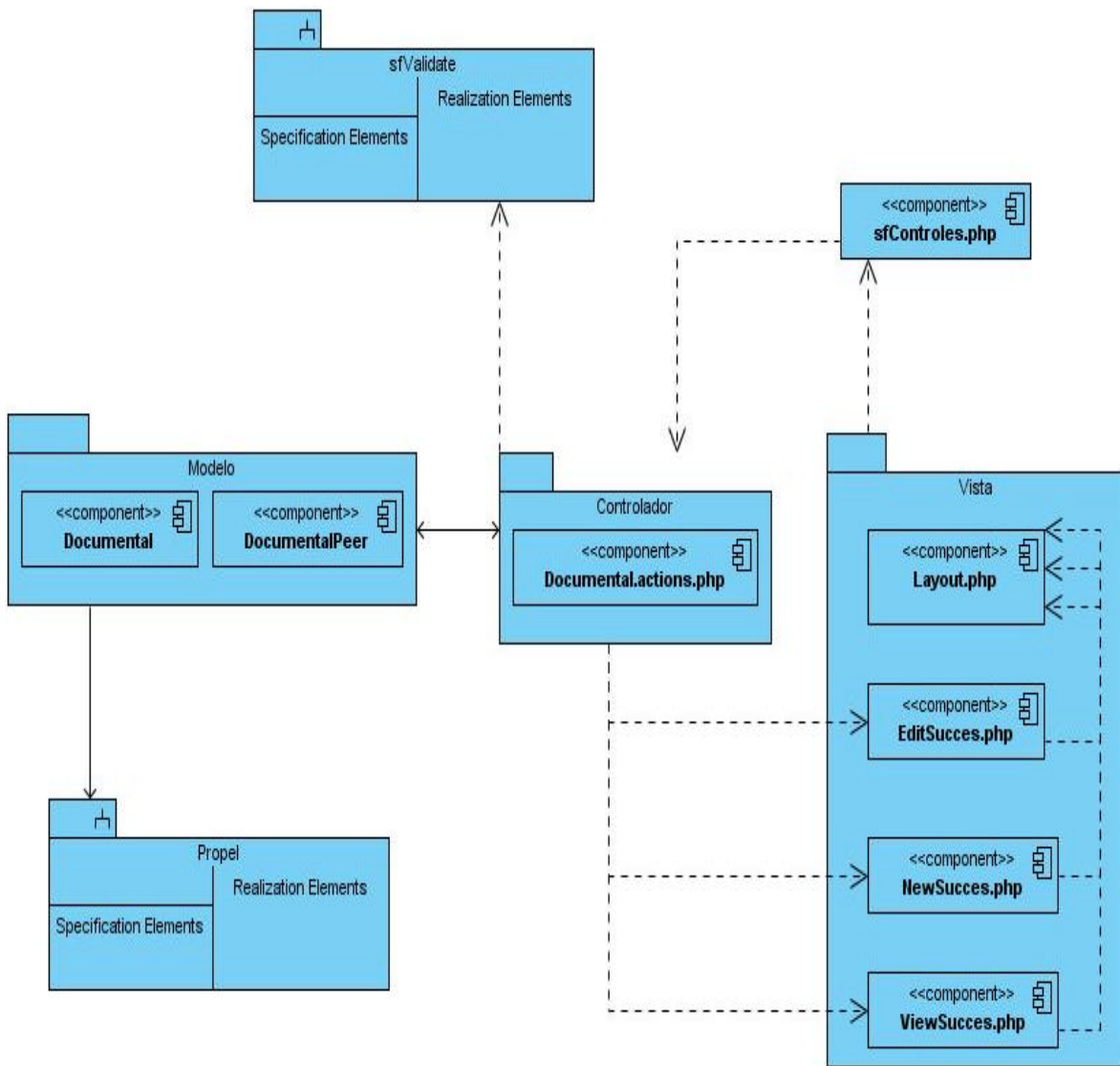


Figura# 16 Diagrama de Componentes: Gestionar Préstamo





Figura# 17 Diagrama de Componentes: Gestionar Usuario



**Figura # 18 Diagrama de Componentes Gestionar Documental.**

### **4.3 Conclusiones.**

En este capítulo se obtuvieron los diagramas de componentes, los cuales brindan una perspectiva de como se divide la aplicación en componentes y subsistemas y que funcionalidad cumple cada uno de estos, también se muestran algunas de las pantallas de la aplicación.

# CONCLUSIONES

Mediante este trabajo se ha logrado implementar un sistema que permite de manera automatizada multiplicar las posibilidades de utilización multifacético y operativa del Fondo de Materiales, mediante el cual se amplía las posibilidades de uso de estos, ofreciendo el adecuado servicio que requieren los Programas de la Batalla de Ideas que utilizan los audiovisuales y la televisión. Durante su realización se arribaron a las siguientes conclusiones:

- ✍ A través del estudio que se llevó a cabo, se lograron analizar sistemas para la gestión de la información de materiales audiovisuales.
- ✍ Se diseñó un sistema para la gestión de la información de los Materiales Audiovisuales del Fondo de la Batalla de Ideas.
- ✍ Para el desarrollo del sistema se usó el Symfony con PHP como lenguaje de programación, el Sistema Gestor de Base de Datos fue Postgre y para la modelación se utilizó UML y RUP.
- ✍ Se logró implementar un sistema para la gestión de la información del Fondo.
- ✍ Producto de la competencia entre los sistemas de gestión de la información a nivel mundial, la implantación de este sistema contribuye a la evolución y desarrollo del Fondo de Materiales de la Batalla de Ideas.
- ✍ El valor social de esta aplicación se expresa en la contribución a mejorar la gestión de la información del Fondo de Materiales de la Batalla de Ideas.

## RECOMENDACIONES

Se recomienda a las personas que van a dar continuidad a este trabajo:

- ✍ Agregar a este sistema el uso de otros servicios a medida que surjan las necesidades dentro del Fondo.
- ✍ Mejorar aun más el diseño del sistema
- ✍ Migrar en un futuro hacia otras versiones del manejador de contenido utilizado Symfony, pues incorpora nuevas características que la versión actual no posee.

# REFERENCIAS BIBLIOGRÁFICAS

[1]. Aplicaciones Cliente-Servidor, 2008.

Disponible en URL: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml> [Fecha de consulta 17 de enero de 2009]

[2] Conferencia de Ingeniería de Software Curso 2008-2009.

Disponible en: <http://www.reynox.com/sistemas/metodologia.php> [Fecha de consulta 18 de enero de 2009].

[3] Entre la XP y el RUP

Disponible en: <http://wigahluk.wordpress.com/2007/06/26/entre-la-xp-y-el-rup/> [Fecha de consulta 18 de enero de 2009].

[4] Rational Rose

Disponible en: [mayi.polanco.googlepages.com/RATIONALROSE.doc](http://mayi.polanco.googlepages.com/RATIONALROSE.doc) [Fecha de consulta 17 de enero de 2009].

[5] Dreamweaver, 2008. Disponible en URL:

<http://www.desarrolloweb.com/articulos/332.php> [Fecha de Consulta 19 de enero de 2009].

[6] Los más comunes editores web, 2008. Disponible en URL:

<http://www.diginota.com/trucos-y-tutoriales/los-mas-comunes-editores-web.html> [Fecha de Consulta 16 de enero de 2009].

[7] Los más comunes editores web, 2008. Disponible en URL:

<http://www.diginota.com/trucos-y-tutoriales/los-mas-comunes-editores-web.html> [Fecha de Consulta 16 de enero de 2009].

[8] Software Libre, 2008.

Disponible en URL: <http://www.gnu.org/philosophy/free-sw.es.html> [Fecha de consulta 17 de enero de 2009]

[9] Introducción a PHP 4, Noviembre de 2000. Disponible en URL:

[http://www.bisente.com/documentos/cursophp/PHP\\_gen.html](http://www.bisente.com/documentos/cursophp/PHP_gen.html). [Fecha de consulta 18 de enero de 2009]

[10] PHP, 2008.

Disponible en URL: <http://mnm.uib.es/gallir/php/>. [Fecha de consulta 11 de enero de 2009]

[11]. Evaluando Zend Studio, 03 nov. 2003. Disponible en URL:

<http://www.maestrosdelweb.com/editorial/zendstudio/> [Fecha de Consulta 19 de enero de 2009].

**[12]**. Los mas comunes editores web, 2008. Disponible en URL:

<http://www.diginota.com/trucos-y-tutoriales/los-mas-comunes-editores-web.html> [Fecha de Consulta 16 de enero de 2009].

**[13]** Eclipse+PHP Disponible en URL: [www.desarrolloweb.com/articulos/pdt-eclipse-php.html](http://www.desarrolloweb.com/articulos/pdt-eclipse-php.html) [Fecha de consulta 19 de enero de 2009].

**[14]**. Los Framework de PHP agilizan tu trabajo,. Disponible en URL: <http://www.maestrosdelweb.com/editorial/los-frameworks-de-php-agilizan-tu-trabajo/> [Fecha de consulta 19 de enero de 2009].

**[15]** CodeIgniter: framework para PHP. Disponible en URL: <http://sentidoweb.com/2007/04/24/codeigniter-framework-para-php.php>. [Fecha de consulta 19 de enero de 2009].

**[16]** Symfony framework para PHP. Disponible en URL: [http://www.librosweb.es/symfony/capitulo1/symfony\\_en\\_pocas\\_palabras.html](http://www.librosweb.es/symfony/capitulo1/symfony_en_pocas_palabras.html) [Fecha de consulta 19 de enero de 2009].

**[17]**. Sistema Gestor de base de datos SGBD, 01 Noviembre. De 2004. Disponible en URL: [http://www.error500.net/garbagecollector/archives/categorias/bases\\_de\\_datos/sistema\\_gestor\\_de\\_base\\_de\\_datos\\_sqbd.php](http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sqbd.php) [Fecha de Consulta 19 de enero de 2009]

**[18]**. MySQL

Disponible en URL: [http://www.netpecos.org/docs/mysql\\_postgres/x57.html](http://www.netpecos.org/docs/mysql_postgres/x57.html) [Fecha de Consulta 18 de enero de 2009] Ernesto Quiñones A.

**[19]**. Introducción a PostgreSQL.

Disponible en

URL: [http://www.postgresql.org.pe/articles/introduccion\\_a\\_postgresql.pdf](http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf). [Fecha de Consulta 19 de enero de 2009] Ernesto Quiñones A.

**[20]** Ventajas de PostgreSQL, 2008. Disponible en URL:

[http://www.netpecos.org/docs/mysql\\_postgres/x15.html](http://www.netpecos.org/docs/mysql_postgres/x15.html). [Fecha de Consulta 19 de enero de 2009]

**[21]**. Introducción a la Arquitectura de Software, 2008.

Disponible en URL: <http://www.willydev.net/descargas/prev/IntroArq.pdf> [Fecha de consulta 17 de enero de 2009]

**[22]** Arquitectura cliente-servidor, 2008.

Disponible en: <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.

[Fecha de consulta 17 de enero de 2009].

**[23]** El Servidor Web Jesús Vegas 2002-03 21

Disponible en:

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html> [Fecha de consulta

18 de enero de 2009]

**[24]**. Conceptos básicos del servidor web, 2008.

[http://www.cibernetia.com/manuales/instalacion\\_servidor\\_web/1\\_conceptos\\_basicos.php](http://www.cibernetia.com/manuales/instalacion_servidor_web/1_conceptos_basicos.php)

[Fecha de consulta 19 de enero de 2009]

**[25]** El servidor web Apache Disponible en:

<http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x31.html> [Fecha de consulta 19 de

enero de 2009] Autor: Álvaro del Castillo San Félix

**[26]** Conferencia 3 Fase de Inicio Curso 2008-2009. Disponible en URL:

<http://teleformacion.uci.cu/mod/resource/view.php?id=11569> [Fecha de consulta 19 de enero

de 2009].

**[27]** Ivar Jacobson, Grady Booch, James Rumbaugh. El proceso unificado de desarrollo de software. s.l. : Pearson Education, 2000. [Fecha de consulta 20 de marzo de 2009].

**[28]**. Introducción a la Arquitectura de Software, 2008.

Disponible en URL: <http://www.willydev.net/descargas/prev/IntroArq.pdf> [Fecha de consulta 17

de marzo de 2009]

**[29]** Arquitectura cliente-servidor, 2008.

Disponible en: <http://www.desarrolloweb.com/articulos/arquitectura-cliente-servidor.html>.

[Fecha de consulta 17 de marzo de 2009].

**[30]**. Introducción a Patrones",2008. Disponible en URL:

<http://www.mcc.unam.mx/~cursos/Algoritmos/javaDC99-2/patrones.html> [Fecha de Consulta

19 de enero de 2009].

**[31]** Patrones y Antipatrones: una Introducción - Parte II, 2008. Disponible en URL:

[http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_3317/default.aspx](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317/default.aspx).

Fecha de Consulta 19 de enero de 2009]. Autor: León Welicki



**[32]** Patrones y Antipatrones: una Introducción - Parte II, 2008. Disponible en URL: [http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ\\_3317/default.aspx](http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3317/default.aspx). Fecha de Consulta 19 de enero de 2009]. Autor: León Welicki.

# BIBLIOGRAFÍA

ABCdatos. (2007). "Guías Webmasters." from

<http://www.abcdatos.com/webmasters/programa/z5030.html>.

Astic. (2002). "Creación de una intranet básica." from

<http://portal.astic.es/Astic/Estudios+ASTIC/Monogr%C3%A1ficos/intranet.htm>.

Cesteros, A. F.-P. and M. M. d. Barrio Estudios de Lingüística del Español (ELiEs). Volume, DOI:

DBS. (1992). "DBS Servicios informáticos." from

[http://www.dbs.es/opencms/servicios/contenidos/gestion\\_de\\_contenidos.html](http://www.dbs.es/opencms/servicios/contenidos/gestion_de_contenidos.html).

DesarrolloWeb. (2004). "Desarrollo Web: Manual "Tutorial de SQL"." from

<http://www.desarrolloweb.com/manuales/9/>.

DesarrolloWeb.com Que es Python? Volume, DOI:

Fernandez, L. R. (2005). from

[http://www.telecom.cide.edu/include/internet\\_conference\\_2005/LRFernandez\\_Internet\\_Cuba.pdf](http://www.telecom.cide.edu/include/internet_conference_2005/LRFernandez_Internet_Cuba.pdf).

Foundation, W. (2005). "Mambo Open Source." From

[http://es.wikipedia.org/wiki/Mambo\\_Open\\_Source](http://es.wikipedia.org/wiki/Mambo_Open_Source).

Foundation, W. (2006). "Joomla." from <http://es.wikipedia.org/wiki/Joomla>.

Foundation, W. (2006). "Sistemas de Gestión de Contenidos." from

[http://es.wikipedia.org/wiki/Sistema\\_de\\_gesti%C3%B3n\\_de\\_contenido](http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenido)

INEI, I. N. d. E. e. I. (2004). Arquitectura Cliente/Servidor.

Infórmate, C. I. "Glosario de términos bibliotecológicos y de Ciencias de la Información." from

[http://www.uh.cu/facultades/fcom/portal/interes\\_glosa\\_terminos.htm](http://www.uh.cu/facultades/fcom/portal/interes_glosa_terminos.htm)

Molpeceres, A. (2005). "Procesos de desarrollo: RUP, XP y FDD." from

<http://www.javahispano.org/articles.article.action?id=76>

Perdure, T. (2005) MySQL and PostgreSQL Compared. Volume, DOI:

Phpnuke. "phpnuke.org." from <http://www.phpnuke.org>.

Quiroga, L. A. (2002). from [http://bvs.sld.cu/revistas/aci/vol10\\_5\\_02/aci04502.htm](http://bvs.sld.cu/revistas/aci/vol10_5_02/aci04502.htm).

Sitepoint (2004) The J2EE guy still doesn't get PHP Volume, DOI:

Tramullas, J. (2005). "Herramientas de Software Libre para la gestión de contenidos." from <http://www.hipertext.net/web/pag258.htm>

# Anexos

## Anexo 1

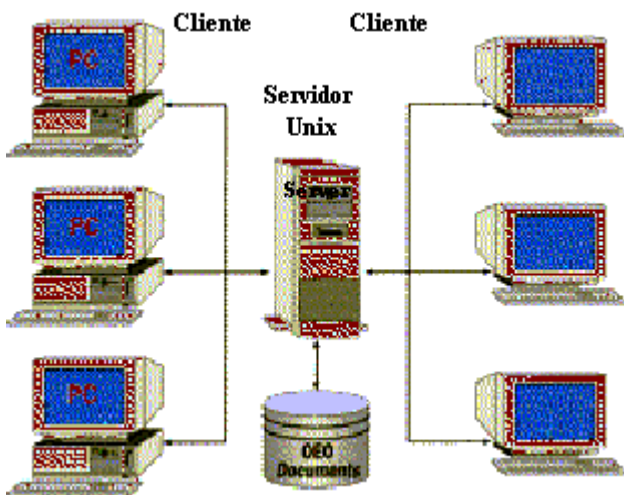


Figura #19 El modelo de la arquitectura cliente/servidor

## Anexo 2

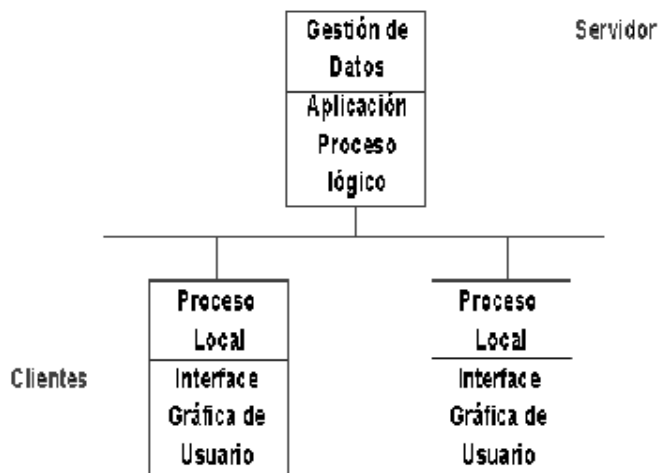
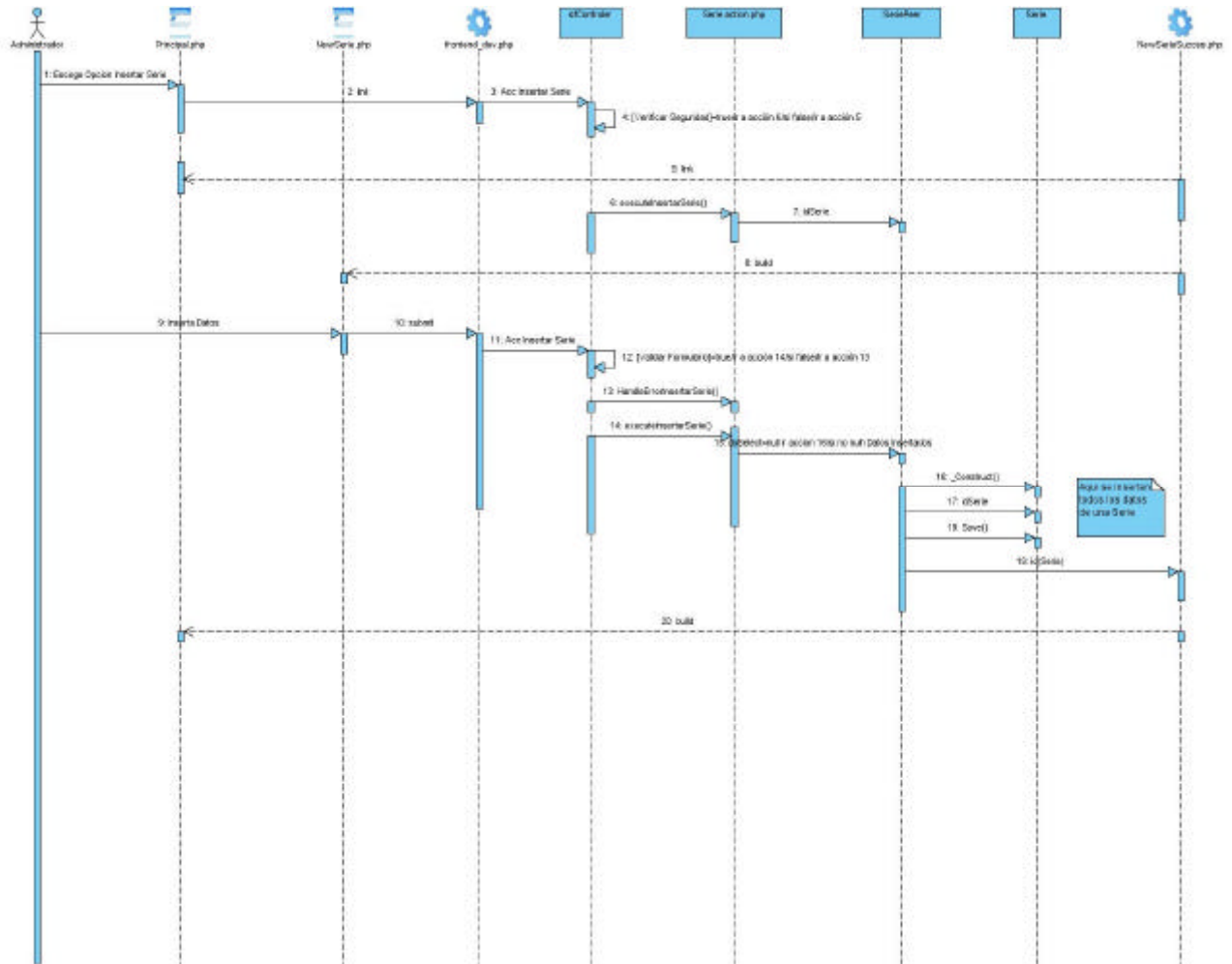


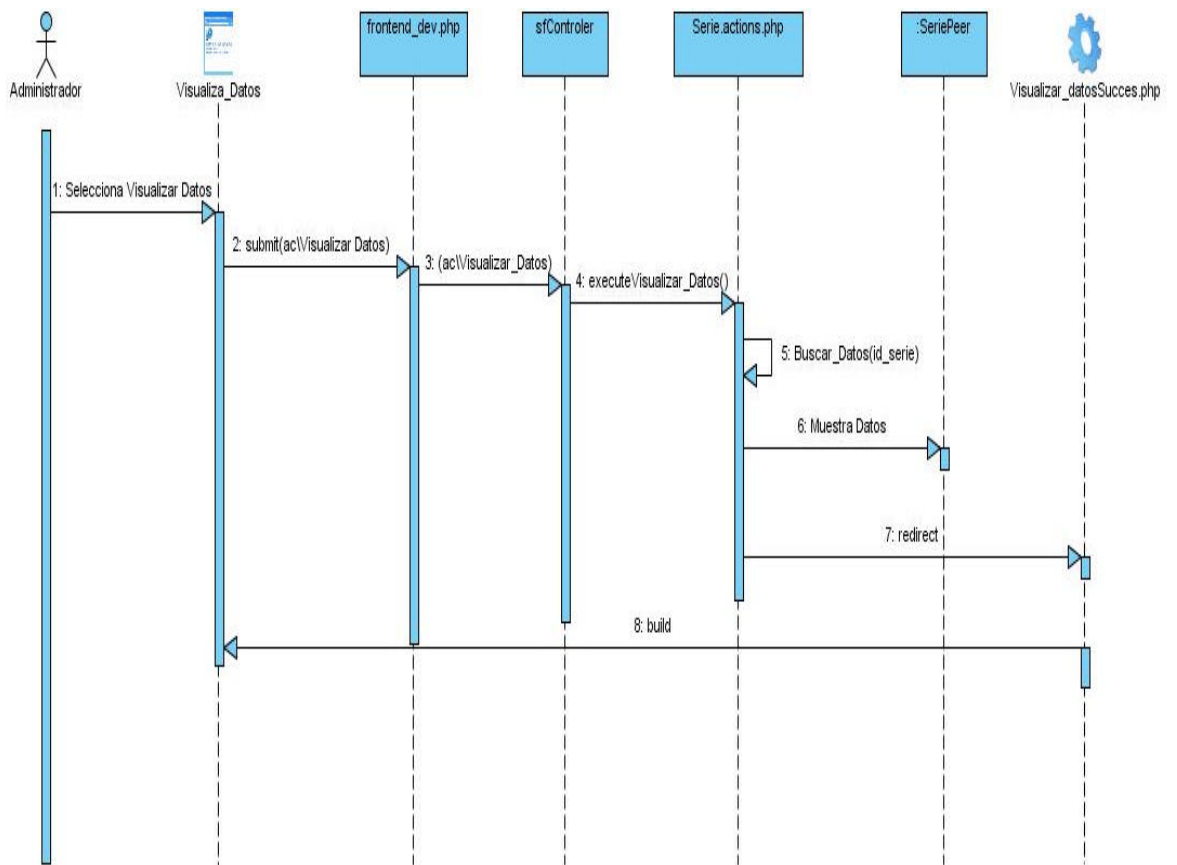
Figura #20 Aplicaciones Cliente – Servidor

### Anexo 3.



Figura# 21 Diagrama de Secuencia Caso de Uso Gestionar Serie  
Escenario Insertar Serie

## Anexo 4.



**Figura# 22 Diagrama de Secuencia Caso de Uso Gestionar Serie Escenario Visualizar Serie**

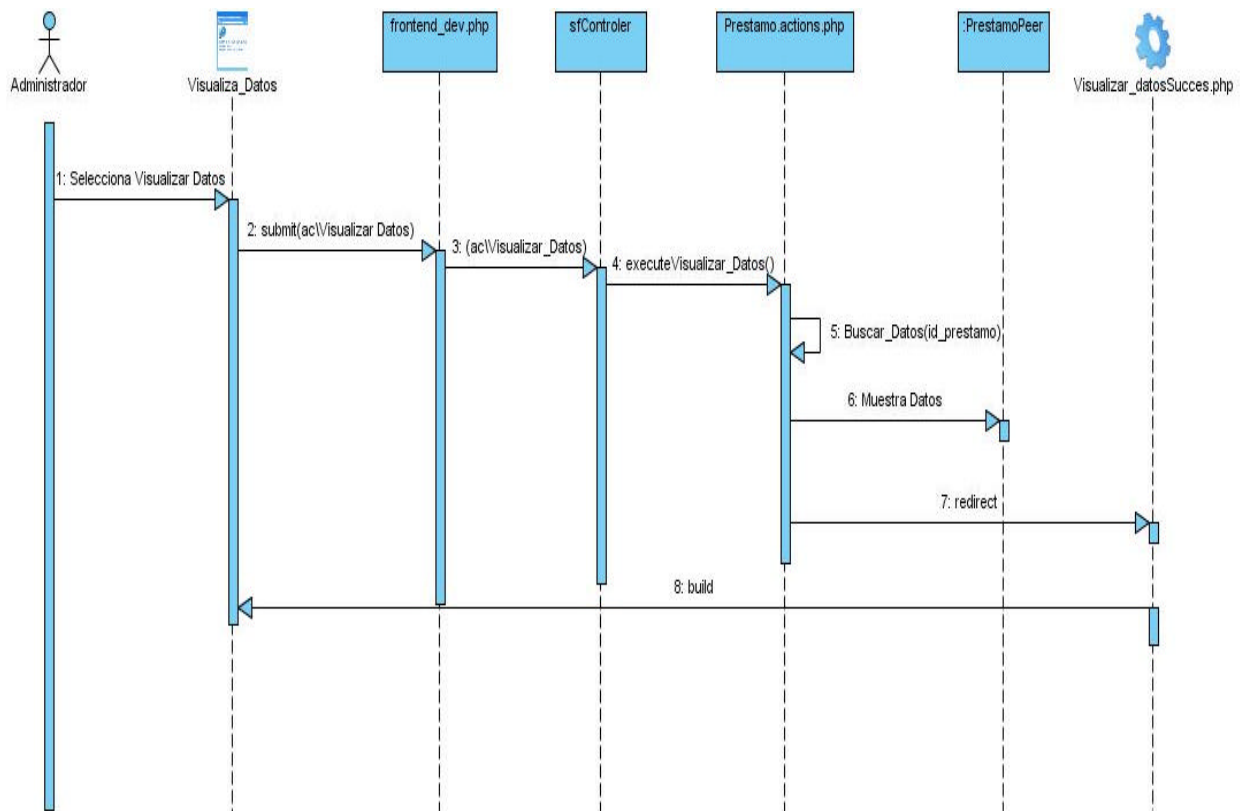








## Anexo 8.



**Figura# 25 Diagrama de Secuencia Caso de Uso Gestionar Préstamo Escenario Visualizar Préstamo**

# GLOSARIO DE TÉRMINOS Y SIGLAS

## Artefactos

Una parte de la información que (1) es producida, modificada, o usada por un proceso, (2) define un área de responsabilidad, y (3) está sujeta al control de versión. Un artefacto puede ser un modelo, un elemento del modelo, o un documento. Un documento puede adjuntar otros documentos. Una parte de la información que es usada o producida por un proceso de desarrollo del software. Un artefacto puede ser un modelo, una descripción, o un software.

**Apache:** El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix, Windows y otras, que implementa el protocolo HTTP/1.1.

**ASP.NET:** es un conjunto de tecnologías de desarrollo de aplicaciones web comercializado por Microsoft. Es usado por programadores para construir sitios web domésticos, aplicaciones web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

**BD:** Conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerado una colección de datos variables en el tiempo

**BSD:** son las iniciales de Berkeley Software Distribution (en español, Distribución de Software Berkeley) y se utiliza para identificar un sistema operativo derivado del sistema Unix nacido a partir de las aportaciones realizadas a ese sistema por la Universidad de California en Berkeley.

**CASE:** (*Computer-Aided Software Engineering*) es el uso de un método asistido por computadora para organizar y controlar el desarrollo de un software.

**Código Abierto:** Es una tendencia internacional del desarrollo de software que profesa la distribución del código junto a las aplicaciones, se rigen por licencias tales como GNU/GPL.

**Empresa:** en términos estrictamente económico, es una unidad económica que reúne una serie de factores de producción: recursos naturales, humanos, tecnológicos (o de capital) y

financieros (que posibilitan la adquisición de los anteriores), y los utiliza para producir bienes y/o servicios, que vende a personas individuales, a otras empresas y/ o a las administraciones públicas.

**Flujos de publicación:** Se llama a aquel proceso a través del cual los contenidos publicados en cualquier portal hayan sido redactados, revisados y publicados por varias personas.

**Gestión:** Conjunto de trámites que se llevan a cabo para resolver un asunto

**Gestión de información:** Comprende las actividades relacionadas con la obtención de la información adecuada, a un precio adecuado, en el tiempo y lugar adecuado, para tomar la decisión adecuada

**GNU/GPL:** Es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**HTML:** el Lenguaje de Marcado de HiperTextos (HTML - *HyperText Markup Language*) es un lenguaje que sirve para especificar la estructura de documentos que se visualizan a través de Internet usando programas de despliegue.

**HTTP:** HTTP o *HiperText Transfer Protocol* (protocolo de transferencia de hipertexto) Es el grupo de reglas, o protocolos, que gobiernan la transferencia de hipertexto entre dos o más computadoras. Es muy cómodo y fácil de usar para transferir texto, imágenes, sonido, etc.

**IDE:** Ambiente de desarrollo integrado. Es como se le llama al ambiente que proporciona al usuario una determinada herramienta de desarrollo.

**IMAP:** (acrónimo inglés de **I**nternet **M**essage **A**ccess **P**rotocol) es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet.

**Interfaz:** frontera convencional entre dos sistemas o dos unidades, que permite intercambio de informaciones.

**Internet:** se llama Internet a un conjunto de redes de computadoras y equipos unidos físicamente mediante cables que conectan puntos de todo el mundo. Estos cables pueden ser: de redes locales, telefónicos convencionales, digitales o canales de fibra óptica. En algunos

tramos, los datos se transmiten vía satélite o a través de servicios como la telefonía celular. Forman parte de Internet computadoras personales, bases de datos, servicios, fax y números telefónicos.

**ISAM:** Son siglas de *Indexed Sequential Access Method* (Método de Acceso Secuencial Indexado), se trata de un método para almacenar información a la que se pueda acceder rápidamente.

**ISS:** Internet Information Services. Servidor de red poderoso que mantiene una infraestructura de aplicación de red muy fiable, manejable, y escalable todas las versiones de Windows Servidor 2003. IIS ayuda a las organizaciones a aumentar los sitios de red y disponibilidad de la aplicación.

**Java:** Java es un lenguaje de programación expresamente diseñado para usarse en el entorno distribuido de Internet. Se diseñó para que se "pareciera" al lenguaje C++, pero es más sencillo de usar que éste y obliga a una visión de la programación completamente orientada a objetos. Java puede usarse para crear aplicaciones completas que corran en un sólo ordenador o se distribuyan entre servidores y clientes de una red. También puede usarse para construir pequeños módulos de aplicación o applets para utilizarlos como parte de una página Web.

**Linux:** es la denominación de un sistema operativo tipo-Unix y el nombre de un núcleo. Es uno de los paradigmas más prominentes del software libre y del desarrollo del código abierto, cuyo código fuente está disponible públicamente.

**LiveJournal:** (a menudo abreviado como LJ) es el nombre de un sitio de weblog que permite a los internautas mantener un periódico o diario en línea. También es el nombre del software de servidor de código abierto que fue diseñado para ejecutarlo.

**Mac OS X:** es el actual sistema operativo de la familia de ordenadores Macintosh.

**Microsoft:** Compañía de software más grande del mundo. Fue fundada en 1975 por Paul Allen y Bill Gates. Aunque también se conoce por sus lenguajes de programación y aplicaciones para computadores personales, el éxito sobresaliente de Microsoft se debe a sus sistemas operativos DOS y Windows.

**Microsoft Windows:** (conocido simplemente como **Windows**) es un sistema operativo con interfaz gráfica para computadoras personales cuyo propietario es la empresa Microsoft.

**MySQL AB :** Es una compañía de software fundada en 1995, creadora del sistema administrador de bases de datos relacionales MySQL, y una de las más grandes empresas de software libre del mundo

**MyISAM:** es la tecnología de almacenamiento de datos usada por defecto por el sistema administrador de bases de datos relacionales MySQL. Este tipo de tablas están basadas en el formato ISAM pero con nuevas extensiones.

**NNTP:** Network News Transport Protocol (NNTP) es una aplicación de internet que consiste en un protocolo usado para la lectura y publicación de artículos de noticias. Su traducción literal al español es "protocolo para la transferencia de noticias en red".

**ON LINE:** En línea o sea que puede ser accedido por la Web

**Plugin:** (o plug-in -en inglés "enchufar"-, también conocido como addin, add-in, addon o add-on) es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, generalmente muy específica, como por ejemplo servir como driver en una aplicación, para hacer así funcionar un dispositivo en otro programa.

**Proceso:** secuencia de actividades invocadas para producir un producto de software.

**Protocolo:** conjunto de normas que especifican como se comunican dos ordenadores entre sí y como intercambian información.

**RAM:** 'Random Access Memory'. La RAM se usa para mantener los programas mientras se están ejecutando, y los datos mientras se los procesa. La RAM es volátil, lo que significa que la información escrita en la RAM desaparecerá cuando se apague la alimentación de energía del ordenador.

**Servidor:** Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un ordenador y los servicios de aplicaciones, que realizan tareas en beneficio directo del usuario final. No es necesariamente una máquina de última generación grande y monstruosa, no es necesariamente un superordenador, un servidor puede ser desde una computadora vieja (ej.: pentium 2) hasta una máquina sumamente potente (ej.: servidores web, bases de datos

grandes, etc. Procesadores especiales y hasta varios y gigas de memoria). Todo esto depende del uso que se le dé al servidor.

**Software:** palabra en inglés utilizada para indicar a los programas de computadoras, a las aplicaciones.

**Software Libre:** Es el software que, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

**SNMP:** El Protocolo Simple de administración de red es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red.

**SQL-92:** nuevo estándar ampliado y revisado del SQL.

**Unix:** Sistema operativo atribuido a Ken Thompson y comercializado por la empresa ATT en la década de los 70s que alcanzó mucho éxito, sobre todo en las universidades y posteriormente en las empresas. Entre sus principales características tenemos que es: portable, robusto, y flexible actualmente goza de gran popularidad dentro de la tecnología de Internet.

**URL:** URL significa un localizador de Recursos Uniformes. De hecho los URLs son para los bits de información, el equivalente a las calles en una dirección de Internet. Los URL son uno de varios métodos para localizar, manejar e identificar bits de información en Web. La mayoría de los URL tienen tres partes principales: el protocolo, el nombre del anfitrión y el directorio. El protocolo representa la manera en que el documento será accedido; el nombre del anfitrión es el sistema informático en Internet donde se almacena la información, y el directorio representa la localización del archivo o cualquier otro tipo de información en el anfitrión.



**Web Services:** Componente de software que puede auto describirse y provee cierta funcionalidad a otras aplicaciones, a través de una conexión de Internet. Esas aplicaciones, acceden los Web Services vía protocolos Web y formatos de datos estándares, tales como HTTP y XML, sin tener en cuenta en absoluto cómo los Web Services están implementados.

**World Wide Web:** WWW (*World Wide Web*) es un sistema de información global, interactivo, dinámico, distribuido, gráfico, basado en Hipertexto, con plataforma de enlaces cruzados, que se ejecuta en Internet. Red alrededor del mundo. Es otra de las maneras de referirse a Internet.

**XML:** lenguaje extensible de marcado (*extensible Markup Language*), con un formato basado en el etiquetado textual para documentos y datos. Proviene del SGML (*Standard Generalized Mark-up Language*). Fue aceptado por la *World Wide Web* (W3C) en Febrero de 1998. XML no es realmente un nuevo lenguaje, es un metalenguaje usado para definir a otros lenguajes, actuando como protocolo integrador entre aplicaciones. Permite crear documentos bien estructurados y como resultado todo lenguaje basado en XML también será bien estructurado, lo que significa que los datos en XML son más fáciles de usar; así XML es una completamente nueva manera de comunicarse a través de Internet, por que permite a los negocios y sus sistemas de computadoras comunicarse más fácilmente.